Blitz Identity Provider

Версия 5.26

Полное руководство

11 октября 2024

Оглавление

Сервер аутентификации Blitz Identity Provider защищает пользовательские учетные записи — предоставляет готовые, гибко настраиваемые под заказчика и реализованные с учетом лучших практик функции защиты учетных записей.

Blitz Identity Provider обеспечивает доступ пользователей Интернет к вебсайтам и мобильным приложениям компании, а также доступ сотрудников к внутренним ресурсам компании и облачным сервисам.

Основные функции Blitz Identity Provider:

- обеспечение единого сквозного входа пользователя в приложения (Single Sign-On);
- двухфакторная аутентификация;
- конфигурируемый пользовательский интерфейс страниц входа, регистрации, восстановления доступа, управления учетной записью;
- вход с использованием сторонних поставщиков идентификации: вход с помощью аккаунтов социальных сетей, банков, Единой системы идентификации и аутентификации (ЕСИА, Госуслуги), Mos ID (СУДИР), федеративный вход пользователей с использованием внешних поставщиков идентификации;
- проверка прав доступа на вход пользователей в приложения;
- проверка прав доступа пользователей и приложений при использовании REST-сервисов;
- протоколирование событий доступа и действий с учетными записями.

Глава 1

Функциональная спецификация

Группа функций	Функции
Технологии единого входа	
OpenID Connect и OAuth 2.0	RFC 6749 “The OAuth 2.0 Authorization Framework”
	OpenID Connect Core 1.0
	Передача атрибутов пользователя в составе id_token/access_token в JSON Web Token (JWT)
	Конфигурируемый REST-сервис UserInfo, настройка возвращаемых атрибутов в зависимости от scope
	RFC 7636 “Proof Key for Code Exchange by OAuth Public Clients”
	RFC 7662 “OAuth 2.0 Token Introspection”
	RFC 7591 “OAuth 2.0 Dynamic Client Registration Protocol”
	RFC 7592 “OAuth 2.0 Dynamic Client Registration Management Protocol”
	RFC 8252 “OAuth 2.0 for Native Apps”
	RFC 8414 “OAuth 2.0 Authorization Server Metadata”
	OpenID Connect RP-Initiated Logout 1.0
	OpenID Connect Front-Channel Logout 1.0
	OpenID Connect Back-Channel Logout 1.0
SAML	SAML Web Browser SSO Profile
	SAML Single Logout Profile
RADIUS	RFC 2865 «Remote Authentication Dial In User Service (RADIUS)»
WS-Federation	WS-Federation (для подключения Microsoft-приложений)
Proxy SSO	Подключения веб-приложений, получающих состояние сессии из HTTP-заголовков и cookies
	Поддержка возможности заполнения за пользователя логина/пароля от учетной записи в размещенное за proxy веб-приложение, не поддерживающее стандартным образом подключения к SSO
Другое	Единый вход работает между приложениями, которые подключены к IDP с использованием любых поддерживаемых технологий (например, SSO между OpenID Connect и SAML-приложениями)
	Поддержка SSO-входа с использованием Kerberos SSO

continues on next page

Таблица 1 – продолжение с предыдущей страницы

Группа функций	Функции
	Поддержка единого SSO с приложениями IBM, использующими для единого входа Ltpa2Token
Идентификация и аутентификация	
Вход по логину и паролю	Проверка логина/пароля при аутентификации
	Возможность в качестве логина одновременно использовать несколько сущностей (телефон, email, логин) и вводить логин в разных форматах (например, вводить телефон как +7..., 8..., с разным вариантом ввода скобок, дефисов, пробелов)
	Запоминание логина, если пользователь ранее уже входил с этого устройства
	Запоминание на устройстве нескольких пользователей. Возможность сменить текущую учетную запись пользователя без необходимости логаута
	Обработка события «пароль требует смены» при входе. Возможности сменить пароль в момент входа
	Проверка соответствия пароля действующей парольной политике при входе. Рекомендация сменить пароль
	Встроенная защита от подбора пароля (перебор паролей на одну учетную запись) и подбора логина (попытка подбора пароля на набор учетных записей): • проверка CAPTCHA (reCAPTCHA или иной сервис, выбранный Заказчиком) • временное блокирование входа по паролю учетной записи при выявленных попытках перебора • замедление входа пользователя (задержка входа, решение браузером вычислительно сложной задачи – Proof of Work)
	Предупреждение пользователя о попытке входа с паролем, который был недавно изменен
Вход на основе сеанса	Идентификация пользователя на основе результата входа в домен (Kerberos)
	Возможность подключения системы входа одновременно к нескольким доменам и обеспечения сквозного входа пользователей из разных доменов
	Возможность настройки, чтобы режим входа на основе сеанса ОС применялся только при входе из внутренних сетей и с ПК, но не применялся при входе с мобильных приложений и из вне рабочей сети

continues on next page

Таблица 1 – продолжение с предыдущей страницы

Группа функций	Функции
Вход через аккаунт социальной сети / стороннего поставщика идентификации	Социальные сети и внешние поставщики идентификации, через которые поддерживается возможность входа пользователей без необходимости доработок и написания коннекторов: • Яндекс, ВКонтакте, Одноклассники, Mail ID, VK ID, Apple ID, Google, Facebook[1] • Вход через ЕСИА в режиме физического лица • Вход через ЕСИА в режиме представителя организации (с выбором организации при входе) • Вход через ЕСИА в режиме цифрового профиля • Вход через банки Сбер ID, T-ID, ВТБ ID, СберБизнес ID, Альфа ID • Вход через Mos ID (СУДИР) • Вход через СУДИС
	Вход через внешний поставщик идентификации с поддержкой OIDC
	Вход через внешний поставщик идентификации с поддержкой SAML
	Сопоставление/регистрация учетной записи в процессе первичного входа через социальную сеть
	Возможность привязки к одной учетной записи пользователя одновременно нескольких учетных записей внешних поставщиков
	Возможность привязки к одной учетной записи внешнего поставщика нескольких учетных записей пользователей
	Возможность программирования собственного алгоритма привязки учетных записей и сопоставления атрибутов
	Возможность сохранять маркеры доступа из внешних поставщиков
Вход на основе запомненного устройства	Автоматическая идентификация пользователя, если он уже входил с этого устройства и согласился запомнить свой вход
	Возможность пользователю отследить, на каких устройствах запомнен вход, и выйти с этих устройств
	Автоматический выход с запомненных устройств при смене/восстановлении пароля пользователем
Автоматическая идентификация по свойствам сессии	Автоматическая идентификация пользователя по свойствам сессии. Поддерживаются все свойства. сессии, которые могут быть определены Заказчиком и предоставлены в Blitz Identity Provider. Гибкая настройка метода и полная кастомизация текстов интерфейса.
Вход с помощью WebAuthn, Passkey, FIDO2	Вход с помощью платформо-независимых ключей безопасности FIDO2

continues on next page

Таблица 1 – продолжение с предыдущей страницы

Группа функций	Функции
	Вход с помощью платформо-зависимых ключей безопасности Passkey / FIDO2 – Windows Hello (пин-код, отпечаток пальца), Passkey, пароль или Touch ID от MacBook, Passkey, Face ID или Touch ID от смартфона или планшета с iOS или Android
Вход с помощью смарт-карты / USB ключа	Вход с помощью средств квалифицированной электронной подписи
	Поддерживаемые средства электронной подписи: КриптоПро CSP 3.9 и выше, VipNet CSP 4.2, Signal-COM CSP 3.0, Рутокен, JaCarta, ISBC ESMART, SafeNet eToken
	Поддерживаемые пользовательские ОС: Windows 8.1/10/11, macOS 10.13/10.14/10.15/11/12/13, Linux Debian 9, Mint 19, Ubuntu 18, Astra Linux 1.7, Red OS 7.3
	Поддерживаемые браузеры: Internet Explorer 11, Chrome, Firefox, Yandex, Спутник
	Возможность сопоставления/регистрации учетных записей в процессе первичного входа на основе данных из сертификата квалифицированной электронной подписи
	Возможность проверки действительности подписи/сертификата встроенными возможностями ПО
	Возможность проверки действительности подписи/сертификата через вызов внешнего сервиса проверки
Двухфакторная аутентификация	Подтверждение входа разовым паролем из SMS (SMS-шлюз предоставляет Заказчик)
	Подтверждение входа разовым паролем из email
	Подтверждение входа разовым паролем TOTP-приложения (RFC 6238 “TOTP: Time-Based One-Time Password Algorithm”)
	Подтверждение входа разовым паролем из аппаратного брелока. Поддержка брелоков HOTP (RFC 4226 “HOTP: An HMAC-Based One-Time Password Algorithm”). Брелоки предоставляет Заказчик
	Подтверждение входа ключом безопасности WebAuthn, Passkey, FIDO2
	Подтверждение входа ключом безопасности U2F
	Подтверждения входа разовым паролем в push-уведомлении в мобильном приложение Заказчика (сервис для отправки push и мобильное приложение предоставляет Заказчик)
	Подтверждение входа по входящему звонку (Flash Call)
Другое	Возможность Заказчику самостоятельно добавить собственный метод аутентификации
	Возможность Заказчику самостоятельно настроить внешний вид страницы входа раздельно для каждого приложения, в которое осуществляется вход
	Предоставление API, позволяющее мобильным приложениям зарегистрировать событие входа и получить маркеры безопасности при входах с использованием ПИН-кода, Touch ID, Face ID

continues on next page

Таблица 1 – продолжение с предыдущей страницы

Группа функций	Функции
	Блокирование учетных записей в случае длительной неактивности
	Запрет на повторное использование идентификатора удаленной учетной записи в течение установленного времени
	Возможность анализа геоданных пользователя
Логаут	
Логаут	Завершение пользовательской сессии при инициировании логаута пользователем
	Завершение пользовательской сессии при смене пароля пользователя в другой сессии или при сбросе/восстановлении пароля пользователю
	Ограничение допустимых ссылок для возврата в приложение после успешного логаута
	Информирование приложений о произведенном едином логауте через браузер (front channel)
	Информирование приложений о произведенном едином логауте через сервер (back channel)
Контроль доступа	
Контроль доступа	Проверка правил доступа при входе пользователя в приложения. Проверка наличия у пользователя прав доступа, членства в группах пользователей, наличия атрибутов с требуемыми значениями
	Проверка правил доступа при вызове приложениями защищаемых REST-сервисов через шлюз безопасности Blitz Keeper (API Security Gateway)
Управление аккаунтом	
Регистрация	Настраиваемое веб-приложение самостоятельной регистрации пользователей. Можно настроить набор атрибутов, заполняемых пользователем при регистрации, требования к подтверждению email/телефона, настроить внешний вид страницы регистрации, вызов сервисов проверки Заказчика
	Можно задать различные настройки веб-приложения самостоятельной регистрации пользователя для различных сценариев вызова регистрации
	Возможность вызова внешнего приложения регистрации с передачей ему контекста входа и сведений,полученных из внешнего поставщика в процессе входа
	По результатам успешной регистрации пользователь автоматически входит в приложение, при попытке входа в которое изначально была инициирована регистрация
	Проверка CAPTCHA (reCAPTCHA или иной сервис, выбранный Заказчиком)

continues on next page

Таблица 1 – продолжение с предыдущей страницы

Группа функций	Функции
Настройки безопасности учетной записи	Веб-приложение, позволяющее пользователю управлять настройками безопасности его учетной записи: • возможность самостоятельно сменить пароль; • возможность редактирования некоторых атрибутов, в т.ч. возможность редактирования телефона с подтверждением через код по SMS и возможность редактирования email с подтверждением через код/ссылку по email; • возможность настроить двухфакторную аутентификацию для своей учетной записи; • возможность посмотреть/отредактировать список запомненных устройств, привязанных учетных записей внешних поставщиков входа; • возможность посмотреть события безопасности со своей учетной записью.
	Предоставление API для возможности встраивания всех вышеперечисленных функций управления настройками безопасности учетной записи в стороннее веб-приложение
Восстановление забытого пароля	Веб-приложения, позволяющего восстановить забытый пароль, с подтверждением email или телефона
	Дополнительные проверки при восстановлении пароля от учетной записи, для которой включена двухфакторная аутентификация
	По результатам успешного восстановления пароля пользователь автоматически входит в приложение, при попытке входа в которое изначально была инициирована процедура восстановления
	Проверка CAPTCHA (reCAPTCHA или иной сервис, выбранный Заказчиком)
Действия с учетной записью в момент входа	Возможность в момент входа задать номер телефона (если отсутствует) в учетной записи или подтвердить актуальность телефона (если наступил срок необходимости подтверждения актуальности)
	Возможность в момент входа задать номер телефона (если отсутствует) в учетной записи или подтвердить актуальность телефона (если наступил срок необходимости подтверждения актуальности)
	Возможность в момент входа задать адрес электронной почты (если отсутствует) в учетной записи или подтвердить актуальность адреса электронной почты (если наступил срок необходимости подтверждения актуальности)
	Возможность в момент входа выпустить Passkey (настроить вход по Face ID / Touch ID)
	Возможность показать пользователю объявление
	Возможность запросить у пользователя согласие

continues on next page

Таблица 1 – продолжение с предыдущей страницы

Группа функций	Функции
	Возможность запросить у пользователя заполнение текстового атрибута
	Возможность в момент входа задать контрольный вопрос
	Возможность встроить собственный бизнес-процесс взаимодействия с пользователем в момент входа в приложение (например, вывести пользователю информационное сообщение в каких-то ситуациях или запросить что-то вести)
Парольные политики	Проверка пароля на соответствие парольной политике: минимальная длина, требования к алфавиту, запрет словарных паролей, запрет повтора паролей, проверка срока действия паролей
Расширенные возможности	
Кастомизация логики работы с помощью программирования на Java	Задание правил входа пользователей в приложения посредством процедур входа и регистрации
	Кастомизация операций с хранилищами данных
Мониторинг и аудит	
Оповещения пользователей о событиях безопасности	Оповещение пользователей о событиях безопасности с их учетными записями: вход с необычного устройства, изменение пароля (сам сменил, администратор сбросил, смена в результате восстановления пароля), привязка учетной записи социальной сети, включение/выключение двухфакторной аутентификации
	Возможность настроить набор событий оповещения и тексты оповещений для SMS и для email
Регистрация событий безопасности	Регистрация успешных и неуспешных событий безопасности с учетной записью: события входа, регистрации, изменения настроек безопасности, восстановления пароля. Должны регистрироваться как действия, инициированные пользователем, так и действия, инициированные администратором
	Регистрация успешных и неуспешных событий безопасности с учетной записью: события входа, регистрации, изменения настроек безопасности, восстановления пароля. Должны регистрироваться как действия, инициированные пользователем, так и действия, инициированные администратором
	Сопоставление IP-адресам геоданных в событиях и уведомлениях (БД в формате mmdb с геоданными предоставляет Заказчик)
	Интерфейс администратора для поиска/просмотра событий безопасности
	Запись событий безопасности: в БД, в лог-файл, в Kafka
Мониторинг	Возможность в момент входа пользователя вызывать системы сбора метрик и статистики, антифрод системы

continues on next page

Таблица 1 – продолжение с предыдущей страницы

Группа функций	Функции
	Возможность осуществлять мониторинг компонент из внешней системы мониторинга (Zabbix и аналоги). Предоставление метрик в формате Prometheus
	Шаблоны дашборда Grafana и job-задания Prometheus в поставке
Очереди	Возможность передавать в очередь RabbitMQ события, связанные с учетными записями пользователей и групп доступа
	Возможность передавать в очередь Kafka события безопасности
Администрирование	

continues on next page

Таблица 1 – продолжение с предыдущей страницы

Группа функций	Функции
Администрирование	Веб-приложение администрирования: • задание настроек подключенных приложений (параметры приложений, разрешенные режимы взаимодействия, правила контроля доступа) • настройка атрибутов пользователей и сопоставление атрибутов хранилищам учетных записей • настройка подключения к хранилищам учетных записей на основе LDAP • настройка подключения к произвольным хранилищам (через предоставленный Заказчиком сервис) • поддержка работы одновременно с несколькими хранилищами учетных записей • настройка методов идентификации/аутентификации и внешних поставщиков входа • настройка подключения к SMTP-службе и к SMS-шлюзу • поддержка ролевого доступа для входа в веб-приложение администратора. Возможность для разных пользователей задать разный набор доступных действий • управление настройками веб-приложений регистрации, управления настройками безопасности, восстановления пароля • администрирование учетных записей пользователей (поиск, просмотр, управление атрибутами, настройками двухфакторной аутентификации, привязками запомненных устройств и социальных сетей, запомненных браузеров пользователя, сброс сессий, сброс пароля, блокирование/разблокирование учетной записи, управление ключами безопасности, управление членством в группах пользователей, назначение/отзыв прав доступа) • администрирование групп пользователей, управления членством пользователей в группах • настройка внешнего вида страниц входа в приложения • просмотр и фильтрация зарегистрированных событий безопасности • возможность входа в веб-приложение администрирования через SSO
	Интерфейс администратора на русском и английском языках
	Возможность добавления переводов на дополнительные языки

[1] Meta признана экстремистской организацией и запрещена в России, деятельность принадлежащих ей соц.сетей Facebook и Instagram также запрещена в РФ.

Глава 2

Администрирование

2.1 Развертывание

2.1.1 Архитектура развертывания

Функционирование Blitz Identity Provider основывается на взаимодействии следующих архитектурных компонентов:

1. Веб-сервер. Можно использовать существующий веб-сервер компании для балансировки нагрузки и снятия SSL-шифрования с входящего трафика.
2. Сервисы Blitz Identity Provider:
   - `blitz-console` – консоль управления Blitz Console;
   - `blitz-idp` - сервис аутентификации и «личный кабинет»;
   - `blitz-registration` – сервис регистрации;
   - `blitz-recovery` – сервис восстановления доступа;
   - `blitz-keeper` – *шлюз безопасности* (страница 546);
   - `blitz-panel` – *витрина* (страница 550), предоставляющая доступ пользователей к подключенным приложениям.

   Примечание: Сервисы регистрации, восстановления доступа, шлюз безопасности и витрину можно не устанавливать, если связанные с ними функции не планируется использовать.

3. СУБД. Можно использовать Couchbase Server, PostgreSQL, Postgres Pro, Jatoba.

   Внимание: Взаимодействие Blitz Identity Provider с PostgreSQL осуществляется по JDBC. Вместо PostgreSQL можно использовать любую реляционную СУБД с поддержкой JDBC, но это должно быть отдельно согласовано с нашими техническими специалистами в рамках соответствующих проектов внедрения.

   - Couchbase Server – рекомендуется при создании систем аутентификации с пиковой нагрузкой более 1000 запросов в секунду, количеством аутентификаций в сутки более 1 млн и с высокими требованиями к отказоустойчивости.
   - PostgreSQL (или иная реляционная СУБД, поддерживающая работу по JDBC) – рекомендуется при создании систем аутентификации с умеренной нагрузкой и средними требованиями к отказоустойчивости, а также при использовании отечественных операционных систем.

4. Хранилище учетных записей и паролей. Можно использовать как существующее, так и специально развернутое в организации хранилище учетных записей.

   Поддерживаются:

   - LDAP-совместимые хранилища. Это может быть любой сервер, поддерживающий протокол LDAP, а также Microsoft Active Directory, Samba4, FreeIPA;
   - иные типы хранилищ, для подключения Blitz Identity Provider к ним необходимо разработать специальные REST-сервисы.

   В случае необходимости развертывания нового LDAP-каталога рекомендуется в качестве LDAP-каталога использовать 389 Directory Server, который входит в состав ОС.

5. Опционально Сервер очередей – используется RabbitMQ. Также можно настроить передачу событий безопасности в Kafka. Установка сервера очередей RabbitMQ требуется, если сервер очередей будет использоваться для *передачи событий в смежные системы* (страница 323) или в качестве *брокера сообщений* (страница 324).

Развертывание возможно в конфигурации с *минимальными ресурсами* (страница 12) либо в *кластерной конфигурации* (страница 14).

2.1.2 Системные требования

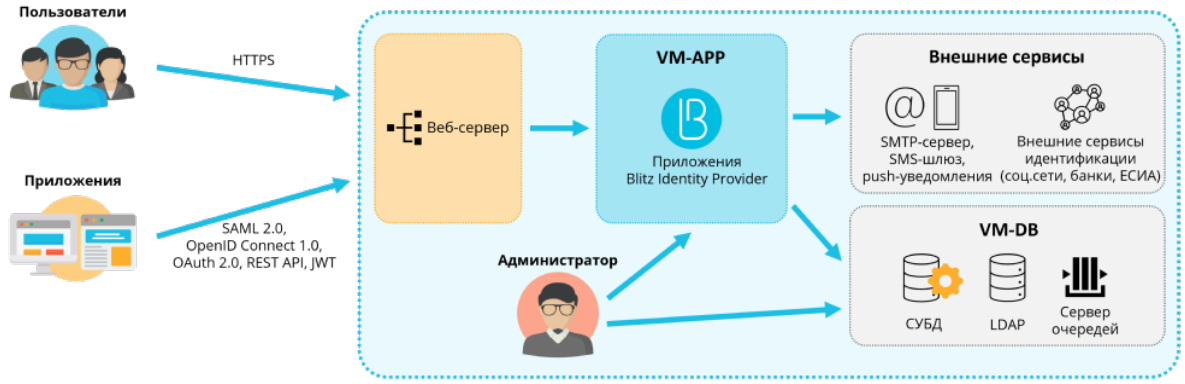
Операционные системы

Все варианты установки Blitz Identity Provider и задействованные в них типы серверов поддерживают следующие операционные системы:

Astra Linux SE 1.6/1.7 РЕД ОС 7.3 Альт Сервер 10 Альт 8 СП Сервер ОСнова 2.5.1 CentOS 7/8 Rocky Linux 8/9 AlmaLinux 8/9 RHEL 7/8/9 Oracle Linux 8/9 РОСА Хром 12 Red OS 8

Минимальные требования

Минимальные требования рекомендуется применять при подготовке сред тестирования и для продуктивных контуров при внедрениях со средними требованиями к обеспечению доступности и производительности согласно приведенной ниже схеме.



Минимально для развертывания необходимо использовать 2 виртуальные машины (далее – ВМ) со следующими характеристиками и ролями.

Минимальные требования к серверам для развертывания

Развертывание на 1-й ВМ не в кластере

Описание	Технические характеристики	ПО
ВМ для приложений и СУБД	4 ядра ЦПУ, 8 ГБ ОЗУ, 50 ГБ НЖМД (HDD)	Blitz Identity Provider: `blitz-idp`, `blitz-console`, `blitz-registration`, `blitz-recovery`, `blitz-panel`; `JDK`, `nginx или HAProxy`, `memcached`, `PostgreSQL`, `LDAP`

Развертывание на 2-х ВМ не в кластере

Описание	Технические характеристики	ПО
ВМ для приложений (VM APP)	4 ядра ЦПУ, 8 ГБ ОЗУ, 50 ГБ НЖМД (HDD)	Blitz Identity Provider: `blitz-idp`, `blitz-console`, `blitz-registration`, `blitz-recovery`, `blitz-keeper`, `blitz-panel`; `JDK`, `nginx или HAProxy`, `memcached`
ВМ для базы данных (VM DB)	4 ядра ЦПУ, 8 ГБ ОЗУ, 100 ГБ НЖМД (HDD)	PostgreSQL (9.6 или новее) или Couchbase Server Community Edition (6.0 или новее); 389 Directory Server или FreeIPA; RabbitMQ (опционально)

Развертывание на 2-х ВМ в кластере

Описание	Кол-в	Технические характеристики	ПО
ВМ для приложений и СУБД	2	4 ядра ЦПУ, 8 ГБ ОЗУ, 50 ГБ НЖМД (HDD)	Blitz Identity Provider: `blitz-idp`, `blitz-registration`, `blitz-recovery`, `blitz-panel`; `JDK`, `nginx или HAProxy`, `memcached`, `PostgreSQL`, `LDAP`

Требуемые версии системного ПО:

- OpenJDK 11, Liberica JDK 11, Axiom JDK 11 Certified или Oracle JDK 11;
- Менеджер памяти Memcached версии 1.4.15 или выше.

Требования к сетевой связности:

- VM-APP должна быть доступна по `80`, `443` (HTTP/HTTPS) из сетей пользователей;
- с VM-APP должен быть доступ:
  - к VM-DB по `8091`, `8092`, `8093`, `11209`, `11210`, `11211`, `4369`, `21100 - 21199`, `11214`, `11215`, `18091`, `18092` (стандартные порты Couchbase Server), `5432` (стандартный порт PostgreSQL), `389`, `636` (стандартные порты LDAP), `5672` (стандартный порт RabbitMQ);
  - к сервисам внешних поставщиков идентификации по `443` (при их использовании):

Ссылки на сервисы внешних поставщиков

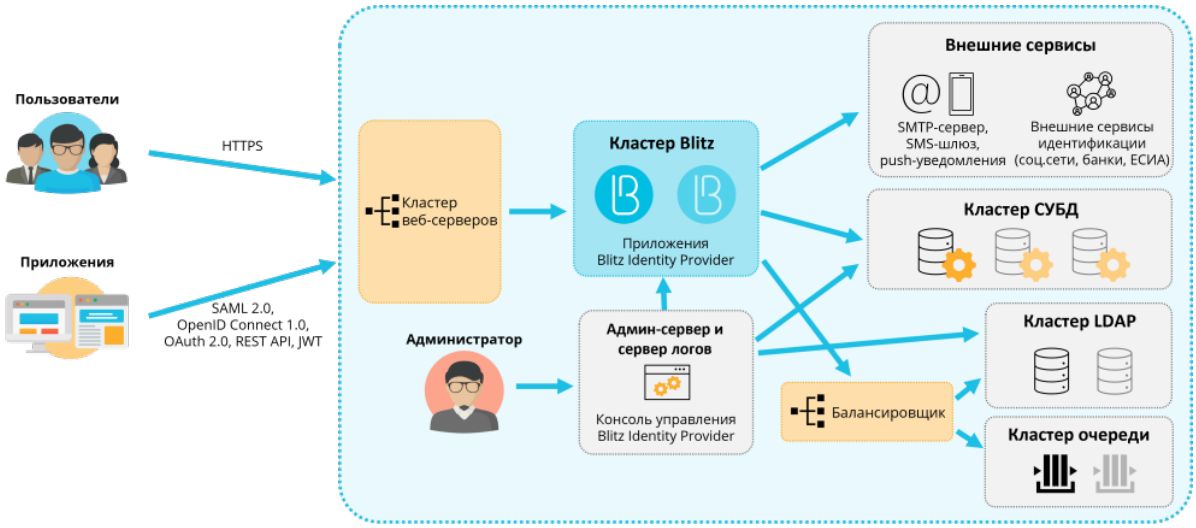
Тип	Ссылка
Социальные сети	`https://appleid.apple.com`
	`https://accounts.google.com`
	`https://graph.facebook.com`[1]
	`https://oauth.yandex.ru`
	`https://oauth.vk.com`
	`https://account.mail.ru`
	`https://api.ok.ru`
ЕСИА и цифровой профиль	`https://esia-portal1.test.gosuslugi.ru`
	`https://esia.gosuslugi.ru`
Банки	`https://online.sberbank.ru`
	`https://business.tinkoff.ru`
	`https://id.vtb.ru`
	`https://sbi.sberbank.ru:9443`
	`https://fintech.sberbank.ru:9443`
	`https://id-sandbox.alfabank.ru`
СУДИР	`https://login.mos.ru`
	`https://login-tech.mos.ru`
	`https://sudir.mos.ru`
	`https://sudir-test.mos.ru`

- к SMS-шлюзу (при его использовании);
- к SMTP (при его использовании);
- к сервису push-уведомлений (при его использовании);
- к сервису Kafka (при его использовании для приема событий безопасности).

Для VM-APP нужно завести публичное DNS-имя (например, `auth.domain.ru`) и выпустить TLS-сертификат на `auth.domain.ru` или `*.domain.ru`.

Рекомендуемые требования для кластера

Схема развертывания в кластерной конфигурации приведена на рисунке ниже. Следуйте приведенным в данном разделе требованиям при построении продуктивных контуров систем аутентификации с высокими требованиями к доступности и пиковой производительности.



[1] Meta признана экстремистской организацией и запрещена в России, деятельность принадлежащих ей соц.сетей Facebook и Instagram также запрещена в РФ.

Для развертывания в кластерной конфигурации рекомендуется использовать виртуальные машины (далее – ВМ) со характеристиками и ролями, указанными в таблице ниже.

Рекомендуемые требования к серверам для развертывания в кластере

Описание	Кол-в	Технические характеристики	ПО
ВМ веб-сервер (VM-WEB)	1-2	4 ядра ЦПУ, 4 ГБ ОЗУ, 50 ГБ НЖМД (HDD)	`nginx или HAProxy`
ВМ приложений Blitz Identity Provider (VM-APP)	2	4 ядра ЦПУ, 8 ГБ ОЗУ, 50 ГБ НЖМД (HDD)	Blitz Identity Provider: `blitz-idp`, `blitz-registration`, `blitz-recovery`, `blitz-keeper`, `blitz-panel`; `memcached`, `JDK`
ВМ для консоли (VM-ADM)	1	2 ядра ЦПУ, 4 ГБ ОЗУ, 100 ГБ НЖМД (HDD)	`memcached`, `JDK`; Blitz Identity Provider: `blitz-console`
ВМ для СУБД (VM-DB)	2-3	Для PostgreSQL: 4 ядра ЦПУ, 8 ГБ ОЗУ, 100 ГБ НЖМД (HDD) (данные), 50 ГБ НЖМД (HDD) (система). Для Couchbase Server[3]: 8 ядер ЦПУ, 16 ГБ ОЗУ, 500 ГБ НЖМД (HDD) (данные), 100 ГБ НЖМД (SSD) (индексы), 50 ГБ НЖМД (HDD) (система)	ПО PostgreSQL (9.6 или новее) или Couchbase Server Community Edition (6.0 или новее)
ВМ для LDAP (VM-LDAP)	2	4 ядра ЦПУ, 8 ГБ ОЗУ, 100 ГБ НЖМД (HDD)	389 Directory Server
ВМ для сервера очередей (VM-MQ)	1-2	4 ядра ЦПУ, 8 ГБ ОЗУ, 50 ГБ НЖМД (HDD)	RabbitMQ версии 3.7.9
ВМ для балансировщика (VM-NLB)	1-2	2 ядра ЦПУ, 4 ГБ ОЗУ, 50 ГБ НЖМД (HDD)	`HAProxy`, `keepalived`

Совет:

- `VM-WEB`:

  Можно использовать существующий веб-сервер для балансировки нагрузки и снятия TLS с входящего трафика.

- `VM-APP`:

  При высокой нагрузке рекомендуется развертывать каждое приложение Blitz Identity Provider в своем кластере на отдельных серверах.

- `VM-ADM`:

  На этот сервер рекомендуется сбор логов с других серверов кластера.

- `VM-DB`:

  Для PostgreSQL рекомендуется выделить один физический сервер под основной экземпляр и один под резерв (standby). Для Couchbase Server рекомендуется минимум[4] 3 ВМ.

[3] https://docs.couchbase.com/server/current/install/install-linux.html
[4] https://docs.couchbase.com/server/current/install/deployment-considerations-lt-3nodes.html

- `VM-LDAP`:

  В качестве хранилища можно использовать существующее хранилище на основе LDAP, Microsoft Active Directory, FreeIPA, либо иную систему хранения учетных записей и паролей (подключение через REST коннектор).

- `VM-MQ`:

  Использование сервера очередей опционально.

- `VM-NLB`:

  Внутренний балансировщик нужен в случае кластеризации LDAP и сервера очередей.

Требуемые версии системного ПО:

- OpenJDK 11, Liberica JDK 11, Axiom JDK 11 Certified или Oracle JDK 11;
- Менеджер памяти Memcached версии 1.4.15 или выше;

Требования к сетевой связности:

- VM-WEB должна быть доступна по `80`, `443` (HTTP/HTTPS) из сетей пользователей;
- с VM-WEB должен быть доступ к VM-APP по `9000` (`blitz-idp`), `9002` (`blitz-registration`), `9003` (`blitz-recovery`), `9012` (`blitz-keeper`), `9013` (`blitz-panel`) и VM-ADM по `9001` (`blitz-console`);
- с VM-APP должен быть доступ:
  - к другим VM-APP и VM-ADM по `11211` (memcached);
  - к VM-DB по `8091`, `8092`, `8093`, `11209`, `11210`, `11211`, `4369`, `21100 - 21199`, `11214`, `11215`, `18091`, `18092` (стандартные порты Couchbase Server) или `5432` (стандартный порт PostgreSQL);
  - к VM-LDAP (VM-NLB) по `389`, `636` (стандартные порты LDAP);
  - к VM-MQ (VM-NLB) по `5672` (стандартный порт RabbitMQ);
  - к сервисам внешних поставщиков идентификации по `443` (при их использовании):

Ссылки на сервисы внешних поставщиков

Тип	Ссылка
Социальные сети	`https://appleid.apple.com`
	`https://accounts.google.com`
	`https://graph.facebook.com`[2]
	`https://oauth.yandex.ru`
	`https://oauth.vk.com`
	`https://account.mail.ru`
	`https://api.ok.ru`
ЕСИА и цифровой профиль	`https://esia-portal1.test.gosuslugi.ru`
	`https://esia.gosuslugi.ru`
Банки	`https://online.sberbank.ru`
	`https://business.tinkoff.ru`
	`https://id.vtb.ru`
	`https://sbi.sberbank.ru:9443`
	`https://fintech.sberbank.ru:9443`
	`https://id-sandbox.alfabank.ru`
СУДИР	`https://login.mos.ru`
	`https://login-tech.mos.ru`
	`https://sudir.mos.ru`
	`https://sudir-test.mos.ru`

  - к SMS-шлюзу (при его использовании);
  - к SMTP (при его использовании);
  - к сервису push-уведомлений (при его использовании);
  - к сервису Kafka (при его использовании для приема событий безопасности).
- с VM-ADM должен быть доступ:
  - к VM-DB по `8091`, `8092`, `8093`, `11209`, `11210`, `11211`, `4369`, `21100 - 21199`, `11214`, `11215`, `18091`, `18092` (стандартные порты Couchbase Server) или `5432` (стандартный порт PostgreSQL);
  - к VM-LDAP (VM_NLB) по `389`, `636` (стандартные порты LDAP);
  - к VM-APP по `22` (ssh), `514` (rsyslog), `873` (rsync), `11211` (memcached);
  - к VM-MQ (VM-NLB) по `5672` (стандартный порт RabbitMQ);
  - к сервису Kafka (при его использовании для приема событий безопасности)
- с VM-DB должен быть доступ до других VM-DB по `8091`, `8092`, `8093`, `11209`, `11210`, `11211`, `4369`, `21100 - 21199`, `11214`, `11215`, `18091`, `18092` (порты Couchbase Server) или `5432` (порт PostgreSQL);
- с VM-LDAP должен быть доступ до других VM-LDAP по `389`, `636` (порты LDAP);
- с VM-MQ должен быть доступ до других VM-MQ по `4369`, `35197`, `5672`.

Для VM-APP нужно завести публичное DNS-имя (например, `auth.domain.ru`) и выпустить TLS-сертификат на `auth.domain.ru` или `*.domain.ru`.

[2] Meta признана экстремистской организацией и запрещена в России, деятельность принадлежащих ей соц.сетей Facebook и Instagram также запрещена в РФ.

2.1.3 Общая инструкция по установке

В общем случае установка Blitz Identity Provider выполняется в описанной ниже последовательности.

Совет: В зависимости от используемой операционной системы есть своя специфика по установке необходимого окружения. Для удобства воспользуйтесь экспресс-инструкциями для *отечественных* (страница 32) и *зарубежных* (страница 56) ОС.

Важно: Перед развертыванием ознакомьтесь с *архитектурой развертывания* (страница 11) Blitz Identity Provider.

Шаг 1. JDK

На серверах, предназначенных для установки ПО сервера Blitz Identity Provider и административной консоли Blitz Identity Provider, необходимо установить и настроить JDK 11, руководствуясь официальной справочной документацией, используя один из следующих продуктов:

- OpenJDK 11 Рекомендуется для сертифицированной версии;

  Примечание: Для установки OpenJDK 11 в CentOS и RHEL выполните команду:

  ```
  sudo yum install java-11-openjdk-devel
  ```

- Liberica JDK 11[5];
- Axiom JDK 11 Certified[6] Рекомендуется для сертифицированной версии;
- Oracle JDK 11[7].

Шаг 2. Memcached

Внимание: Версия `memcached` должна быть 1.4.15 или выше. Сервис memcached должен быть установлен на серверах, предназначенных для установки сервисов Blitz Identity Provider: `blitz-console`, `blitz-idp`, `blitz-registration`, `blitz-recovery`.

CentOS и RHEL

1. Выполнить команду:

   ```
   yum -y install memcached
   ```

2. После завершения установки добавить сервис `memcached` в автозапуск и запустить сервис:

   ```
   systemctl enable memcached
   systemctl start memcached
   ```

[5] https://docs.bell-sw.com/liberica-jdk/11.0.23b12/general/install-guide/
[6] https://axiomjdk.ru/pages/axiomjdk-install-guide-11.0.17/
[7] https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html

Astra Linux Special Edition 1.6

1. Выполнить команду:

```
apt-get install memcached
```

2. После завершения установки добавить сервис `memcached` в автозапуск и запустить сервис:

```
systemctl enable memcached
systemctl start memcached
```

Важно: Сервис `memcached` запускается на порту `11211`. Нужно убедиться, что этот порт открыт на межсетевых экранах и может быть использован для соединения между сервисами Blitz Identity Provider.

Шаг 3. СУБД

Установка Couchbase Server

Инструкция по установке Couchbase Server приводится для CentOS 7 и RHEL 7. В случае развертывания под отечественные операционные системы в качестве СУБД рекомендуется использовать PostgreSQL.

1. Установить Couchbase Server на каждый из выделенных под установку СУБД серверов согласно инструкции[10]. Дистрибутив Couchbase Server можно скачать здесь[11].

   Важно: В DEV/TEST-средах допустимо Couchbase Server устанавливать на существующие сервера с Blitz Identity Provider, но в этом случае нужно учесть, что в Couchbase Server используется своя встроенная memcached-служба, и во избежание конфликта необходимо скорректировать используемые Memcached порты в Blitz Identity Provider/Couchbase Server.

   ---

2. Добавить сервис Couchbase Server в автозапуск и запустить сервис:

```
systemctl enable couchbase-server
systemctl start couchbase-server
```

3. Проверить работоспособность сервиса, выполнив команду:

```
systemctl status couchbase-server
```

4. Инициализировать на каждом сервере кластер Couchbase Server согласно инструкции[12] (на первом сервере инициализируется кластер, остальные сервера включаются в кластер). Все настройки можно задать как предложено по умолчанию, только нужно для каждого сервера в `hostname` задать полное имя сервера. В качестве имени сервера не рекомендуется использовать его IP-адрес.

5. На любом из серверов кластера Couchbase Server выполнить скрипт по подготовке Couchbase Server к использованию Blitz Identity Provider. Скрипт находятся в директории `couchbase` в архиве `resources.zip` в составе дистрибутива Blitz Identity Provider. Скрипт нужно скопировать на любой сервер кластера Couchbase Server, перейти в директорию и выполнить скрипт создания `buckets` для хранения информации Blitz Identity Provider и индексов для выполнения поисковых запросов Blitz Identity Provider в БД:

```
./cb_init.sh
```

[10] https://docs.couchbase.com/server/current/install/install-linux.html
[11] https://www.couchbase.com/downloads
[12] https://docs.couchbase.com/server/current/manage/manage-nodes/initialize-node.html

В процессе выполнения скрипта понадобится ввести:

- имя URL сервера Couchbase Server – ввести строку вида `http://<hostname>:8091`, где в качестве hostname указать имя хоста сервера, с которого выполняется скрипт;
- логин учетной записи администратора Couchbase Server – задается при инициализации кластера при выполнении предыдущего пункта инструкции;
- пароль учетной записи администратора Couchbase Server – задается при инициализации кластера при выполнении предыдущего пункта инструкции;
- логин учетной записи Couchbase Server, которая создастся в процессе выполнения этого скрипта для подключения сервисов Blitz Identity Provider;

  Совет: Рекомендуется задать имя `blitz`.

- пароль учетной записи Couchbase Server для подключения приложений Blitz Identity Provider.

6. После выполнения скрипта произвести следующие настройки:
   1. В консоли администрирования Couchbase Server отредактировать настройки количества копий данных на различных экземплярах Couchbase. Для этого в меню Buckets поочередно выбрать каждый `bucket`, нажать на нем Edit и задать значение настройки Enable в блоке Replicas и установить число реплик. Для кластера из трех серверов рекомендуется задать в настройке значение `1` для числа реплик. Затем в меню Settings рекомендуется включить настройку Enable auto-failover и задать значение «Timeout» в `30` секунд (auto-failover будет работать, только если в кластере СУБД не менее 3 серверов и настроена репликация для `bucket`).
   2. Настроить резервное копирование БД[13].

Установка и настройка PostgreSQL

Внимание: Версия PostgreSQL должна быть 9.6 или новее.

CentOS и RHEL

Установить PostgreSQL согласно инструкции[14].

Astra Linux Special Edition 1.6:

1. Выполнить команду:

```
apt-get install postgresql
```

2. После завершения установки запустить сервис:

```
systemctl start postgresql
```

После завершения установки PostgreSQL в выбранной ОС необходимо выполнить скрипт по подготовке PostgreSQL к использованию Blitz Identity Provider. Скрипты находятся в директории `postgres` в архиве `resources.zip` в составе дистрибутива Blitz Identity Provider. Скрипты нужно скопировать на сервер PostgreSQL, перейти в директорию и по очереди выполнить команды:

[13] https://docs.couchbase.com/server/current/manage/manage-backup-and-restore/manage-backup-and-restore.html
[14] https://www.postgresql.org/download/linux/redhat/

```
su - postgres

createdb blitzdb

psql
CREATE USER blitz WITH ENCRYPTED PASSWORD 'set-your-pwd';
GRANT ALL PRIVILEGES ON DATABASE blitzdb TO blitz;
GRANT ALL ON ALL TABLES IN SCHEMA public TO blitz;

psql -d blitzdb -U blitz -f 000-SCRIPT000.sql
…
psql -d blitzdb -U blitz -f NNN-SCRIPTNNN.sql
```

Вместо `set-your-pwd` нужно вставить пароль, который будет использоваться для подключения к PostgreSQL.

Вместо `000-SCRIPT000.sql … NNN-SCRIPTNNN.sql` нужно вставить имена скриптов из директории `postgres/ddl` из архива `resources.zip`. Например:

```
psql -d blitzdb -U blitz -f 000-service-tasks.sql
psql -d blitzdb -U blitz -f 001-init-database.sql
psql -d blitzdb -U blitz -f 002-new_pp_columns.sql
psql -d blitzdb -U blitz -f 003-usd_id_table.sql
psql -d blitzdb -U blitz -f 004-usr_auth_table.sql
psql -d blitzdb -U blitz -f 005-usr_agt_table.sql
psql -d blitzdb -U blitz -f 006-usr_htp_hmc_alg.sql
psql -d blitzdb -U blitz -f 007-usr_atr_cfm.sql
psql -d blitzdb -U blitz -f 008-wak.sql
psql -d blitzdb -U blitz -f 009-fix_pp_column.sql
psql -d blitzdb -U blitz -f 010-add_usr_prp.sql
psql -d blitzdb -U blitz -f 011-pp_audit.sql
psql -d blitzdb -U blitz -f 012-geo_to_audit.sql
psql -d blitzdb -U blitz -f 013-tasks.sql
psql -d blitzdb -U blitz -f 014-sec_ch_ua.sql
psql -d blitzdb -U blitz -f 015-5.12.0.sql
psql -d blitzdb -U blitz -f 016-5.13.0.sql
psql -d blitzdb -U blitz -f 017-5.15.0.sql
psql -d blitzdb -U blitz -f 018-5.17.0.sql
psql -d blitzdb -U blitz -f 019-5.18.0.sql
psql -d blitzdb -U blitz -f 020-5.20.0.sql
psql -d blitzdb -U blitz -f 021-5.21.0.sql
psql -d blitzdb -U blitz -f 022-5.23.0.sql
psql -d blitzdb -U blitz -f 023-5.26.0.sql
```

После выполнения скрипта необходимо настроить резервное копирование БД[16].

Шаг 4. RabbitMQ

Опционально

Установка сервера очередей RabbitMQ опциональна и требуется, если сервер очередей будет использоваться для *передачи событий в смежные системы* (страница 323) или в качестве *брокера сообщений* (страница 324).

[16] https://postgrespro.ru/docs/postgresql/9.6/backup-dump#backup-dump-all

CentOS и RHEL

Установить RabbitMQ согласно инструкции[17].

Astra Linux Special Edition 1.6:

1. Выполнить команду:

```
apt-get install rabbitmq-server
```

2. После завершения установки запустить сервис:

```
systemctl start rabbitmq-server
```

Шаг 5. Blitz Identity Provider

Для установки сервисов `blitz-console`, `blitz-idp`, `blitz-registration`, `blitz-recovery` используется единый установщик `blitz-5.X.X.bin`.

При установке сертифицированной версии Blitz Identity Provider дополнительно используется файл `blitz-idp-thirdparty-5.X.X.tar.gz`, содержащий архив с используемыми Blitz Identity Provider сторонними библиотеками.

Важно: Консоль управления можно установить на любой сервер, где установлен сервер Blitz Identity Provider, но рекомендуется выделить под установку консоли управления отдельный административный сервер. На сервере предварительно должны быть установлены *JDK* (страница 18) и *memcached* (страница 18).

Для установки приложений `blitz-console`, `blitz-idp`, `blitz-registration`, `blitz-recovery` необходимо:

1. На предназначенные для установки сервера скопировать (например, в директорию `/tmp`) из дистрибутива Blitz Identity Provider файл `blitz-5.X.X.bin`.

   В случае установки сертифицированной версии необходимо также скопировать `blitz-idp-thirdparty-5.X.X.tar.gz`.

2. Запустить установщик `blitz-5.X.X.bin`, указав параметры запуска:

   - `-i` – список устанавливаемых приложений, разделенных через пробел (например, `idp console registration recovery`);
   - `-j` – значение `JAVA_HOME` – директория, в которую на сервере установлен JDK.

   Установка будет произведена в директорию `/usr/share/identityblitz`.

Список 1: Пример запуска установщика

```
cd /tmp
chmod +x blitz-5.X.X.bin
./blitz-5.X.X.bin -- -j /opt/oracle/jdk -i "idp console recovery registration"
```

[17] https://www.rabbitmq.com/install-rpm.html

Список 2: Пример вывода в консоль при работе установщика

```
Verifying archive integrity...  100%   MD5 checksums are OK. All good.
Uncompressing Blitz IDP  100%
**************************************************************
Application blitz-registration installed
Application blitz-recovery installed
Application blitz-console installed
Application blitz-idp installed
**************************************************************
```

3. Создать файл `blitz_param.txt`, в котором задать первичные настройки Blitz Identity Provider:

Couchbase Server

- `DOMAIN` – внешнее имя домена, на котором будет функционировать Blitz Identity Provider;
- `ROOT_CONTEXT` – URL-путь, на котором будет функционировать Blitz Identity Provider;

> **Примечание:** Если параметр не указывать, то по умолчанию будет задан `/blitz`.

- `ADMIN_USERNAME` – имя учетной записи администратора в Blitz Identity Provider;

> **Примечание:** Если параметр не указывать, то по умолчанию будет задан `admin`.

- `ADMIN_PASSWORD` – пароль от учетной записи администратора в Blitz Identity Provider;
- `KEYSTORE_PASSWORD` – пароль от создаваемого в процессе установки ключевого контейнера;

> **Примечание:** Если параметры `ADMIN_PASSWORD` и `KEYSTORE_PASSWORD` не указывать, то эти пароли будут автоматически сгенерированы и выведены в результатах работы скрипта конфигурации.

- `MEMCACHED_SERVERS` – адреса серверов с `memcached`;
- `DB_MODE` – используемая СУБД: `CB` для Couchbase Server;
- `CB_NODES` – адреса серверов с СУБД Couchbase Server;
- `CB_USERNAME` – имя учетной записи в СУБД Couchbase Server (по умолчанию `blitz`);
- `CB_PASSWORD` – пароль от учетной записи в СУБД Couchbase Server;
- `TRUSTED_SERVERS` – адреса подсетей серверов с сервисами Blitz Identity Provider (по умолчанию `127.0.0.1/32`).

Список 3: Пример конфигурационного файла

```
DOMAIN=test
MEMCACHED_SERVERS="192.168.122.10 127.0.0.1"
DB_MODE=CB
CB_NODES="192.168.122.20 192.168.122.21 192.168.122.22"
CB_USERNAME=blitz
CB_PASSWORD=12ABcd45
```

PostgreSQL

- DOMAIN – внешнее имя домена, на котором будет функционировать Blitz Identity Provider;
- ROOT_CONTEXT – URL-путь, на котором будет функционировать Blitz Identity Provider;

> **Примечание:** Если параметр не указывать, то по умолчанию будет задан /blitz.

- ADMIN_USERNAME – имя учетной записи администратора в Blitz Identity Provider;

> **Примечание:** Если параметр не указывать, то по умолчанию будет задан admin.

- ADMIN_PASSWORD – пароль от учетной записи администратора в Blitz Identity Provider;
- KEYSTORE_PASSWORD – пароль от создаваемого в процессе установки ключевого контейнера;

> **Примечание:** Если параметры ADMIN_PASSWORD и KEYSTORE_PASSWORD не указывать, то эти пароли будут автоматически сгенерированы и выведены в результатах работы скрипта конфигурации.

- MEMCACHED_SERVERS – адреса серверов с memcached;
- DB_MODE – используемая СУБД: PG для PostgreSQL;
- PG_HOSTNAME – адрес СУБД PostgreSQL;
- PG_DB_NAME – имя БД в СУБД PostgreSQL;

Совет: Рекомендуется использовать blitzdb.

- PG_USERNAME – имя учетной записи в СУБД PostgreSQL;

Совет: Рекомендуется использовать blitz.

- PG_PASSWORD – пароль от учетной записи в СУБД PostgreSQL;
- TRUSTED_SERVERS – адреса подсетей серверов с сервисами Blitz Identity Provider (по умолчанию 127.0.0.1/32).

Список 4: Пример конфигурационного файла

```
DOMAIN=test
ROOT_CONTEXT=/blitz
MEMCACHED_SERVERS="127.0.0.1 192.168.122.96"
DB_MODE=PG
PG_HOSTNAME=192.168.122.20
PG_DB_NAME=blitzdb
PG_USERNAME=blitz
PG_PASSWORD=123456
TRUSTED_SERVERS="127.0.0.1/32 192.168.122.96/32 192.168.122.0/24"
ADMIN_USERNAME=admin1
ADMIN_PASSWORD=0123456789
KEYSTORE_PASSWORD=0123456789
```

4. Запустить скрипт первоначальной настройки Blitz Identity Provider, указав путь к файлу blitz_param.txt:

```
/usr/share/identityblitz/blitz-console/bin/configure -f blitz_param.txt
```

В результате выполнения скрипта будут настроены конфигурационные файлы, а также будет сгенерирован и показан логин/пароль администратора Blitz Identity Provider и сгенерирован пароль от ключевого контейнера:

```
*****************************************************************
Your instance is configured on domain: test.loc
The Administration Console available on addresses:
 http://testinstallation.local:9001/blitz/console

Administration user credentials of Console:
  username - admin
  password - 98aAB0D3f2
Your can change user credentials at file - /usr/share/identityblitz/blitz-
↪config/credentials

Create keystore /usr/share/identityblitz/blitz-config/blitz-keystore.bks and␣
↪generate:
 - JWS(RSA256) keypair - jws_rs256_rsa_default
 - AES(AES128) security key - jdbc

Generated password for keystore: BeEBcd2239
*****************************************************************
```

Совет: Если при запуске установщика были допущены ошибки ввода, так что установка была проведена с неправильными параметрами, то можно воспользоваться следующей командой для удаления файлов, которые создал установщик, чтобы иметь возможность вновь провести установку начисто:

```
rm -rf /usr/share/identityblitz /etc/default/blitz-* /etc/blitz-* /var/log/
↪identityblitz/ /lib/systemd/system/blitz-*
```

---

5. Добавить сервисы в автозапуск на соответствующих им серверах и запустить их:

```
systemctl enable blitz-console
systemctl start blitz-console
systemctl enable blitz-idp
systemctl start blitz-idp
systemctl enable blitz-registration
systemctl start blitz-registration
systemctl enable blitz-recovery
systemctl start blitz-recovery
```

Шаг 6. Синхронизация файлов конфигурации

Только для установки в кластере

При развертывании Blitz Identity Provider в кластере необходимо настроить синхронизацию конфигурации между серверами кластера Blitz Identity Provider:

Действия на сервере с консолью управления

1. Установить `rsync` и `incron`:

```
sudo yum install rsync incron
```

   или (для Astra Linux Special Edition 1.6)

```
sudo apt install rsync incron
```

2. Переключиться в пользователя `blitz`:

```
sudo su - blitz
```

3. Сгенерировать ssh-ключ командой (на все задаваемые утилитой вопросы рекомендуется выбрать ответы по умолчанию):

```
ssh-keygen
```

4. Прочитать и сохранить для дальнейшего использования публичный ssh-ключ:

```
cat /usr/share/identityblitz/.ssh/id_rsa.pub
```

5. Открыть настройки `incrontab`:

```
incrontab -e
```

   В открывшемся окне редактора вставить следующее:

```
/usr/share/identityblitz/blitz-config IN_MODIFY,IN_ATTRIB,IN_CREATE,IN_DELETE,
↪IN_CLOSE_WRITE /usr/share/identityblitz/scripts/config_sync.sh ./ $# $%
/usr/share/identityblitz/blitz-config/assets IN_MODIFY,IN_ATTRIB,IN_CREATE,IN_
↪DELETE,IN_CLOSE_WRITE /usr/share/identityblitz/scripts/config_sync.sh assets
↪$# $%
/usr/share/identityblitz/blitz-config/assets/services IN_MODIFY,IN_ATTRIB,IN_
↪CREATE,IN_DELETE,IN_CLOSE_WRITE /usr/share/identityblitz/scripts/config_sync.
↪sh assets $# $%
/usr/share/identityblitz/blitz-config/assets/themes IN_MODIFY,IN_ATTRIB,IN_
↪CREATE,IN_DELETE,IN_CLOSE_WRITE /usr/share/identityblitz/scripts/config_sync.
↪sh assets $# $%
/usr/share/identityblitz/blitz-config/apps IN_MODIFY,IN_ATTRIB,IN_CREATE,IN_
↪DELETE,IN_CLOSE_WRITE /usr/share/identityblitz/scripts/config_sync.sh apps $
↪# $%
/usr/share/identityblitz/blitz-config/saml IN_MODIFY,IN_ATTRIB,IN_CREATE,IN_
↪DELETE,IN_CLOSE_WRITE /usr/share/identityblitz/scripts/config_sync.sh saml $
↪# $%
/usr/share/identityblitz/blitz-config/saml/conf IN_MODIFY,IN_ATTRIB,IN_CREATE,
↪IN_DELETE,IN_CLOSE_WRITE /usr/share/identityblitz/scripts/config_sync.sh␣
↪saml $# $%
/usr/share/identityblitz/blitz-config/saml/credentials IN_MODIFY,IN_ATTRIB,IN_
↪CREATE,IN_DELETE,IN_CLOSE_WRITE /usr/share/identityblitz/scripts/config_sync.
```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪sh saml $# $%
/usr/share/identityblitz/blitz-config/saml/metadata IN_MODIFY,IN_ATTRIB,IN_
↪CREATE,IN_DELETE,IN_CLOSE_WRITE /usr/share/identityblitz/scripts/config_sync.
↪sh saml $# $%
/usr/share/identityblitz/blitz-config/custom_messages IN_MODIFY,IN_ATTRIB,IN_
↪CREATE,IN_DELETE,IN_CLOSE_WRITE /usr/share/identityblitz/scripts/config_sync.
↪sh custom_messages $# $%
/usr/share/identityblitz/blitz-config/custom_messages/dics IN_MODIFY,IN_ATTRIB,
↪IN_CREATE,IN_DELETE,IN_CLOSE_WRITE /usr/share/identityblitz/scripts/config_
↪sync.sh custom_messages $# $%
/usr/share/identityblitz/blitz-config/devices IN_MODIFY,IN_ATTRIB,IN_CREATE,IN_
↪DELETE,IN_CLOSE_WRITE /usr/share/identityblitz/scripts/config_sync.sh␣
↪devices $# $%
/usr/share/identityblitz/blitz-config/simple IN_MODIFY,IN_ATTRIB,IN_CREATE,IN_
↪DELETE,IN_CLOSE_WRITE /usr/share/identityblitz/scripts/config_sync.sh simple
↪$# $%
/usr/share/identityblitz/blitz-config/certs IN_MODIFY,IN_ATTRIB,IN_CREATE,IN_
↪DELETE,IN_CLOSE_WRITE /usr/share/identityblitz/scripts/config_sync.sh certs $
↪# $%
/usr/share/identityblitz/blitz-config/flows/login IN_MODIFY,IN_ATTRIB,IN_
↪CREATE,IN_DELETE,IN_CLOSE_WRITE /usr/share/identityblitz/scripts/config_sync.
↪sh flows $# $%
/usr/share/identityblitz/blitz-config/flows/reg IN_MODIFY,IN_ATTRIB,IN_CREATE,
↪IN_DELETE,IN_CLOSE_WRITE /usr/share/identityblitz/scripts/config_sync.sh␣
↪flows $# $%
/usr/share/identityblitz/blitz-config/flows/extIdps IN_MODIFY,IN_ATTRIB,IN_
↪CREATE,IN_DELETE,IN_CLOSE_WRITE /usr/share/identityblitz/scripts/config_sync.
↪sh flows $# $%
/usr/share/identityblitz/blitz-config/token_exchange IN_MODIFY,IN_ATTRIB,IN_
↪CREATE,IN_DELETE,IN_CLOSE_WRITE /usr/share/identityblitz/scripts/config_sync.
↪sh token_exchange $# $%
/usr/share/identityblitz/blitz-config/token_exchange/rules IN_MODIFY,IN_ATTRIB,
↪IN_CREATE,IN_DELETE,IN_CLOSE_WRITE /usr/share/identityblitz/scripts/config_
↪sync.sh token_exchange $# $%
```

6. Создать файл `/usr/share/identityblitz/scripts/config_sync.sh` и вставить в него скрипт:

```
#!/bin/bash
app_dir=/usr/share/identityblitz/blitz-config
node_list="NODES_LIST"
for node in $(echo "${node_list}"); do
rsync -r -a --delete ${app_dir}/${1} ${USER}@${node}:${app_dir};
done
```

7. В качестве значения `node_list`, вместо `NODES_LIST`, необходимо прописать список `hostname` узлов кластера Blitz (кроме узла консоли управления Blitz Console). Вписывать значения нужно через пробел. Например:

```
node_list="app1.local app2.local"
```

8. Сделать файл `/usr/share/identityblitz/scripts/config_sync.sh` исполняемым:

```
chmod +x /usr/share/identityblitz/scripts/config_sync.sh
```

9. Запустить `incrontab`, выполнив под пользователем root команду:

```
systemctl enable incrond
systemctl start incrond
```

Действия на остальных серверах Blitz Identity Provider

1. Установить `rsync`:

```
sudo yum install rsync
```

   или (для Astra Linux Special Edition 1.6)

```
sudo apt install rsync
```

2. Переключиться в пользователя `blitz`:

```
sudo su - blitz
```

3. Выполнить следующий скрипт:

```
mkdir .ssh
touch .ssh/authorized_keys
chmod 700 .ssh
chmod 640 .ssh/authorized_keys
```

4. Открыть файл `.ssh/authorized_keys` любым редактором, например, vim, и вставить публичный ssh-ключ, полученный ранее на сервере консоли управления Blitz Console.

Шаг 7. Веб-сервер

В качестве веб-сервера рекомендуется использовать nginx или HAProxy.

nginx

Пример настроечного файла для nginx включен в дистрибутив Blitz Identity Provider – это файл `blitz-idp.conf` из директории `nginx` в архиве `resources.zip`. Нужно скорректировать следующие блоки настроек, после чего загрузить файл на сервер с nginx (каталог `/etc/nginx/conf.d`):

1. Скорректировать блок настроек балансировки:

```
upstream blitz-idp {
    server [BLITZ-IDP-NODE-01]:9000 max_fails=3 fail_timeout=120;
    server [BLITZ-IDP-NODE-02]:9000 max_fails=3 fail_timeout=120;
}
upstream blitz-reg {
    server [BLITZ-REG-NODE-01]:9002 max_fails=3 fail_timeout=120;
    server [BLITZ-REG-NODE-02]:9002 max_fails=3 fail_timeout=120;
}
upstream blitz-rec {
    server [BLITZ-REC-NODE-01]:9003 max_fails=3 fail_timeout=120;
    server [BLITZ-REC-NODE-02]:9003 max_fails=3 fail_timeout=120;
}
upstream blitz-console {
    server [BLITZ-CONSOLE-NODE-01]:9001 max_fails=3 fail_timeout=120;
}
```

Параметры имеют следующие назначения:

- `[BLITZ-%%%-NODE-XX]` – имена (`hostname`) серверов с сервисами Blitz Identity Provider (`blitz-idp`, `blitz-registration`, `blitz-recovery`);
- `[BLITZ-CONSOLE-NODE-01]` – имя (`hostname`) сервера с Blitz Console.

2. Скорректировать блок настроек терминирования TLS:

```
ssl_certificate          [BLITZ-SSL-CERT-FILE];
ssl_certificate_key      [BLITZ-SSL-PRIVATEKEY-FILE];
```

Параметры имеют следующие назначения:

- `[BLITZ-SSL-CERT-FILE]` – путь (полное имя) к файлу с TLS-сертификатом сервера;
- `[BLITZ-IDP-CONSOLE-NODE-01]` – путь (полное имя) к файлу с TLS-ключом сервера.

3. Следует учесть, что Blitz Identity Provider игнорирует заголовок `X-Forwarded-Proto https`, если в nginx `X-Forwarded-For` содержит более одного IP-адреса, например:

```
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
```

В этом случае рекомендуется использовать следующее значение директивы:

```
proxy_set_header X-Forwarded-For $client_ip
```

При этом `client_ip` вычисляется с помощью `map`. Из списка всегда всегда берется первое значение:

```
map $http_x_forwarded_for $client_ip {
    default $remote_addr;
    "~(?<IP>([0-9]{1,3}\.){3}[0-9]{1,3})*" $IP;
    "~(?<IP>([0-9]{1,3}\.){3}[0-9]{1,3}),.*" $IP;
}
```

4. Скопировать на сервер nginx в папку `/usr/share/nginx/html` папку `static_errors` с файлами страниц отображения ошибок сервера. Файлы с примерами оформления страниц ошибок можно взять в дистрибутиве Blitz Identity Provider – это папка `static_errors` в архиве `resources.zip`.

HAProxy

Пример настроечного файла для HAProxy включен в дистрибутив Blitz Identity Provider – это файл `haproxy.cfg` из директории `haproxy` в архиве `resources.zip`. Конфигурационный файл рассчитан на HAProxy версии 2.2+, поскольку в блоке используются директивы `http-errors`, `errorfile`, `errorfiles` для переопределения использования страниц ошибок из поставки Blitz Identity Provider.

Ограничение - статический контент, в отличие от nginx, HAProxy не поддерживаются. Изображения, favicon и т.п. необходимо подгружать с отдельного сервера. За это отвечает правило `acl is-blitz-static`, маршрут `use_backend blitz-static if is-blitz-static` и `backend blitz-static`.

Блок `global` - определяет системные параметры (пользователь, от имени которого выполняется процесс, параметры логирования, разрешенные алгоритмы шифрования и т.д.).

Секция `defaults` - определяет общие параметры для следующих за ним секций.

Секция `frontend` - определяет правила обработки входящих запросов.

Секция `backend` - описывает сервера которым будут перенаправлены запросы.

Основной интерес представляют секции backend с включенными активными проверками:

1. Запустить алгоритм балансировки:

```
balance roundrobin
```

2. Включить http-проверки backend:

```
option httpchk
```

3. Произведение начала сценария проверки.
4. Открыть новое соединение:

```
http-check connect
```

5. Отправить get-запрос к лк:

```
http-check send meth GET uri /blitz/profile hdr Host rocky8
```

6. Проверить установку сессионной cookie blc:

```
http-check expect hdr name set-cookie value -m beg "blc="
```

7. Проверить наличие редиректа:

```
http-check expect status 303
```

8. Появятся записи для каждого сервера приложений. Интервал проверки сервера составляет 5 секунд. В случае 2-х ошибок подряд пометить, как недоступный. После 1-ой успешной проверки пометить, как доступный.
9. Контролировать поток ошибок на уровне l7. Если возникнет более 10-ти ошибок, тогда пометить, как недоступный. При успешном health-check будет отмечен, как доступный.
10. Ввести в балансировку постепенно за 10с:

```
server blitz-idp-local 127.0.0.1:9000 check inter 5s rise 1 fall 2 observe␣
↪layer7 error-limit 10 on-error mark-down slowstart 10s
```

Шаг 8. LDAP-каталог

Опционально

См.также:

Список поддерживаемых каталогов (страница 11).

В случае необходимости развертывания нового LDAP-каталога рекомендуется в качестве LDAP-каталога использовать 389 Directory Server, который входит в состав ОС:

CentOS и RHEL

1. Выполнить команды установки:

```
yum install 389-ds-base 389-adminutil 389-admin 389-admin-console 389-console␣
↪389-ds-console
yum install xauth
```

2. Установить `limits` в соответствии с рекомендациями 389 Directory Server:

```
echo "fs.file-max = 64000" >> /etc/sysctl.conf
echo "* soft nofile 8192" >> /etc/security/limits.conf
echo "* hard nofile 8192" >> /etc/security/limits.conf
echo "ulimit -n 8192" >> /etc/profile
```

3. Инициализировать LDAP-каталог. Ответить на вопросы установщика:

```
setup-ds-admin.pl
```

4. После завершения установки добавить LDAP-каталог в автозапуск и запустить сервис:

```
systemctl enable dirsrv.target
systemctl start dirsrv.target
```

Astra Linux Special Edition 1.6

1. Выполнить команду установки и скрипт инициализации каталога:

```
apt-get install 389-ds-base
setup-ds
```

2. После завершения установки добавить LDAP-каталог в автозапуск и запустить сервис:

```
systemctl enable dirsrv.target
systemctl start dirsrv.target
```

После установки 389 Directory Server выполнить его настройку для подготовки использования совместно с Blitz Identity Provider. Для этого:

1. Скопировать на LDAP-сервер конфигурационные скрипты LDAP из состава дистрибутива Blitz Identity Provider (это папка `ldap` в архиве `resources.zip`).
2. Выполнить скрипт первоначальной настройки `ldap_init.sh` – скрипт создаст ветку `sub` для хранения пользователей, сервисного пользователя `reader`, настроит права доступа пользователя и его парольную политику (бессрочный пароль для сервисного пользователя), создаст класс `blitz-schema` с атрибутами `uid`, `mail`, `mobile`, `sn`, `name`:

```
chmod +x ldap_init.sh
./ldap_init.sh
```

3. Выполнить скрипт настройки TLS на сервере LDAP (скрипт создает копию текущей `NSS DB`, затем создает новую `NSS DB`, сертификаты и файл `pin.txt` для запуска сервера без ввода пароля):

```
chmod +x ldap_ssl.sh
./ldap_ssl.sh
```

4. После выполнения скрипта перезапустить LDAP-каталог:

```
systemctl restart dirsrv.target
```

5. Если требуется настроить и включить глобальные парольные политики в LDAP, то скорректировать и выполнить скрипт `ldap_pwdpolicy.sh`:

```
chmod +x ldap_pwdpolicy.sh
./ldap_pwdpolicy.sh
```

6. Если требуется создать дополнительные атрибуты:
    1. подготовить текстовый файл, в котором на каждой строке привести имя создаваемого атрибута (т.е. текстовый файл со столбцом создаваемых атрибутов);
    2. выполнить скрипт создания дополнительных атрибутов, ответить на его вопросы:

```
chmod +x ldap_add_attr.sh
./ldap_add_attr.sh
```

3. отредактировать текстовый файл по адресу `/etc/dirsrv/slapd-<название инстанса>/schema/99user.ldif`, добавить новые атрибуты в `objectclass` с именем `blitz-schema` в раздел `MAY`;
4. перезапустить LDAP-каталог, чтобы применить изменения схемы каталога:

```
systemctl restart dirsrv.target
```

2.1.4 Экспресс-инструкции для разных ОС

В данном разделе приведены экспресс-инструкции по установке Blitz Identity Provider в различных операционных системах.

Ограничения при использовании инструкций

Предупреждение: В экспресс-инструкциях по установке рассматривается минимальная конфигурация без обеспечения отказоустойчивости с размещением всех компонент на 1 виртуальной машине.

Важно: Перед выполнением работ необходимо обновить операционную систему до актуальных патчей.

Инструкции приводятся для случая наличия подключения виртуальной машины к сети интернет. В качестве доменного имени для установки в инструкциях используется имя `testinstallation.local` (его нужно скорректировать). В применяемых для настройки скриптах в качестве пароля используется строка `CHANGE_ME` (его нужно скорректировать). Все действия выполняются с привилегиями пользователя `root`.

Перед установкой на сервер в каталог `~/tmp/blitz` должны быть загружены и распакованы файлы дистрибутива Blitz Identity Provider (проверить правильность версии в `BLITZ_REL`):

```
export BLITZ_REL=5.18.0
mkdir -p ~/tmp/blitz
wget -q 'https://nc.idblitz.ru/nextcloud/index.php/s/3W48EBrNXf3R3WC/download?path=
↪%2F'$BLITZ_REL'&files=blitz-'$BLITZ_REL.bin -O ~/tmp/blitz/blitz-$BLITZ_REL.bin
wget -q 'https://nc.idblitz.ru/nextcloud/index.php/s/3W48EBrNXf3R3WC/download?path=
↪%2F'$BLITZ_REL'&files=resources.zip' -O ~/tmp/blitz/resources.zip
unzip ~/tmp/blitz/resources.zip -d ~/tmp/blitz
find ~/tmp/blitz -name *.sh -o -name *.bin|xargs chmod +x
```

Astra, Альт, ОСнова, Red OS, РОСА

Важно: См. *ограничения* (страница 32) при использовании экспресс-инструкций.

Список ОС, для которых приведены инструкции по установке, и их обозначение в данном разделе:

- Astra 1.7: Astra Linux Special Edition 1.7;
- Альт 8: Альт 8 СП Сервер;
- Альт 10: Альт Сервер 10;

- ОСнова: ОСнова 2.5.1;
- Red OS: Red OS 7.3, Red OS 8;
- РОСА: РОСА Хром 12.

Шаг 1. JDK

Astra 1.7

Установить дистрибутив Liberica JDK 11[19]. Также возможно использовать Axiom JDK 11 Certified[20].

Альт 8

Установить дистрибутив JDK:

```
apt-get install java-11-openjdk-devel
```

Альт 10

Установить дистрибутив JDK:

```
apt-get install java-11-openjdk-devel
```

ОСнова

Установить дистрибутив Liberica JDK 11[21]. Также возможно использовать Axiom JDK 11 Certified[22].

Red OS

Установить дистрибутив:

```
dnf install java-11-openjdk-devel
```

РОСА

Установить дистрибутив:

```
dnf install java-11-openjdk-devel rootcerts-java
```

[19] https://docs.bell-sw.com/liberica-jdk/11.0.23b12/general/install-guide/
[20] https://axiomjdk.ru/pages/axiomjdk-install-guide-11.0.5/
[21] https://docs.bell-sw.com/liberica-jdk/11.0.23b12/general/install-guide/
[22] https://axiomjdk.ru/pages/axiomjdk-install-guide-11.0.5/

Шаг 2. Memcached

Astra 1.7

Установить дистрибутив:

```
apt install memcached
```

Запустить службу:

```
systemctl enable memcached && systemctl start memcached
```

```
apt-get install java-1.8.0-openjdk-devel
```

Альт 8

Установить дистрибутив:

```
apt-get install memcached
```

Запустить службу:

```
systemctl enable memcached && systemctl start memcached
```

Альт 10

Установить дистрибутив:

```
apt-get install memcached
```

Запустить службу:

```
systemctl enable memcached && systemctl start memcached
```

ОСнова

Установить дистрибутив:

```
apt install memcached
```

Запустить службу:

```
systemctl enable memcached && systemctl start memcached
```

Red OS

Установить дистрибутив:

```
dnf install memcached
```

Запустить службу:

```
systemctl enable memcached && systemctl start memcached
```

POCA

Установить дистрибутив:

```
dnf install memcached
```

Запустить службу:

```
systemctl enable memcached && systemctl start memcached
```

Шаг 3. PostgreSQL

Astra 1.7

Установить дистрибутив:

```
apt install postgresql-11
```

Добавить разрешение в `/etc/postgresql/11/main/pg_hba.conf` на подключение к СУБД пользователю `blitz`:

```
host blitzdb blitz 127.0.0.1/32 scram-sha-256
```

Указать алгоритм шифрования паролей в `/etc/postgresql/11/main/postgresql.conf`:

```
password_encryption = scram-sha-256
```

Перезапустить службу:

```
systemctl restart postgresql@11-main
```

Подключиться к СУБД и провести первичную настройку

```
su - postgres
psql
create database blitzdb;
create user blitz with encrypted password 'CHANGE_ME';
grant ALL PRIVILEGES ON DATABASE blitzdb to blitz;
grant ALL on ALL tables in schema public to blitz;
```

Вернуться в shell пользователя `root` и выполнить скрипты создания и обновления структуры БД `blitzdb`:

```
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/000-service-tasks.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/001-init-database.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/002-new_pp_columns.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/003-usd_id_table.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/004-usr_auth_table.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/005-usr_agt_table.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/006-usr_htp_hmc_alg.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/007-usr_atr_cfm.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/008-wak.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/009-fix_pp_column.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/010-add_usr_prp.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/011-pp_audit.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/012-geo_to_audit.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/013-tasks.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/014-sec_ch_ua.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/015-5.12.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/016-5.13.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/017-5.15.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/018-5.17.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/019-5.18.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/020-5.20.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/021-5.21.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/022-5.23.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/023-5.26.0.sql
```

Альт 8

Установить дистрибутив:

```
apt-get install postgresql11-server
```

Инициализировать СУБД командой:

```
/etc/init.d/postgresql initdb
```

Добавить разрешение в `/etc/postgresql/11/main/pg_hba.conf` на подключение к СУБД пользователю `blitz`:

```
host blitzdb blitz 127.0.0.1/32 scram-sha-256
```

Указать алгоритм шифрования паролей в `/etc/postgresql/11/main/postgresql.conf`:

```
password_encryption = scram-sha-256
```

Запустить службу:

```
systemctl enable postgresql && systemctl start postgresql
```

Подключиться к СУБД и провести первичную настройку

```
psql -U postgres

create database blitzdb;
create user blitz with encrypted password 'CHANGE_ME';
grant ALL PRIVILEGES ON DATABASE blitzdb to blitz;
grant ALL on ALL tables in schema public to blitz;
```

Вернуться в shell и выполнить скрипты создания и обновления структуры БД `blitzdb`:

```
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/000-service-tasks.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/001-init-database.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/002-new_pp_columns.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/003-usd_id_table.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/004-usr_auth_table.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/005-usr_agt_table.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/006-usr_htp_hmc_alg.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/007-usr_atr_cfm.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/008-wak.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/009-fix_pp_column.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/010-add_usr_prp.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/011-pp_audit.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/012-geo_to_audit.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/013-tasks.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/014-sec_ch_ua.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/015-5.12.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/016-5.13.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/017-5.15.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/018-5.17.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/019-5.18.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/020-5.20.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/021-5.21.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/022-5.23.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/023-5.26.0.sql
```

Альт 10

Установить дистрибутив:

```
apt-get install postgresql14-server
```

Инициализировать СУБД командой:

```
/etc/init.d/postgresql initdb
```

Запустить службу:

```
systemctl enable postgresql && systemctl start postgresql
```

Подключиться к СУБД и провести первичную настройку

```
psql -U postgres

create database blitzdb;
create user blitz with encrypted password 'CHANGE_ME';
grant ALL PRIVILEGES ON DATABASE blitzdb to blitz;
grant ALL on ALL tables in schema public to blitz;
```

Вернуться в shell и выполнить скрипты создания и обновления структуры БД `blitzdb`:

```
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/000-service-tasks.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/001-init-database.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/002-new_pp_columns.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/003-usd_id_table.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/004-usr_auth_table.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/005-usr_agt_table.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/006-usr_htp_hmc_alg.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/007-usr_atr_cfm.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/008-wak.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/009-fix_pp_column.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/010-add_usr_prp.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/011-pp_audit.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/012-geo_to_audit.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/013-tasks.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/014-sec_ch_ua.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/015-5.12.0.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/016-5.13.0.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/017-5.15.0.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/018-5.17.0.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/019-5.18.0.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/020-5.20.0.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/021-5.21.0.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/022-5.23.0.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/023-5.26.0.sql
```

ОСнова

Установить дистрибутив:

```
apt install postgresql-11 postgresql-client-11
```

Инициализировать СУБД командой:

```
/etc/init.d/postgresql initdb
```

Добавить разрешение в `/etc/postgresql/11/main/pg_hba.conf` на подключение к БД пользователю `blitz`:

```
host blitzdb blitz 127.0.0.1/32 scram-sha-256
```

Указать алгоритм шифрования паролей в `/etc/postgresql/11/main/postgresql.conf`:

```
password_encryption = scram-sha-256
```

Перезапустить службу:

```
systemctl restart postgresql
```

Подключиться к СУБД и провести первичную настройку

```
su - postgres
psql

create database blitzdb;
create user blitz with encrypted password 'CHANGE_ME';
grant ALL PRIVILEGES ON DATABASE blitzdb to blitz;
grant ALL on ALL tables in schema public to blitz;
```

Вернуться в shell пользователя `root` и выполнить скрипты создания и обновления структуры БД `blitzdb`:

```
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/000-service-tasks.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/001-init-database.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/002-new_pp_columns.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/003-usd_id_table.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/004-usr_auth_table.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/005-usr_agt_table.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/006-usr_htp_hmc_alg.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/007-usr_atr_cfm.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/008-wak.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/009-fix_pp_column.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/010-add_usr_prp.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/011-pp_audit.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/012-geo_to_audit.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/013-tasks.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/014-sec_ch_ua.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/015-5.12.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/016-5.13.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/017-5.15.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/018-5.17.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/019-5.18.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/020-5.20.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/021-5.21.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/022-5.23.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/023-5.26.0.sql
```

Red OS

Установить дистрибутив:

```
dnf install postgresql14-server
```

Инициализировать СУБД командой:

```
/usr/bin/postgresql-14-setup initdb
```

Добавить разрешение в `/var/lib/pgsql/14/data/pg_hba.conf` на подключение к БД пользователю `blitz`:

```
host blitzdb blitz 127.0.0.1/32 scram-sha-256
```

Указать алгоритм шифрования паролей в `/var/lib/pgsql/14/data/postgresql.conf`:

```
password_encryption = scram-sha-256
```

Запустить службу:

```
systemctl enable postgresql-14 && systemctl start postgresql-14
```

Подключиться к СУБД и провести первичную настройку

```
su - postgres
psql

create database blitzdb;
create user blitz with encrypted password 'CHANGE_ME';
grant ALL PRIVILEGES ON DATABASE blitzdb to blitz;
grant ALL on ALL tables in schema public to blitz;
```

Вернуться в shell пользователя `root` и выполнить скрипты создания и обновления структуры БД `blitzdb`:

```
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/000-service-tasks.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/001-init-database.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/002-new_pp_columns.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/003-usd_id_table.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/004-usr_auth_table.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/005-usr_agt_table.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/006-usr_htp_hmc_alg.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/007-usr_atr_cfm.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/008-wak.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/009-fix_pp_column.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/010-add_usr_prp.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/011-pp_audit.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/012-geo_to_audit.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/013-tasks.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/014-sec_ch_ua.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/015-5.12.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/016-5.13.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/017-5.15.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/018-5.17.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/019-5.18.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/020-5.20.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/021-5.21.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/022-5.23.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/023-5.26.0.sql
```

РОСА

Установить дистрибутив:

```
dnf install postgresql14-server
```

Инициализировать СУБД командой:

```
/usr/bin/initdb --pgdata=/var/lib/pgsql/data
```

Добавить разрешение в `/var/lib/pgsql/data/pg_hba.conf` на подключение к БД пользователю `blitz`:

```
host    blitzdb    blitz    127.0.0.1/32    scram-sha-256
```

Указать алгоритм шифрования паролей в `/var/lib/pgsql/data/postgresql.conf`:

```
password_encryption = scram-sha-256
```

Запустить службу:

```
systemctl enable postgresql14.service && systemctl start postgresql14.service
```

Подключиться к СУБД и провести первичную настройку:

```
su - postgres
psql

create database blitzdb;
create user blitz with encrypted password 'CHANGE_ME';
grant ALL PRIVILEGES ON DATABASE blitzdb to blitz;
grant ALL on ALL tables in schema public to blitz;
```

Выполнить скрипты создания и обновления структуры БД `blitzdb`:

```
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/000-service-tasks.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/001-init-database.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/002-new_pp_columns.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/003-usd_id_table.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/004-usr_auth_table.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/005-usr_agt_table.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/006-usr_htp_hmc_alg.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/007-usr_atr_cfm.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/008-wak.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/009-fix_pp_column.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/010-add_usr_prp.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/011-pp_audit.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/012-geo_to_audit.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/013-tasks.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/014-sec_ch_ua.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/015-5.12.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/016-5.13.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/017-5.15.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/018-5.17.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/019-5.18.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/020-5.20.0.sql
```

(continues on next page)

(продолжение с предыдущей страницы)

```
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/021-5.21.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/022-5.23.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/023-5.26.0.sql
```

Шаг 4. RabbitMQ

Astra 1.7

Установить дистрибутив:

```
apt-get install rabbitmq-server
```

Запустить службу:

```
systemctl enable rabbitmq-server && systemctl start rabbitmq-server
```

Подготовить очередь для взаимодействия:

```
rabbitmqctl add_user blitz CHANGE_ME
rabbitmqctl set_permissions blitz ".*" ".*" ".*"
rabbitmq-plugins enable rabbitmq_management
curl -vvk 127.0.0.1:15672/cli/rabbitmqadmin >rabbitmqadmin
chmod +x rabbitmqadmin
./rabbitmqadmin declare exchange name=blitz-tasks-exh type=direct
./rabbitmqadmin declare queue name=blitz-tasks durable=true
./rabbitmqadmin declare binding source="blitz-tasks-exh"
destination_type="queue" destination="blitz-tasks"
routing_key="blitz-tasks"
```

Альт 8

Установить дистрибутив:

```
apt-get install rabbitmq-server
```

Запустить службу:

```
systemctl enable rabbitmq && systemctl start rabbitmq
```

Подготовить очередь для взаимодействия:

```
rabbitmqctl add_user blitz CHANGE_ME
rabbitmqctl set_permissions blitz ".*" ".*" ".*"
rabbitmq-plugins enable rabbitmq_management
curl -vvk 127.0.0.1:15672/cli/rabbitmqadmin >rabbitmqadmin
chmod +x rabbitmqadmin
./rabbitmqadmin declare exchange name=blitz-tasks-exh type=direct
./rabbitmqadmin declare queue name=blitz-tasks durable=true
./rabbitmqadmin declare binding source="blitz-tasks-exh"
destination_type="queue" destination="blitz-tasks"
routing_key="blitz-tasks"
```

Альт 10

Установить дистрибутив:

```
apt-get install rabbitmq-server
```

Запустить службу:

```
systemctl enable rabbitmq && systemctl start rabbitmq
```

Подготовить очередь для взаимодействия:

```
rabbitmqctl add_user blitz CHANGE_ME
rabbitmqctl set_permissions blitz ".*" ".*" ".*"
rabbitmq-plugins enable rabbitmq_management
curl -vvk 127.0.0.1:15672/cli/rabbitmqadmin >rabbitmqadmin
chmod +x rabbitmqadmin
./rabbitmqadmin declare exchange name=blitz-tasks-exh type=direct
./rabbitmqadmin declare queue name=blitz-tasks durable=true
./rabbitmqadmin declare binding source="blitz-tasks-exh"
destination_type="queue" destination="blitz-tasks"
routing_key="blitz-tasks"
```

ОСнова

Установить дистрибутив:

```
apt-get install rabbitmq-server
```

Запустить службу:

```
systemctl enable rabbitmq && systemctl start rabbitmq
```

Подготовить очередь для взаимодействия:

```
rabbitmqctl add_user blitz CHANGE_ME
rabbitmqctl set_permissions blitz ".*" ".*" ".*"
rabbitmq-plugins enable rabbitmq_management
curl -vvk 127.0.0.1:15672/cli/rabbitmqadmin >rabbitmqadmin
chmod +x rabbitmqadmin
./rabbitmqadmin declare exchange name=blitz-tasks-exh type=direct
./rabbitmqadmin declare queue name=blitz-tasks durable=true
./rabbitmqadmin declare binding source="blitz-tasks-exh"
destination_type="queue" destination="blitz-tasks"
routing_key="blitz-tasks"
```

Red OS

Установить дистрибутив:

```
dnf install rabbitmq-server
```

Запустить службу:

```
systemctl enable rabbitmq-server && systemctl start rabbitmq-server
```

Подготовить очередь для взаимодействия:

```
rabbitmqctl add_user blitz CHANGE_ME
rabbitmqctl set_permissions blitz ".*" ".*" ".*"
rabbitmq-plugins enable rabbitmq_management
curl -vvk 127.0.0.1:15672/cli/rabbitmqadmin >rabbitmqadmin
chmod +x rabbitmqadmin
./rabbitmqadmin declare exchange name=blitz-tasks-exh type=direct
./rabbitmqadmin declare queue name=blitz-tasks durable=true
./rabbitmqadmin declare binding source="blitz-tasks-exh"
destination_type="queue" destination="blitz-tasks"
routing_key="blitz-tasks"
```

Шаг 5. 389 Directory Server

Astra 1.7

Установить дистрибутив:

```
apt-get install 389-ds-base
```

Включить автоматический запуск службы:

```
systemctl enable dirsrv.target
```

Инициализировать LDAP-каталог:

```
dscreate interactive
```

Выполнить первичную настройку каталога:

```
/tmp/blitz/ldap/ldap_init.sh
```

Альт 8

Установить дистрибутив:

```
apt-get install 389-ds-base openldap-clients
```

Включить автоматический запуск службы:

```
systemctl enable dirsrv.target
```

Инициализировать LDAP-каталог:

```
dscreate interactive
```

Выполнить первичную настройку каталога:

```
/tmp/blitz/ldap/ldap_init.sh
```

Альт 10

Установить дистрибутив:

```
apt-get install 389-ds-base
apt-get install openldap-clients
```

Включить автоматический запуск службы:

```
systemctl enable dirsrv.target
```

Инициализировать LDAP-каталог:

```
dscreate interactive
```

Выполнить первичную настройку каталога:

```
/tmp/blitz/ldap/ldap_init.sh
```

ОСнова

Установить дистрибутив:

```
apt-get install 389-ds-base openldap-clients
```

Включить автоматический запуск службы:

```
systemctl enable dirsrv.target
```

Инициализировать LDAP-каталог:

```
dscreate interactive
```

Выполнить первичную настройку каталога:

```
/tmp/blitz/ldap/ldap_init.sh
```

Red OS

Установить дистрибутив:

```
dnf install 389-ds-base
```

Включить автоматический запуск службы:

```
systemctl enable dirsrv.target
```

Инициализировать LDAP-каталог:

```
dscreate interactive
```

Выполнить первичную настройку каталога:

```
/tmp/blitz/ldap/ldap_init.sh
```

POCA

Установить дистрибутив:

```
dnf install 389-ds-base openldap-clients
```

Включить автоматический запуск службы:

```
systemctl enable dirsrv.target
```

Инициализировать LDAP-каталог:

```
dscreate interactive
```

Выполнить первичную настройку каталога:

```
/tmp/blitz/ldap/ldap_init.sh
```

Шаг 6. Nginx

Astra 1.7

Установить дистрибутив:

```
apt-get install nginx-light
```

Скопировать файлы для использования:

```
cp /tmp/blitz/nginx/blitz-idp.conf /etc/nginx/sites-enabled/
cp -R /tmp/blitz/static_errors /usr/share/nginx/html
```

Альт 8

Установить дистрибутив:

```
apt-get install nginx
```

Скопировать файлы для использования:

```
mkdir -p /var/www/html
cp /tmp/blitz/nginx/blitz-idp.conf /etc/nginx/sites-enabled.d/
cp -R /tmp/blitz/static_errors /var/www/html
```

Добавить директиву в `/etc/nginx/sites-enabled.d/blitz-idp.conf`:

```
location /static_errors {
    root /var/www/html;
}
```

Включить автоматический запуск службы:

```
systemctl enable nginx
```

Альт 10

Установить дистрибутив:

```
apt-get install nginx
```

Создать каталог для размещения страниц с ошибками:

```
mkdir -p /var/www/html
```

Скопировать файлы для использования:

```
cp /tmp/blitz/nginx/blitz-idp.conf /etc/nginx/sites-enabled.d/
cp -R /tmp/blitz/static_errors /var/www/html
```

Добавить директиву в `/etc/nginx/sites-enabled.d/blitz-idp.conf`:

```
location /static_errors {
    root /var/www/html;
}
```

Включить автоматический запуск службы:

```
systemctl enable nginx && systemctl start nginx
```

ОСнова

Установить дистрибутив:

```
apt-get install nginx
```

Скопировать файлы для использования:

```
mkdir -p /var/www/html
cp /tmp/blitz/nginx/blitz-idp.conf /etc/nginx/sites-enabled.d/
cp -R /tmp/blitz/static_errors /var/www/html
```

Добавить директиву в `/etc/nginx/sites-enabled.d/blitz-idp.conf`:

```
location /static_errors {
    root /var/www/html;
}
```

Включить автоматический запуск службы:

```
systemctl enable nginx
```

Red OS

Установить дистрибутив:

```
dnf install nginx
```

Скопировать файлы для использования:

```
cp /tmp/blitz/nginx/blitz-idp.conf /etc/nginx/sites-enabled.d/
cp -R /tmp/blitz/static_errors /usr/share/nginx/html
```

Включить автоматический запуск службы:

```
systemctl enable nginx
```

РОСА

Установить дистрибутив:

```
dnf install nginx
```

Создать каталог для конфигурационных файлов:

```
mkdir /etc/nginx/conf.d
```

Добавить в `/etc/nginx/nginx.conf` путь к каталогу с конфигурационными файлами:

```
include /etc/nginx/conf.d/*.conf;
```

Скопировать файлы для использования:

```
cp /tmp/blitz/nginx/blitz-idp.conf /etc/nginx/conf.d
cp -R /tmp/blitz/static_errors /usr/share/nginx/html
```

Включить автоматический запуск службы:

```
systemctl enable nginx.service
```

Шаг 7. Blitz Identity Provider

Astra 1.7

Установить дистрибутив (подставить в имя файла правильную версию, JAVA_HOME и состав устанавливаемых приложений):

```
/tmp/blitz/blitz-5.X.X.bin -- -j <JAVA_HOME> -i "idp console recovery registration"
```

Создать конфигурационный файл blitz_param.txt следующего содержания, скорректировав в нем настройки на требуемые:

```
DOMAIN=testinstallation.local
MEMCACHED_SERVERS="127.0.0.1"
DB_MODE=PG
PG_HOSTNAME=127.0.0.1
PG_DB_NAME=blitzdb
PG_USER_NAME=blitz
PG_USER_PASSWORD=12ABcd45
```

Запустить скрипт первоначальной настройки Blitz Identity Provider, указав путь к файлу blitz_param.txt:

```
/usr/share/identityblitz/blitz-console/bin/configure -f blitz_param.txt
```

В результате выполнения скрипта будут настроены конфигурационные файлы, а также будет сгенерирован и показан логин/пароль администратора Blitz Identity Provider и сгенерирован пароль от ключевого контейнера:

```
*************************************************************
Your instance is configured on domain: test.loc
The Administration Console available on addresses:
  http://testinstallation.local:9001/blitz/console

Administration user credentials of Blitz Console:
  username - admin
  password - 98aAB0D3f2
Your can change user credentials at file - /usr/share/identityblitz/blitz-config/
↪credentials

Create keystore /usr/share/identityblitz/blitz-config/blitz-keystore.bks and␣
↪generate:
 - JWS(RSA256) keypair - jws_rs256_rsa_default
 - AES(AES128) security key - jdbc

Generated password for keystore: BeEBcd2239
*************************************************************
```

В случае использования ключей, созданных на этапе установки, перезапустить nginx:

```
systemctl restart nginx
```

Добавить сопоставление адреса loopback-интерфейса и доменного имени, указанного при установке в `/etc/hosts`:

```
127.0.0.1 localhost.localdomain localhost testinstallation.local
```

Запустить службы:

```
systemctl enable blitz-idp && systemctl start blitz-idp
systemctl enable blitz-console && systemctl start blitz-console
systemctl enable blitz-registration && systemctl start blitz-registration
systemctl enable blitz-recovery && systemctl start blitz-recovery
```

После успешного завершения установки и настройки Blitz Identity Provider возможно подключиться к консоли управления по доменному имени, указанному на этапе установки дистрибутива, например, `https://testinstallation.local/blitz/console`.

Альт 8

Установить дистрибутив (подставить в имя файла правильную версию, `JAVA_HOME` и состав устанавливаемых приложений):

```
/tmp/blitz/blitz-5.X.X.bin -- -j <JAVA_HOME> -i "idp console recovery registration"
```

Создать конфигурационный файл `blitz_param.txt` следующего содержания, скорректировав в нем настройки на требуемые:

```
DOMAIN=testinstallation.local
MEMCACHED_SERVERS="127.0.0.1"
DB_MODE=PG
PG_HOSTNAME=127.0.0.1
PG_DB_NAME=blitzdb
PG_USER_NAME=blitz
PG_USER_PASSWORD=12ABcd45
```

Запустить скрипт первоначальной настройки Blitz Identity Provider, указав путь к файлу `blitz_param.txt`:

```
/usr/share/identityblitz/blitz-console/bin/configure -f blitz_param.txt
```

В результате выполнения скрипта будут настроены конфигурационные файлы, а также будет сгенерирован и показан логин/пароль администратора Blitz Identity Provider и сгенерирован пароль от ключевого контейнера:

```
****************************************************************
Your instance is configured on domain: test.loc
The Administration Console available on addresses:
  http://testinstallation.local:9001/blitz/console

Administration user credentials of Blitz Console:
  username - admin
  password - 98aAB0D3f2
Your can change user credentials at file - /usr/share/identityblitz/blitz-config/
↪credentials
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Create keystore /usr/share/identityblitz/blitz-config/blitz-keystore.bks and␣
↪generate:
 - JWS(RSA256) keypair - jws_rs256_rsa_default
 - AES(AES128) security key - jdbc

Generated password for keystore: BeEBcd2239
****************************************************************
```

В случае использования ключей, созданных на этапе установки, запустить nginx:

```
systemctl start nginx
```

Добавить сопоставление адреса loopback-интерфейса и доменного имени, указанного при установке в `/etc/hosts`:

```
127.0.0.1 localhost.localdomain localhost testinstallation.local
```

Запустить службы:

```
systemctl enable blitz-idp && systemctl start blitz-idp
systemctl enable blitz-console && systemctl start blitz-console
systemctl enable blitz-registration && systemctl start blitz-registration
systemctl enable blitz-recovery && systemctl start blitz-recovery
```

После успешного завершения установки и настройки Blitz Identity Provider возможно подключиться к консоли управления по доменному имени, указанному на этапе установки дистрибутива, например, `https://testinstallation.local/blitz/console`.

Альт 10

Установить дистрибутив (подставить в имя файла правильную версию, `JAVA_HOME` и состав устанавливаемых приложений):

```
/tmp/blitz/blitz-5.X.X.bin -- -j <JAVA_HOME> -i "idp console recovery registration"
```

Создать конфигурационный файл `blitz_param.txt` следующего содержания, скорректировав в нем настройки на требуемые:

```
DOMAIN=testinstallation.local
MEMCACHED_SERVERS="127.0.0.1"
DB_MODE=PG
PG_HOSTNAME=127.0.0.1
PG_DB_NAME=blitzdb
PG_USER_NAME=blitz
PG_USER_PASSWORD=12ABcd45
```

Запустить скрипт первоначальной настройки Blitz Identity Provider, указав путь к файлу `blitz_param.txt`:

```
/usr/share/identityblitz/blitz-console/bin/configure -f blitz_param.txt
```

В результате выполнения скрипта будут настроены конфигурационные файлы, а также будет сгенерирован и показан логин/пароль администратора Blitz Identity Provider и сгенерирован пароль от ключевого контейнера:

```
****************************************************************
Your instance is configured on domain: test.loc
The Administration Console available on addresses:
  http://testinstallation.local:9001/blitz/console

Administration user credentials of Blitz Console:
  username - admin
  password - 98aAB0D3f2
Your can change user credentials at file - /usr/share/identityblitz/blitz-config/
↪credentials

Create keystore /usr/share/identityblitz/blitz-config/blitz-keystore.bks and␣
↪generate:
 - JWS(RSA256) keypair - jws_rs256_rsa_default
 - AES(AES128) security key - jdbc

Generated password for keystore: BeEBcd2239
****************************************************************
```

В случае использования ключей, созданных на этапе установки, перезапустить nginx:

```
systemctl restart nginx
```

Добавить сопоставление адреса loopback-интерфейса и доменного имени, указанного при установке в `/etc/hosts`:

```
127.0.0.1 localhost.localdomain localhost testinstallation.local
```

Запустить службы:

```
systemctl enable blitz-idp && systemctl start blitz-idp
systemctl enable blitz-console && systemctl start blitz-console
systemctl enable blitz-registration && systemctl start blitz-registration
systemctl enable blitz-recovery && systemctl start blitz-recovery
```

После успешного завершения установки и настройки Blitz Identity Provider возможно подключиться к консоли управления по доменному имени, указанному на этапе установки дистрибутива, например, `https://testinstallation.local/blitz/console`.

ОСнова

Установить дистрибутив (подставить в имя файла правильную версию, `JAVA_HOME` и состав устанавливаемых приложений):

```
/tmp/blitz/blitz-5.X.X.bin -- -j <JAVA_HOME> -i "idp console recovery registration"
```

Создать конфигурационный файл `blitz_param.txt` следующего содержания, скорректировав в нем настройки на требуемые:

```
DOMAIN=testinstallation.local
MEMCACHED_SERVERS="127.0.0.1"
DB_MODE=PG
PG_HOSTNAME=127.0.0.1
PG_DB_NAME=blitzdb
PG_USER_NAME=blitz
PG_USER_PASSWORD=12ABcd45
```

Запустить скрипт первоначальной настройки Blitz Identity Provider, указав путь к файлу `blitz_param.txt`:

```
/usr/share/identityblitz/blitz-console/bin/configure -f blitz_param.txt
```

В результате выполнения скрипта будут настроены конфигурационные файлы, а также будет сгенерирован и показан логин/пароль администратора Blitz Identity Provider и сгенерирован пароль от ключевого контейнера:

```
*****************************************************************
Your instance is configured on domain: test.loc
The Administration Console available on addresses:
  http://testinstallation.local:9001/blitz/console

Administration user credentials of Blitz Console:
  username - admin
  password - 98aAB0D3f2
Your can change user credentials at file - /usr/share/identityblitz/blitz-config/
↪credentials

Create keystore /usr/share/identityblitz/blitz-config/blitz-keystore.bks and␣
↪generate:
 - JWS(RSA256) keypair - jws_rs256_rsa_default
 - AES(AES128) security key - jdbc

Generated password for keystore: BeEBcd2239
*****************************************************************
```

В случае использования ключей, созданных на этапе установки, перезапустить nginx:

```
systemctl restart nginx
```

Добавить сопоставление адреса loopback-интерфейса и доменного имени, указанного при установке в `/etc/hosts`:

```
127.0.0.1 localhost.localdomain localhost testinstallation.local
```

Запустить службы:

```
systemctl enable blitz-idp && systemctl start blitz-idp
systemctl enable blitz-console && systemctl start blitz-console
systemctl enable blitz-registration && systemctl start blitz-registration
systemctl enable blitz-recovery && systemctl start blitz-recovery
```

После успешного завершения установки и настройки Blitz Identity Provider возможно подключиться к консоли управления по доменному имени, указанному на этапе установки дистрибутива, например, `https://testinstallation.local/blitz/console`.

Red OS

Установить дистрибутив (подставить в имя файла правильную версию, JAVA_HOME и состав устанавливаемых приложений):

```
/tmp/blitz/blitz-5.X.X.bin -- -j <JAVA_HOME> -i "idp console recovery registration"
```

Создать конфигурационный файл blitz_param.txt следующего содержания, скорректировав в нем настройки на требуемые:

```
DOMAIN=testinstallation.local
MEMCACHED_SERVERS="127.0.0.1"
DB_MODE=PG
PG_HOSTNAME=127.0.0.1
PG_DB_NAME=blitzdb
PG_USER_NAME=blitz
PG_USER_PASSWORD=12ABcd45
```

Запустить скрипт первоначальной настройки Blitz Identity Provider, указав путь к файлу blitz_param.txt:

```
/usr/share/identityblitz/blitz-console/bin/configure -f blitz_param.txt
```

В результате выполнения скрипта будут настроены конфигурационные файлы, а также будет сгенерирован и показан логин/пароль администратора Blitz Identity Provider и сгенерирован пароль от ключевого контейнера:

```
************************************************************
Your instance is configured on domain: test.loc
The Administration Console available on addresses:
  http://testinstallation.local:9001/blitz/console

Administration user credentials of Blitz Console:
  username - admin
  password - 98aAB0D3f2
Your can change user credentials at file - /usr/share/identityblitz/blitz-config/
↪credentials

Create keystore /usr/share/identityblitz/blitz-config/blitz-keystore.bks and␣
↪generate:
 - JWS(RSA256) keypair - jws_rs256_rsa_default
 - AES(AES128) security key - jdbc

Generated password for keystore: BeEBcd2239
************************************************************
```

В случае использования ключей, созданных на этапе установки, перезапустить nginx:

```
systemctl restart nginx
```

Добавить сопоставление адреса loopback-интерфейса и доменного имени, указанного при установке в /etc/hosts:

```
127.0.0.1 localhost.localdomain localhost testinstallation.local
```

Запустить службы:

```
systemctl enable blitz-idp && systemctl start blitz-idp
systemctl enable blitz-console && systemctl start blitz-console
systemctl enable blitz-registration && systemctl start blitz-registration
systemctl enable blitz-recovery && systemctl start blitz-recovery
```

После успешного завершения установки и настройки Blitz Identity Provider возможно подключиться к консоли управления по доменному имени, указанному на этапе установки дистрибутива, например, `https://testinstallation.local/blitz/console`.

РОСА

Установить дистрибутив:

```
/tmp/blitz/blitz-5.23.2.bin -- -j /usr/lib/jvm/java-11 -i "idp console recovery␣
↪registration"
```

Создать конфигурационный файл `blitz_param.txt` следующего содержания, скорректировав в нем настройки на требуемые:

```
DOMAIN=testinstallation.local
MEMCACHED_SERVERS="127.0.0.1"
DB_MODE=PG
PG_HOSTNAME=127.0.0.1
PG_DB_NAME=blitzdb
PG_USERNAME=blitz
PG_PASSWORD=CHANGE_ME
```

Определить JAVA_HOME и запустить скрипт первоначальной настройки Blitz Identity Provider, указав путь к файлу `blitz_param.txt`:

```
export JAVA_HOME=/usr/lib/jvm/java-11
/usr/share/identityblitz/blitz-console/bin/configure -f blitz_param.txt
```

В случае использования ключей созданных на этапе установки запустить nginx:

```
systemctl start nginx
```

Добавить сопоставление адреса loopback-интерфейса и доменного имени, указанного при установке в `/etc/hosts`:

```
127.0.0.1       localhost.localdomain localhost CHANGE_ME
```

Запустить службы:

```
systemctl enable blitz-idp.service && systemctl start blitz-idp.service
systemctl enable blitz-console.service && systemctl start blitz-console.service
systemctl enable blitz-registration.service && systemctl start blitz-registration.
↪service
systemctl enable blitz-recovery.service && systemctl start blitz-recovery.service
```

После успешного завершения установки и настройки Blitz Identity Provider возможно подключиться к консоли управления по доменному имени, указанному на этапе установки дистрибутива, например, `https://testinstallation.local/blitz/console`.

Rocky Linux, AlmaLinux, Oracle Linux и RHEL

Важно: См. *ограничения* (страница 32) при использовании экспресс-инструкций.

Список ОС, для которых приведены инструкции по установке, и их обозначение в данном разделе:

- Rocky 8: Rocky Linux 8;
- Alma 8: AlmaLinux 8;
- Oracle 8: Oracle Linux 8;
- RHEL 8: RHEL 8;
- Rocky 9: Rocky Linux 9;
- Alma 9: AlmaLinux 9;
- Oracle 9: Oracle Linux 9;
- RHEL 9: RHEL 9.

Шаг 1. JDK

Rocky, Alma, Oracle, RHEL 8

Установить дистрибутив:

```
dnf install java-11-openjdk-devel
```

Rocky, Alma, Oracle, RHEL 9

Установить дистрибутив:

```
dnf install java-11-openjdk-devel
```

Шаг 2. Memcached

Rocky, Alma, Oracle, RHEL 8

Установить дистрибутив:

```
dnf install memcached
```

Запустить службу:

```
systemctl enable memcached && systemctl start memcached
```

Rocky, Alma, Oracle, RHEL 9

Установить дистрибутив:

```
dnf install memcached
```

Запустить службу:

```
systemctl enable memcached && systemctl start memcached
```

Шаг 3. PostgreSQL

Rocky, Alma, Oracle, RHEL 8

Установить дистрибутив:

```
dnf install postgresql
```

Инициализировать СУБД командой:

```
postgresql-setup initdb
```

Добавить разрешение в `/var/lib/pgsql/data/pg_hba.conf` на подключение к БД пользователю `blitz`:

```
host blitzdb blitz 127.0.0.1/32 scram-sha-256
```

Указать алгоритм шифрования паролей в `/var/lib/pgsql/data/postgresql.conf`:

```
password_encryption = scram-sha-256
```

Запустить службу:

```
systemctl enable postgresql && systemctl start postgresql
```

Подключиться к СУБД и провести первичную настройку

```
su - postgres
psql

create database blitzdb;
create user blitz with encrypted password 'CHANGE_ME';
grant ALL PRIVILEGES ON DATABASE blitzdb to blitz;
grant ALL on ALL tables in schema public to blitz;
```

Вернуться в shell пользователя `root` и выполнить скрипты создания и обновления структуры БД `blitzdb`:

```
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/000-service-tasks.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/001-init-database.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/002-new_pp_columns.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/003-usd_id_table.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/004-usr_auth_table.
```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/005-usr_agt_table.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/006-usr_htp_hmc_alg.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/007-usr_atr_cfm.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/008-wak.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/009-fix_pp_column.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/010-add_usr_prp.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/011-pp_audit.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/012-geo_to_audit.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/013-tasks.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/014-sec_ch_ua.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/015-5.12.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/016-5.13.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/017-5.15.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/018-5.17.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/019-5.18.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/020-5.20.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/021-5.21.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/022-5.23.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/023-5.26.0.sql
```

Rocky, Alma, Oracle, RHEL 9

Установить дистрибутив:

```
dnf install postgresql-server
```

Инициализировать СУБД командой:

```
postgresql-setup -initdb -unit postgresql
```

Добавить разрешение в `/var/lib/pgsql/data/pg_hba.conf` на подключение к БД пользователю `blitz`:

```
host blitzdb blitz 127.0.0.1/32 scram-sha-256
```

Указать алгоритм шифрования паролей в `/var/lib/pgsql/data/postgresql.conf`:

```
password_encryption = scram-sha-256
```

Запустить службу:

```
systemctl enable postgresql && systemctl start postgresql
```

Вернуться в shell пользователя `root` и выполнить скрипты создания и обновления структуры БД `blitzdb`:

```
su - postgres
psql

create database blitzdb;
create user blitz with encrypted password 'CHANGE_ME';
grant ALL PRIVILEGES ON DATABASE blitzdb to blitz;
grant ALL on ALL tables in schema public to blitz;
```

Выполнить скрипты создания и обновления структуры БД `blitzdb`:

```
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/000-service-tasks.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/001-init-database.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/002-new_pp_columns.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/003-usd_id_table.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/004-usr_auth_table.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/005-usr_agt_table.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/006-usr_htp_hmc_alg.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/007-usr_atr_cfm.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/008-wak.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/009-fix_pp_column.
↪sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/010-add_usr_prp.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/011-pp_audit.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/012-geo_to_audit.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/013-tasks.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/014-sec_ch_ua.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/015-5.12.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/016-5.13.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/017-5.15.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/018-5.17.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/019-5.18.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/020-5.20.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/021-5.21.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/022-5.23.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/023-5.26.0.sql
```

Шаг 4. RabbitMQ

Rocky, Alma, Oracle, RHEL 8

Подготовить конфигурационный файл с репозиториями для RabbitMQ в `/etc/yum.repos.d/rabbitmq.repo`:

```
##
## Zero dependency Erlang
##

[rabbitmq_erlang]
name=rabbitmq_erlang
baseurl=https://packagecloud.io/rabbitmq/erlang/el/8/$basearch
repo_gpgcheck=1
gpgcheck=1
enabled=1
# PackageCloud's repository key and RabbitMQ package signing key
gpgkey=https://packagecloud.io/rabbitmq/erlang/gpgkey

https://github.com/rabbitmq/signingkeys/releases/download/2.0/rabbitmq-release-
↪signing-key.asc
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300
```

(continues on next page)

(продолжение с предыдущей страницы)

```
##
## RabbitMQ server
##

[rabbitmq_server]
name=rabbitmq_server
baseurl=https://packagecloud.io/rabbitmq/rabbitmqserver/el/8/$basearch
repo_gpgcheck=1
gpgcheck=0
enabled=1
# PackageCloud's repository key and RabbitMQ package signing key
gpgkey=https://packagecloud.io/rabbitmq/rabbitmq-server/gpgkey

https://github.com/rabbitmq/signingkeys/releases/download/2.0/rabbitmq-release-
↪signing-key.asc
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300
```

Установить дистрибутив:

```
dnf install rabbitmq-server
```

Запустить службу:

```
systemctl enable rabbitmq-server && systemctl start rabbitmq-server
```

Подготовить очередь для взаимодействия:

```
rabbitmqctl add_user blitz CHANGE_ME
rabbitmqctl set_permissions blitz ".*" ".*" ".*"
rabbitmq-plugins enable rabbitmq_management
curl -vvk 127.0.0.1:15672/cli/rabbitmqadmin >rabbitmqadmin
chmod +x rabbitmqadmin
./rabbitmqadmin declare exchange name=blitz-tasks-exh type=direct
./rabbitmqadmin declare queue name=blitz-tasks durable=true
./rabbitmqadmin declare binding source="blitz-tasks-exh"
destination_type="queue" destination="blitz-tasks"
routing_key="blitz-tasks"
```

Rocky, Alma, Oracle, RHEL 9

Подготовить конфигурационный файл с репозиториями для RabbitMQ в `/etc/yum.repos.d/rabbitmq.repo`:

```
##
## Zero dependency Erlang
##

[rabbitmq_erlang]
name=rabbitmq_erlang
baseurl=https://packagecloud.io/rabbitmq/erlang/el/9/$basearch
repo_gpgcheck=1
gpgcheck=1
enabled=1
# PackageCloud's repository key and RabbitMQ package signing key
gpgkey=https://packagecloud.io/rabbitmq/erlang/gpgkey
```

(continues on next page)

(продолжение с предыдущей страницы)

```
https://github.com/rabbitmq/signingkeys/releases/download/2.0/rabbitmq-release-
↪signing-key.asc
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300

##
## RabbitMQ server
##

[rabbitmq_server]
name=rabbitmq_server
baseurl=https://packagecloud.io/rabbitmq/rabbitmqserver/el/9/$basearch
repo_gpgcheck=1
gpgcheck=0
enabled=1
# PackageCloud's repository key and RabbitMQ package signing key
gpgkey=https://packagecloud.io/rabbitmq/rabbitmq-server/gpgkey

https://github.com/rabbitmq/signingkeys/releases/download/2.0/rabbitmq-release-
↪signing-key.asc
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300
```

Установить дистрибутив:

```
dnf install rabbitmq-server
```

Запустить службу:

```
systemctl enable rabbitmq-server && systemctl start rabbitmq-server
```

Подготовить очередь для взаимодействия:

```
rabbitmqctl add_user blitz CHANGE_ME
rabbitmqctl set_permissions blitz ".*" ".*" ".*"
rabbitmq-plugins enable rabbitmq_management
curl -vvk 127.0.0.1:15672/cli/rabbitmqadmin >rabbitmqadmin
chmod +x rabbitmqadmin
./rabbitmqadmin declare exchange name=blitz-tasks-exh type=direct
./rabbitmqadmin declare queue name=blitz-tasks durable=true
./rabbitmqadmin declare binding source="blitz-tasks-exh"
destination_type="queue" destination="blitz-tasks"
routing_key="blitz-tasks"
```

Шаг 5. 389 Directory Server

Rocky, Alma, Oracle, RHEL 8

Установить дистрибутив:

```
dnf module enable 389-directory-server:stable
dnf install 389-ds-base
```

Включить автоматический запуск службы:

```
systemctl enable dirsrv.target
```

Инициализировать LDAP-каталог:

```
dscreate interactive
```

Выполнить первичную настройку каталога:

```
/tmp/blitz/ldap/ldap_init.sh
```

Rocky, Alma, Oracle, RHEL 9

Установить дистрибутив:

```
dnf install 389-ds-base
```

Включить автоматический запуск службы:

```
systemctl enable dirsrv.target
```

Инициализировать LDAP-каталог:

```
dscreate interactive
```

Выполнить первичную настройку каталога:

```
/tmp/blitz/ldap/ldap_init.sh
```

Шаг 6. Nginx

Rocky, Alma, Oracle, RHEL 8

Установить дистрибутив:

```
dnf install nginx
```

Скопировать файлы для использования:

```
cp /tmp/blitz/nginx/blitz-idp.conf /etc/nginx/conf.d/
cp -R /tmp/blitz/static_errors /usr/share/nginx/html
```

Включить автоматический запуск службы:

```
systemctl enable nginx
```

Rocky, Alma, Oracle, RHEL 9

Установить дистрибутив:

```
dnf install nginx
```

Скопировать файлы для использования:

```
cp /tmp/blitz/nginx/blitz-idp.conf /etc/nginx/conf.d/
cp -R /tmp/blitz/static_errors /usr/share/nginx/html
```

Включить автоматический запуск службы:

```
systemctl enable nginx
```

Шаг 7. Blitz Identity Provider

Rocky, Alma, Oracle, RHEL 8

Установить дистрибутив (подставить в имя файла правильную версию, `JAVA_HOME` и состав устанавливаемых приложений):

```
/tmp/blitz/blitz-5.X.X.bin -- -j <JAVA_HOME> -i "idp console recovery registration"
```

Создать конфигурационный файл `blitz_param.txt` следующего содержания, скорректировав в нем настройки на требуемые:

```
DOMAIN=testinstallation.local
MEMCACHED_SERVERS="127.0.0.1"
DB_MODE=PG
PG_HOSTNAME=127.0.0.1
PG_DB_NAME=blitzdb
PG_USERNAME=blitz
PG_PASSWORD=12ABcd45
```

Запустить скрипт первоначальной настройки Blitz Identity Provider, указав путь к файлу `blitz_param.txt`:

```
/usr/share/identityblitz/blitz-console/bin/configure -f blitz_param.txt
```

В результате выполнения скрипта будут настроены конфигурационные файлы, а также будет сгенерирован и показан логин/пароль администратора Blitz Identity Provider и сгенерирован пароль от ключевого контейнера:

```
*****************************************************************
Your instance is configured on domain: test.loc
The Administration Console available on addresses:
 http://testinstallation.local:9001/blitz/console

Administration user credentials of Console:
  username - admin
  password - 98aAB0D3f2
Your can change user credentials at file - /usr/share/identityblitz/blitz-config/
↪credentials

Create keystore /usr/share/identityblitz/blitz-config/blitz-keystore.bks and␣
```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪generate:
 - JWS(RSA256) keypair - jws_rs256_rsa_default
 - AES(AES128) security key - jdbc

Generated password for keystore: BeEBcd2239
****************************************************************
```

В случае использования ключей, созданных на этапе установки, перезапустить nginx:

```
systemctl restart nginx
```

Добавить сопоставление адреса loopback-интерфейса и доменного имени, указанного при установке в `/etc/hosts`:

```
127.0.0.1 localhost.localdomain localhost testinstallation.local
```

Запустить службы:

```
systemctl enable blitz-idp && systemctl start blitz-idp
systemctl enable blitz-console && systemctl start blitz-console
systemctl enable blitz-registration && systemctl start blitz-registration
systemctl enable blitz-recovery && systemctl start blitz-recovery
```

После успешного завершения установки и настройки Blitz Identity Provider возможно подключиться к консоли управления по доменному имени, указанному на этапе установки дистрибутива, например, `https://testinstallation.local/blitz/console`.

Rocky, Alma, Oracle, RHEL 9

Установить дистрибутив (подставить в имя файла правильную версию, `JAVA_HOME` и состав устанавливаемых приложений):

```
/tmp/blitz/blitz-5.X.X.bin -- -j <JAVA_HOME> -i "idp console recovery registration"
```

Создать конфигурационный файл `blitz_param.txt` следующего содержания, скорректировав в нем настройки на требуемые:

```
DOMAIN=testinstallation.local
MEMCACHED_SERVERS="127.0.0.1"
DB_MODE=PG
PG_HOSTNAME=127.0.0.1
PG_DB_NAME=blitzdb
PG_USERNAME=blitz
PG_PASSWORD=12ABcd45
```

Запустить скрипт первоначальной настройки Blitz Identity Provider, указав путь к файлу `blitz_param.txt`:

```
/usr/share/identityblitz/blitz-console/bin/configure -f blitz_param.txt
```

В результате выполнения скрипта будут настроены конфигурационные файлы, а также будет сгенерирован и показан логин/пароль администратора Blitz Identity Provider и сгенерирован пароль от ключевого контейнера:

```
****************************************************************
Your instance is configured on domain: test.loc
The Administration Console available on addresses:
 http://testinstallation.local:9001/blitz/console

Administration user credentials of Console:
  username - admin
  password - 98aAB0D3f2
Your can change user credentials at file - /usr/share/identityblitz/blitz-config/
↪credentials

Create keystore /usr/share/identityblitz/blitz-config/blitz-keystore.bks and␣
↪generate:
 - JWS(RSA256) keypair - jws_rs256_rsa_default
 - AES(AES128) security key - jdbc

Generated password for keystore: BeEBcd2239
****************************************************************
```

В случае использования ключей, созданных на этапе установки, перезапустить nginx:

```
systemctl restart nginx
```

Добавить сопоставление адреса loopback-интерфейса и доменного имени, указанного при установке в `/etc/hosts`:

```
127.0.0.1 localhost.localdomain localhost testinstallation.local
```

Запустить службы:

```
systemctl enable blitz-idp && systemctl start blitz-idp
systemctl enable blitz-console && systemctl start blitz-console
systemctl enable blitz-registration && systemctl start blitz-registration
systemctl enable blitz-recovery && systemctl start blitz-recovery
```

После успешного завершения установки и настройки Blitz Identity Provider возможно подключиться к консоли управления по доменному имени, указанному на этапе установки дистрибутива, например, `https://testinstallation.local/blitz/console`.

2.1.5 Первые шаги после установки

Настоящий раздел содержит информацию, которая может вам понадобиться непосредственно после установки Blitz Identity Provider.

Настройка опций запуска приложений Blitz Identity Provider

Для приложений Blitz Identity Provider доступны следующие Java-опции, определяющие включение особых режимов функционирования приложений и переопределить стандартные режимы работы:

Внимание: Перед установкой опций рекомендуется проконсультироваться с технической поддержкой Blitz Identity Provider.

- `blitz.login.cookie.sameSite` – задает флаг, с которым должны создаваться сессионные cookies в Blitz Identity Provider. По умолчанию cookies создаются с флагом `sameSite=Lax`. Можно переопределить на значение `None`.

- `blitz.login.outside.flow.callback.ttl.sec` – задает время ожидания ответа от вызванного из Blitz Identity Provider внешнего метода аутентификации. По умолчанию значение `300` секунд.
- `blitz.login.mus.cookie.unused.ttl.sec` – задает срок жизни cookie, отвечающей за запоминание списка залогиненных в текущем браузере пользователей. По умолчанию значение соответствует `365` дней (значение задается в секундах);
- `blitz.login.bua.cookie.ttl.sec` – задает время действия cookie, используемой для запоминания браузера пользователя. По умолчанию значение соответствует `365` дней (значение задается в секундах);
- `blitz.login.setLastAuth.disabled` – позволяет отключить запись в базу данных времени последней аутентификации пользователя. По умолчанию время последней аутентификации пользователя пишется в базу данных. Отключение записи времени последней аутентификации позволяет повысить производительность базы данных, но не позволяет задействовать *функцию блокирования учетных записей по неактивности* (страница 333);
- `blitzDispatchedQueues` – задает имя очереди, из которой приложение Blitz Identity Provider обрабатывает задачи на отправку писем, регистрацию пользователей и восстановление паролей. По умолчанию используется очередь с именем `default`;
- `blitz.stores.united.u-cache.ttlInSec` – срок действия кэша данных учетной записи, предоставляемых через REST API. По умолчанию `1` секунда;
- `blitz.csrf.cookie.ttlInSec` – задает время действия cookie, препятствующей CSRF. По умолчанию соответствует `6` часам (значение задается в секундах). Это максимальное время с момента открытия пользователем страницы и до выполнения заполненной страницы пользователем на сервер;
- `blitz.jdbc.cols.types.strings` – задает тип колонки, используемой для сохранения строковых атрибутов в реляционной СУБД (PostgreSQL). По умолчанию используется тип `text`;
- `blitz.jdbc.pool.stat-period` – задает периодичность, с которой статистика использования JDBC записывается в лог. По умолчанию `300` секунд;
- `saml.numThreads` – задает количество потоков, которые в Blitz Identity Provider обрабатывают запросы на вход через SAML. По умолчанию `32` потока;
- `blitz.oauth.exchange.rules.fs.cache.capacity` – задает размер кэша, используемый Blitz Identity Provider для проверки правил доступа к микросервисам. По умолчанию размер кэша в `10000` проверок;
- `blitz.oauth.dyn.reg.clientSecretLength` – задает размер `client_secret`, генерируемого при динамической регистрации пары `client_id` и `client_secret`. По умолчанию генерируется `client_secret` размером в `15` символов.
- `blitz.oauth.dyn.reg.clientAttachingTllInSec` – задает время, в течение которого сгенерированная при динамической регистрации пара `client_id` и `client_secret` должна быть ассоциирована с пользователем (если в течение этого времени пара не будет ассоциирована с пользователем, то она будет аннулирована). По умолчанию соответствует `1` часу (значение задается в секундах).
- `blitz.session.checkRemoteAddress.disabled` – задайте `true` для отключения проверки равенства IP-адреса сессии и входящего запроса (рекомендуется при наличии пользователей с динамическими IP-адресами).
- `blitz.webauthn.residentKey.preferred` – если опция задана, то ключи безопасности регистрируются с параметром `residentKey=preferred`. При этом, в случае если опция задана как `true`, то `requireResidentKey=true`, а если опция `false`, то `requireResidentKey=false`.
- `blitz.ldap.store.extension.class` – при передаче в опцию значения `com.identityblitz.idp.store.ldap.custom.PasswordMigrationExt` включается режим миграции пароля.

- `blitz.ldap.store.extension.PasswordMigrationExt.passwordHashAttr` – задает имя LDAP-атрибута, в котором храниться хеш-пароля для опции миграции пароля. Хэш должен содержать префикс `{bcrypt}` для миграции паролей из хэшей с алгоритмом `bcrypt`.
- `extensionsDir` – адрес директории с *модулями расширений* (страница 314).
- `metrics` – позволяет отключить сбор метрик функционирования в формате **`Prometheus`**. Для этого нужно выставить значение `false`. По умолчанию сбор метрик включен
- `couchbase.durability.mode` – задает режим сохранения данных в Couchbase Server. В случае использования Couchbase Server версии 6.0.1 и более старых должен обязательно использоваться режим `clientVerified`. В случае использования Couchbase Server версий 6.5, 7.0 или новее режим `clientVerified` использовать нельзя. Параметр в Couchbase Server версий 6.5, 7.0 становится опционален (при отсутствии параметра используется режим `majority`) и позволяет выбрать требуемый режим гарантированности сохранения данных в кластере с репликацией из следующих вариантов[23]:
  - `disabled` – ожидание записи только в память на основном узле кластера;
  - `majority` – ожидание записи в память на основном узле и большинстве реплик;
  - `majorityAndPersistActive` – ожидание записи на диск на основном узле и записи в память большинства реплик;
  - `persistToMajority` – ожидание записи на диск на основном узле и в большинстве реплик.
- `akka.http.parsing.max-uri-length` – задает максимальную длину URI в строке браузера. В некоторых случаях может потребоваться увеличить размер строки, тогда рекомендуется в этом параметре задать значение `16k`.
- `akka.http.parsing.max-header-value-length` – задает максимально допустимый размер HTTP-заголовка. В некоторых случаях может потребоваться увеличить размер заголовка, тогда рекомендуется в этом параметре задать значение `16k`.
- `akka.coordinated-shutdown.phases.service-stop.timeout` – задает время ожидания после получения команды на остановку сервиса, в течение которого сервис может завершить взятые в работу задачи. В случае использования встроенного в Blitz Identity Provider брокера сообщений рекомендуется выставить для приложения параметр в значение `30s`.
- `memcached.locator.tries` – определяет количество попыток найти работающий сервер `Memcached` в случае обработки сбоя обращения к серверу `Memcached`.

Предупреждение: Не гарантируется, что используемые опции будут сохранены в будущих версиях Blitz Identity Provider.

Для задания опций со значениями, отличающимися от значений по умолчанию, необходимо отредактировать файл `/etc/default/blitz-idp`. Задать в нем необходимые `JAVA_OPTS`. Ниже приведен пример файла, в котором среди Java-опций также заданы опции `blitz.csrf.cookie.ttlInSec` и `blitz.login.cookie.sameSite`. После изменения `JAVA_OPTS` необходимо перезапустить приложения Blitz Identity Provider, на которых сделаны изменения.

```
export JAVA_HOME=/usr/java/default
export PIDFILE=/usr/share/identityblitz/blitz-idp/RUNNING_PID
export JAVA_OPTS="-server -Xms512m -Xmx1G -XX:MaxMetaspaceSize=512m -Xmn256m -Dcom.
↪couchbase.connectTimeout=30000 -Dakka.http.parsing.max-uri-length=16k"
export JAVA_OPTS="$JAVA_OPTS -Dblitz.csrf.cookie.ttlInSec=36000 -Dblitz.login.
↪cookie.sameSite=None -Dplay.filters.headers.frameOptions=null"
```

[23] https://docs.couchbase.com/server/current/learn/data/durability.html

Вход в консоль управления

После установки Blitz Identity Provider основная настройка системы осуществляется в консоли управления, которая доступна по ссылке, обозначенной в результатах установки продукта. Для первого входа в консоль управления нужно использовать логин и пароль, сгенерированные в момент установки консоли управления.

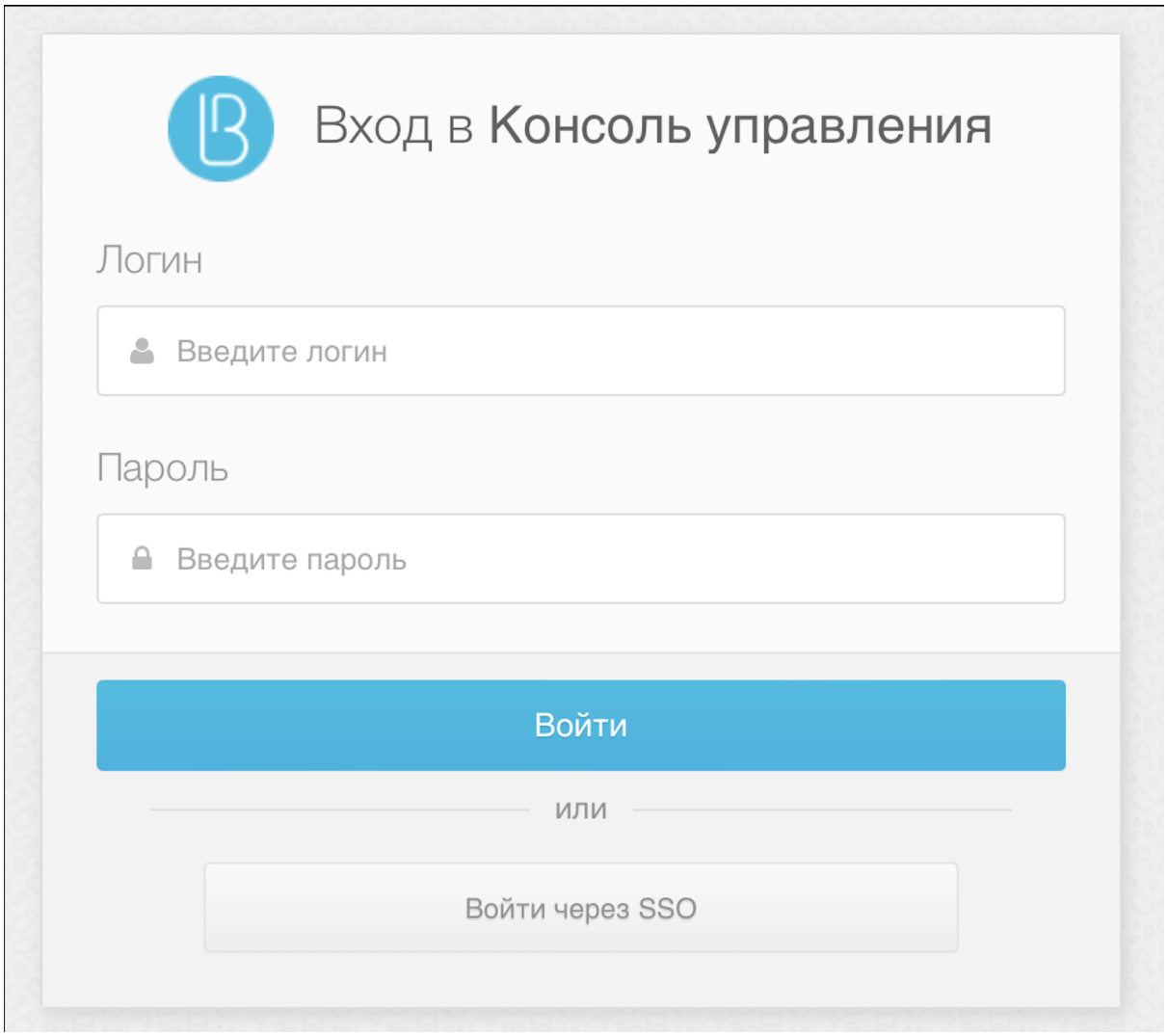
Обычно ссылка имеет вид:

```
https://<blitz_domain>/blitz/console
```

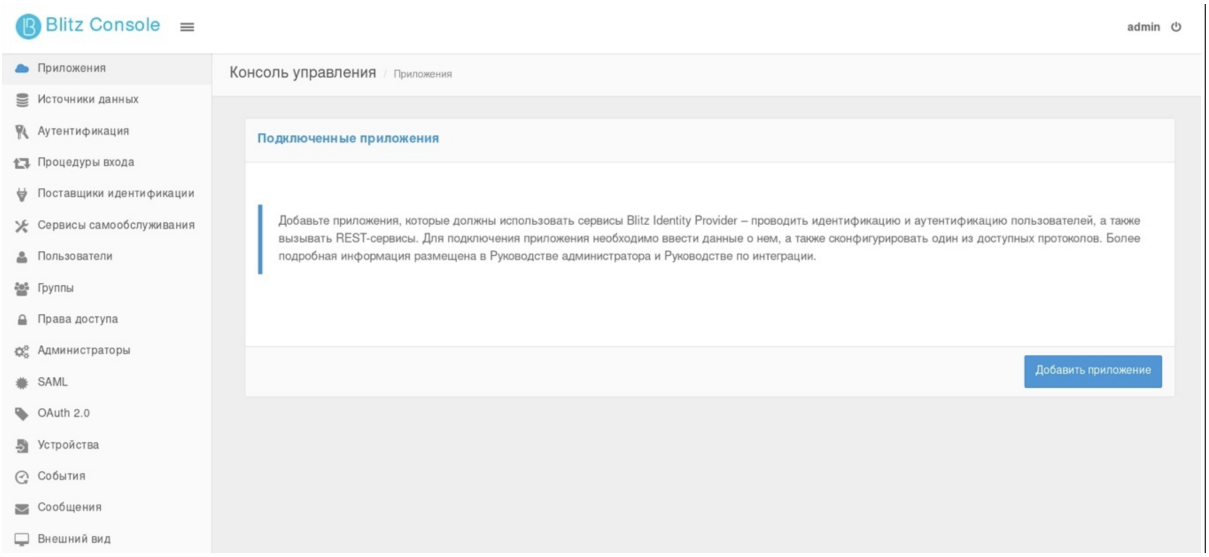
или

```
http://<blitz_console_host>:9001/blitz/console
```

Стандартный вид экрана входа в консоль управления приведен на рисунке:



После успешного входа откроется главная страница консоли управления, вид которой приведен ниже. Навигация между различными настройками Blitz Identity Provider осуществляется с помощью меню, расположенного в левой части экрана.

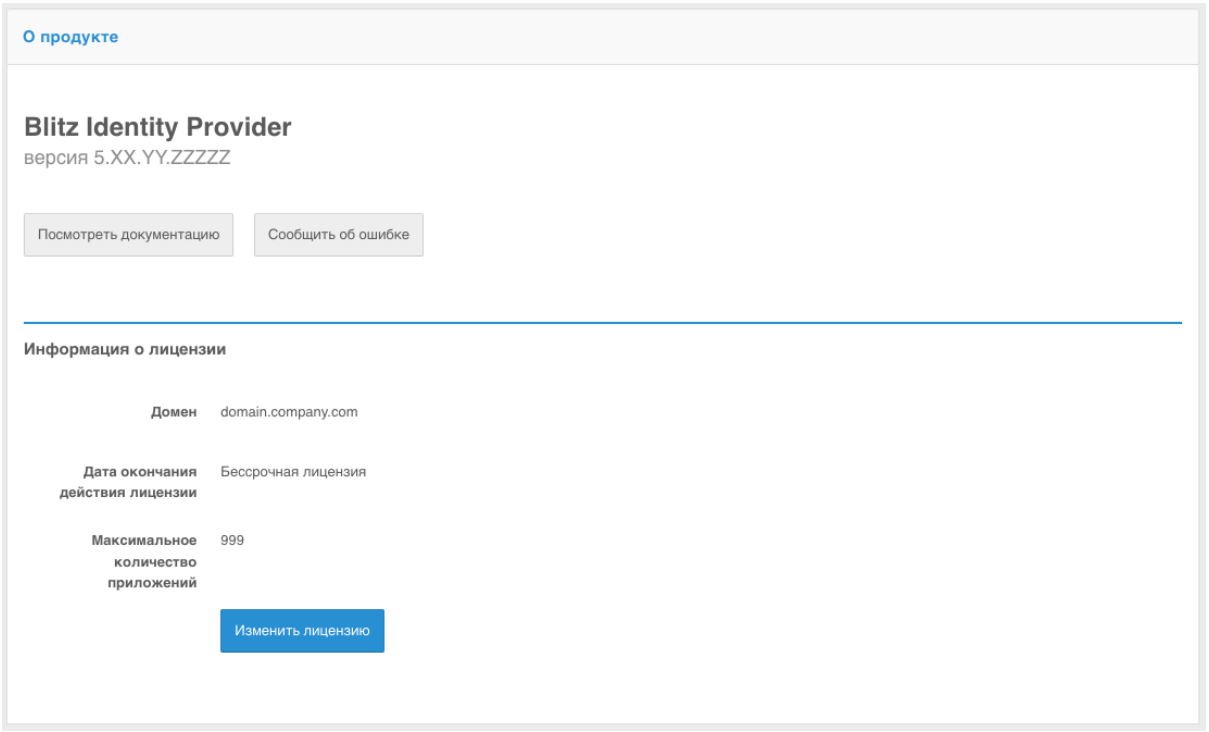


Установка лицензионного ключа

Если нажать на ссылке Вы используете …, версия … в футере любой страницы консоли управления Blitz Identity Provider, то будет отображен экран, приведенный далее.

На этом экране можно ознакомиться с номером версии текущей установки Blitz Identity Provider, перейти на сайт документации ПО и форму обратной связи.

В блоке Информация о лицензии можно посмотреть срок окончания лицензии и предельно разрешенное лицензией количество подключаемых приложений. При нажатии кнопки Изменить лицензию можно ввести новый лицензионный ключ.



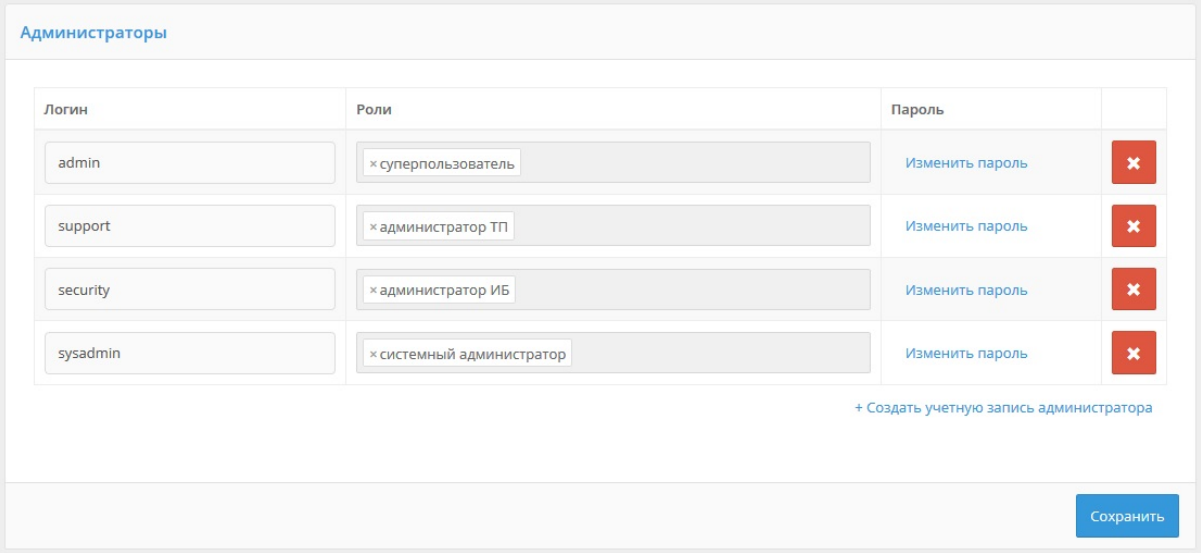
После установки нового лицензионного ключа рекомендуется перезапустить приложения Blitz Identity Provider.

Также задать лицензионный ключ можно через редактирование конфигурационного файла `blitz.conf` в каталоге `/usr/share/identityblitz/blitz-config`. Нужно найти блок настроек `blitz.prod.local.idp.license` и скорректировать его следующим образом (задать лицензионный ключ в параметре `key`):

```
"license" : {
  "key" : "MEQC…U"
}
```

Управление учетными записями администраторов

После установки Blitz Identity Provider рекомендуется создать дополнительные учетные записи администраторов, назначить им пароли и административные роли. Управление учетными записями администраторов доступно в разделе Администраторы.



В разделе Администраторы доступны следующие действия:

- создание и удаление учетных записей администраторов;
- изменение паролей учетных записей администраторов;
- назначение и отзыв ролей администраторов.

По умолчанию в Blitz Identity Provider доступны роли, приведенные в таблице. Можно перенастроить существующие роли или создать новые через настройки конфигурационного файла `credentials`.

Стандартные роли администраторов в Blitz Identity Provider

Роль	Доступные разделы консоли управления
суперпользователь (root)	Доступно все
администратор ИБ (security)	Администраторы, События
системный администратор (sysadmin)	Источники данных, Аутентификация, Процедуры входа, Поставщики идентификации, SAML, OAuth 2.0, Устройства, Сообщения
администратор приложений (app_admin)	Приложения
Администратор интерфейса (ui_admin)	Сервисы самообслуживания, Внешний вид
администратор ТП (support)	Пользователи, Группы, Права доступа, События

Дополнительно к стандартной идентификации и аутентификации администраторов по логину и паролю при входе в консоль управления можно настроить использование идентификации и аутентификации пользователей в консоль управления с использованием сервера аутентификации Blitz Identity Provider. Настройки выполняются через конфигурационный файл `console.conf`.

Перезапуск сервисов Blitz Identity Provider

Для перезапуска сервисов Blitz Identity Provider необходимо использовать команду:

```
systemctl restart APP_NAME
```

Вместо `APP_NAME` нужно указать имя перезапускаемого приложения: `blitz-console`, `blitz-idp`, `blitz-registration`, `blitz-recovery`, `blitz-keeper`.

Пример команды для перезапуска приложения сервиса аутентификации:

```
systemctl restart blitz-idp
```

Удаление использованных при установке файлов

При первом запуске Blitz Identity Provider зашифровывает созданные при установке пароли администратора и пароли подключения к СУБД. При этом первоначальный конфигурационный файл копируется в каталог `/usr/share/identityblitz/blitz-config/.snapshot`. Рекомендуется удалить использованные при установке файлы `blitz_param.txt` и первые созданные копии конфигурационного файла `blitz.conf`. Для этого можно выполнить команду:

```
rm blitz_param.txt /usr/share/identityblitz/blitz-config/.snapshot/blitz.conf.*
```

2.2 Базовое конфигурирование

2.2.1 Атрибуты учетных записей

Учетная запись пользователя в Blitz Identity Provider описывается набором атрибутов. Данный раздел посвящен всем аспектам работы с ними.

Что представляет собой атрибут учетной записи

Учетная запись пользователя описывается набором атрибутов.

Значения атрибутов формируются следующими способами:

- считываются из *подключенных хранилищ атрибутов* (страница 78);
- считываются из базы данных Blitz Identity Provider;

  Примечание: Чтение и сохранение атрибута в базе данных осуществляется в случае, если для атрибута не настроена связка с атрибутом в подключенном хранилище атрибутов.

- вычисляются из других атрибутов или заполняются константными значениями.

  Совет: Можно вычислять атрибут `домен пользователя` из адреса электронной почты или создать композитный атрибут `ФИО` из отдельных атрибутов с фамилией, именем и отчеством пользователя.

Конфигурирование атрибутов состоит из:

- настройки хранимых атрибутов, т.е. тех, которые ведутся в подключенных хранилищах или в базе данных Blitz Identity Provider;
- настройки вычисляемых атрибутов, т.е. тех, которые должны принимать константное значение или которые вычисляются по правилам.
- настройки правил преобразования входных значений, позволяющих преобразовывать значения атрибутов при изменении (например, при редактировании пользователем или при вызове соответствующих API);
- настройки правил преобразования выходных значений, позволяющих провести дополнительные преобразования с вычисляемыми атрибутами;
- настройки назначения атрибутов – определение идентификатора в системе и атрибутов, отвечающих за номер мобильного телефона, адрес электронной почты.

Внимание: Для корректной работы Blitz Identity Provider как минимум должны быть выполнены следующие настройки:

- сконфигурированы необходимые атрибуты;
- один из атрибутов определен в качестве идентификатора.

Конфигурирование доступных атрибутов

Хранимые атрибуты

Необходимо в разделе Источники данных перейти в блок Хранимые атрибуты и выполнить следующие шаги:

- добавить новый атрибут, нажав на ссылку `+Добавить атрибут`;
- указать наименование атрибута, которое будет использоваться в Blitz Identity Provider. Наименование атрибута может отличаться от его имени во внешнем хранилище – в таком случае необходимо указать правило преобразования в *настройках* (страница 78) этого хранилища;
- указать тип значения данных – формат данных (`String`, `Number`, `Boolean`, `Bytes`, `Array of Strings`);
- определить параметры атрибута:
  - возможно ли производить по нему поиск (столбец `Поиск`);

    Совет: Если это атрибут из подключенного хранилища, то в целях производительности рекомендуется создать по нему поисковый индекс.

  - является ли атрибут обязательным (столбец `Обяз`);
  - должно ли значение атрибута быть уникальным в системе (столбец `Уник`).

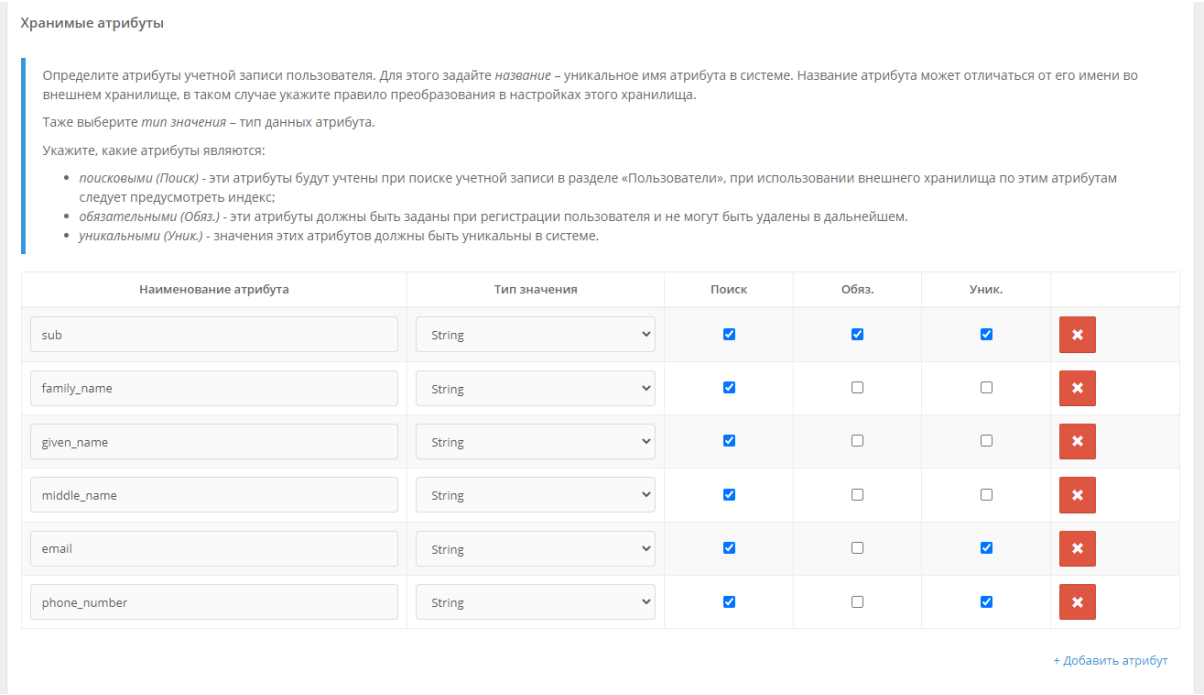
После добавления атрибута недопустимо менять его имя. При необходимости переименования атрибута следует удалить атрибут и создать новый.

Важно: В разделе Пользователи в *карточке пользователя* (страница 203) атрибуты будут показываться в том порядке, в котором они созданы. Через консоль управления изменить порядок атрибутов нельзя. При необходимости изменить порядок атрибутов необходимо вручную их переупорядочить в конфигурационном файле `blitz.conf` в секции настроек `blitz.prod.local.idp.id-attrs`. Чтобы в разделе `Пользователи` вместо системных имен атрибутов показывались их текстовые названия с учетом языка интерфейса пользователя, необходимо для созданных атрибутов *определить* (страница 301) в `messages` строки с описанием названий атрибутов для используемых языков. Строки должны иметь вид `custom.user.attr.name.<имя атрибута>`.

При создании нового атрибута автоматически также создается маппинг нового атрибута во всех подключенных хранилищах атрибутов на атрибут с таким же названием. После создания новых атрибутов необходимо проверить и отредактировать настройки маппинга в подключенных хранилищах. Если атрибут не предполагается считывать из хранилища, то нужно удалить строку маппинга – в таком случае атрибут будет вестись в базе данных Blitz Identity Provider.

Важно: Если в качестве СУБД используется PostgreSQL, то необходимо создать столбец в таблице `USR_ATR`, а также в таблице `USR` (только в случае, если используется *внутреннее хранилище* (страница 77)). Имя столбца должно соответствовать имени добавляемого атрибута с нормализацией из `lowerCamelCase` в `UPPERCASE_SEPARATED_BY_UNDERSCORE`, например, `middleName` -> `MIDDLE_NAME`. Тип столбца должен быть выбран в зависимости от типа значения атрибута:

- столбец с типом `text` для атрибутов с типом `String` и `Bytes` (в этом случае значение будет сохранено в `Base64`);
- столбец с типом `text[]` для атрибута с типом `Array of strings`;
- столбец с подходящим числовым типом (`bigint`, `integer`, `smallint`) для атрибутов с типом `Number`;
- столбец с типом `bool` для атрибута с типом `Boolean`.



Предусмотрена возможность назначить для атрибута LDAP-каталога транслятор, осуществляющий преобразование атрибута из хранимого в LDAP формата в требуемый формат в Blitz Identity Provider. Например, это может быть полезно при необходимости обрабатывать в Blitz Identity Provider атрибут `objectGUID` из LDAP-каталога Active Directory, чтобы этот атрибут представлялся не в байтовом виде, а в форме строки GUID. Настройка *выполняется* (страница 315) через конфигурационный файл.

Вычисляемые атрибуты

Для настройки вычисляемых атрибутов в блоке Вычисляемые атрибуты необходимо совершить следующие действия:

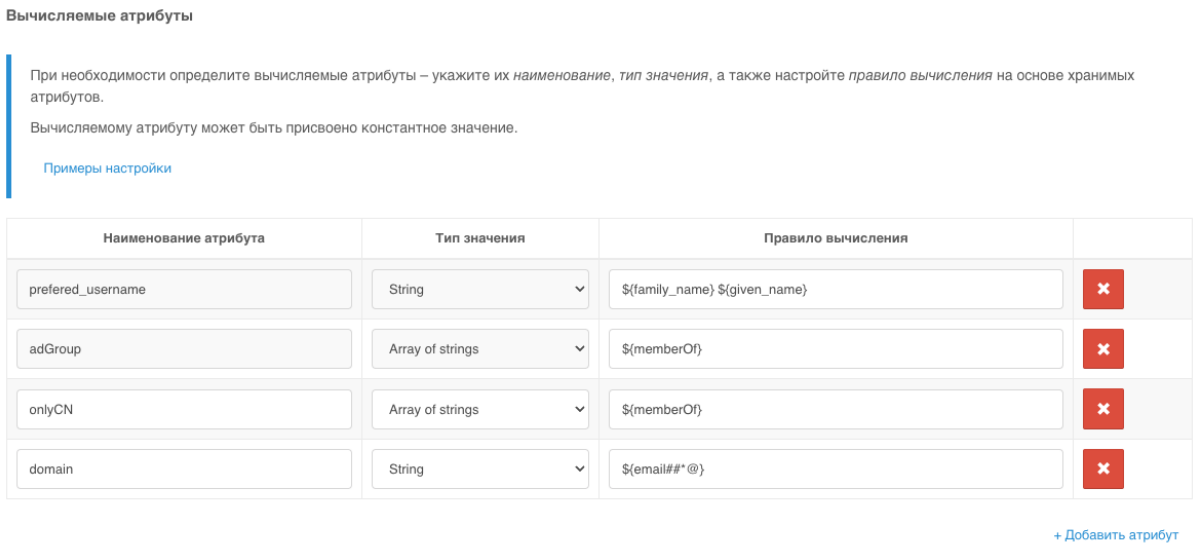
- добавить новый атрибут, нажав на ссылку `+Добавить атрибут`;
- указать наименование вычисляемого атрибута;
- указать тип значения данных – формат данных;
- указать правило вычисления атрибута на основе других атрибутов или присвоения ему константного значения.

Примеры правил:

- чтобы создать атрибут `Имя и фамилия` из хранимых атрибутов `family_name` и `given_name` необходимо определить хранимые атрибуты `family_name` и `given_name`, а далее задать вычисляемый атрибут `full_name` с правилом вычисления – `${family_name} ${given_name}`.
- чтобы создать атрибут `домен электронной почты` из хранимого атрибута `email` необходимо определить хранимый атрибут `email`, а далее задать вычисляемый атрибут `domain` и определить его правило вычисления `${email##*@}`.

Примечание: Справку по поддерживаемым параметрам строк подстановки можно посмотреть здесь[24].

[24] http://tldp.org/LDP/abs/html/parameter-substitution.html

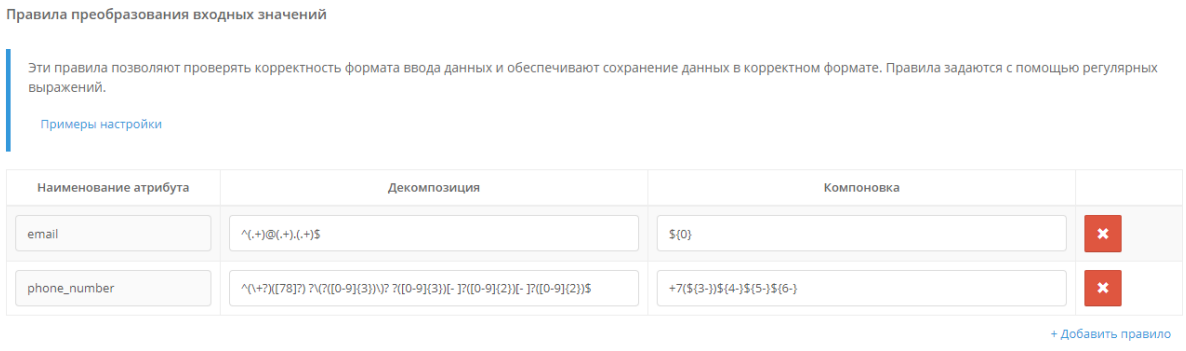


Правила преобразования входных значений

Правила преобразования входных значений позволяют проверять корректность формата ввода данных и обеспечивают сохранение данных в корректном формате. Правила задаются с помощью регулярных выражений. Каждое правило включает в себя регулярное выражение, позволяющее провести декомпозицию (разбиения на части) введенного значения, и правило сохранения полученных частей (компоновка).

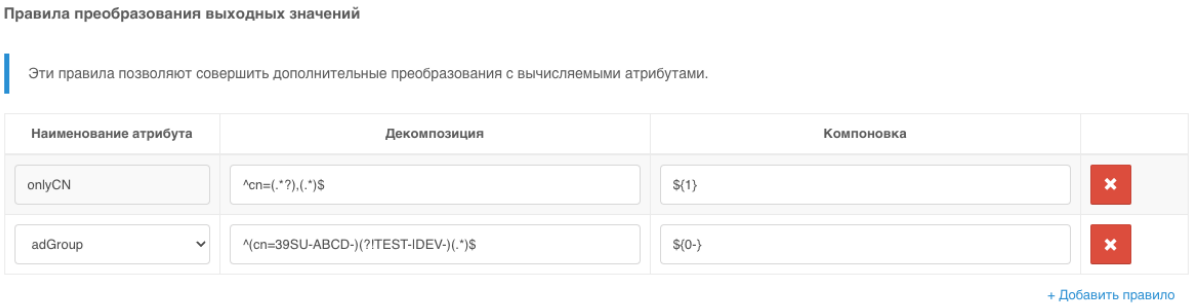
Пример решаемых задач:

- для проверки, что атрибут `email` содержит знак `@`, необходимо указать выражение декомпозиции `^(.+)@(.+)$` и выражение компоновки `${0-}`;
- для проверки формата мобильного телефона (`phone_number`) и сохранения его в формате `+7(999)1234567`, необходимо указать выражение декомпозиции `^(\+?)([78]?) ?\(?([0-9]{3})\)? ?([0-9]{3})[- ]?([0-9]{2})[- ]?([0-9]{2})$` и выражение компоновки `+7(${3-})${4-}${5-}${6-}`.



Правила преобразования выходных значений

Эти правила позволяют совершить дополнительные преобразования с вычисляемыми атрибутами. Например, из атрибута с массивом групп пользователей могут быть извлечены только необходимые группы, либо значения групп из формата `CN=name, DC=...` должны быть преобразованы просто к именам `CN`. Примеры настроек таких правил преобразования представлены на рисунке ниже (предварительно необходимо *создать* (страница 74) соответствующие вычисляемые атрибуты).



Настройка назначения атрибутов

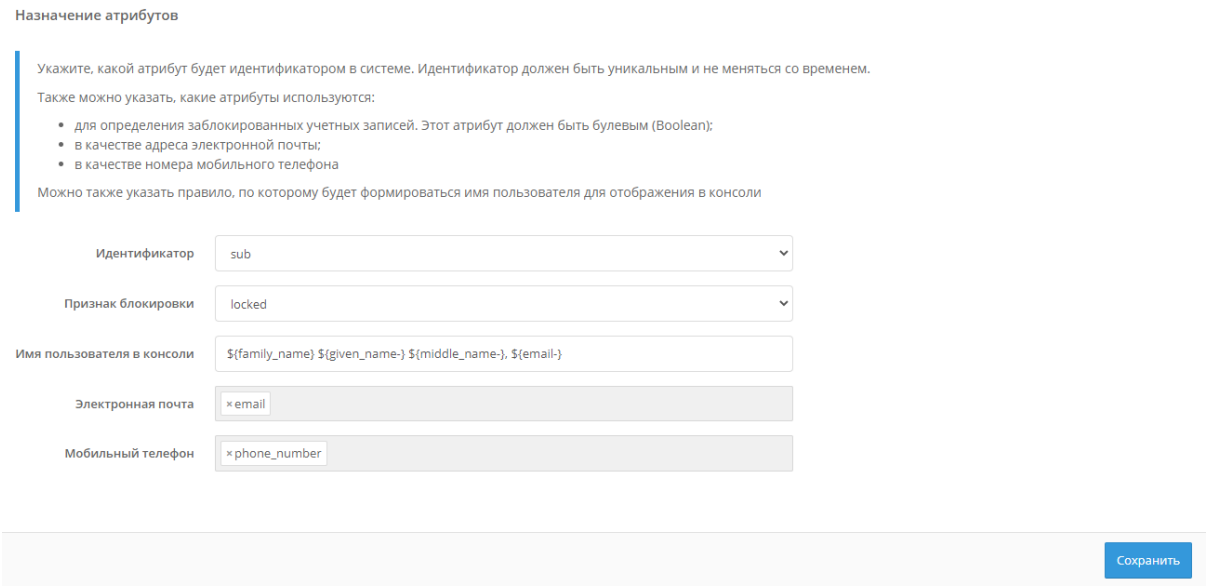
Необходимо указать, какой атрибут будет идентификатором в системе. Идентификатор должен быть уникальным и не меняться со временем.

Примечание: Не рекомендуется в будущем менять базовый идентификатор, т.к. к нему привязываются все пользовательские настройки. При изменении базового идентификатора будут потеряны настройки двухфакторной аутентификации, зарегистрированные события безопасности, запомненные списки устройств пользователей, связи с внешними учетными записями, хранимые в базе данных Blitz Identity Provider атрибуты пользователей.

Также можно указать, какие атрибуты используются для специальных целей:

1. Атрибут, используемый в качестве признака блокировки учетной записи. Этот атрибут должен иметь тип значения `Boolean`. Blitz Identity Provider поддерживает блокировку пользователей, хранимых в LDAP-каталоге. Для использования этой функции также требуется настроить соответствующий атрибут в *настройках* (страница 78) LDAP-каталога.
2. Выражение, определяющее имя пользователя в консоли. Например, выражение `${family_name} ${given_name} ${middle_name-}` позволяет отображать у учетной записи (например, в разделе Пользователи) фамилию, имя и отчество (если есть).
3. Атрибуты, используемые для хранения адресов электронной почты.
4. Атрибуты, используемые для хранения номером мобильных телефонов.

В качестве электронной почты и мобильного телефона могут быть указаны несколько атрибутов (например, для личного и рабочего адреса электронной почты).



Подключение хранилищ атрибутов

Типы хранилищ

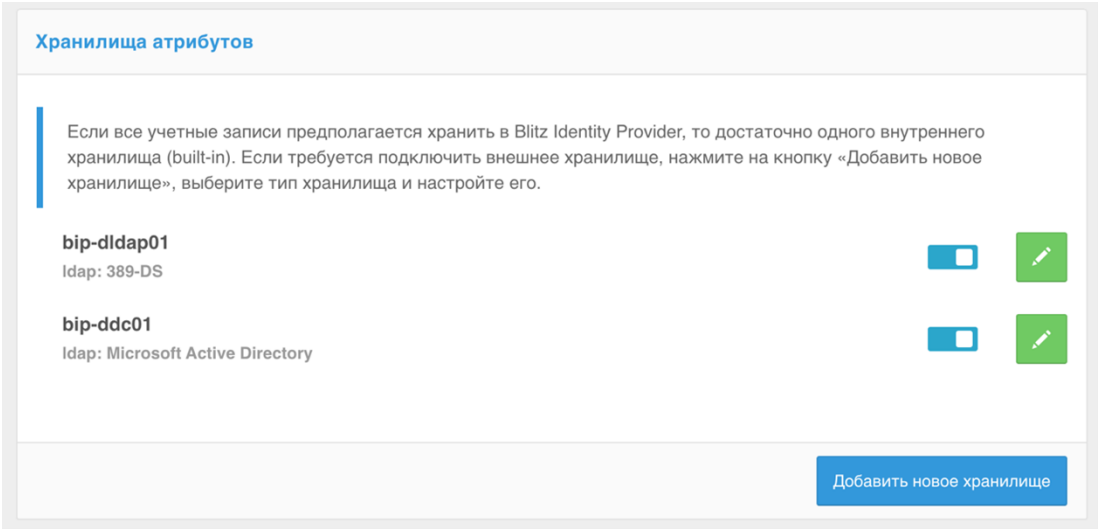
В качестве хранилищ атрибутов пользователей Blitz Identity Provider позволяет использовать:

1. Внешнее (подключенное) хранилище. В качестве такового может выступать:
   - LDAP-хранилище – это может быть любой сервер, поддерживающий протокол LDAP (389 Directory Server, OpenLDAP, FreeIPA и другие), а также Microsoft Active Directory или Samba4;
   - иное хранилище, для подключения которого к Blitz Identity Provider необходимо разработать специальные *REST-сервисы* (страница 82).
2. Внутреннее хранилище. Все атрибуты пользователей хранятся в базе данных Blitz Identity Provider. В случае если в качестве СУБД используется Couchbase Server, то базу данных Blitz Identity Provider можно использовать для хранения небольшого числа учетных записей. В случае если в качестве СУБД используется PostgreSQL, то можно хранить любое число учетных записей.

Для корректной работы Blitz Identity Provider требуется настройка хотя бы одного хранилища и конфигурирование *атрибутов* (страница 73). По умолчанию настроено внутреннее хранилище и добавлен ряд атрибутов.

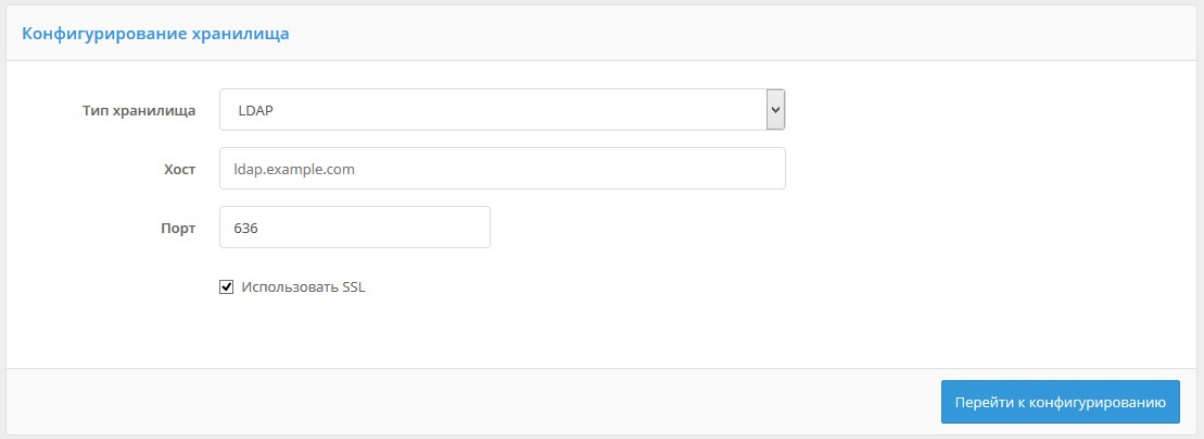
Примечание: Каждая учетная запись пользователя хранится в каком-то одном определенном хранилище. Blitz Identity Provider допускает конфигурирование и подключение нескольких хранилищ, однако рекомендуется использовать одно основное хранилище для работы. Решение об использовании второго хранилища должно быть принято с учетом применяемой модели данных. Например, в подключенном корпоративном Active Directory могут храниться данные сотрудников организации, а в дополнительном LDAP-хранилище – данные специально зарегистрированных «внешних» пользователей (сотрудники партнерских организаций, фрилансеры и пр.).

Выбор и настройка используемого хранилища осуществляется после настройки атрибутов в разделе Источники данных в разделе Хранилища атрибутов. По умолчанию настроено внутреннее хранилище. Для добавления внешнего хранилища следует нажать на кнопку Добавить новое хранилище, после чего указать тип внешнего хранилища и настроить параметры взаимодействия с ним. Хранилища после создания создаются выключенными – их нужно включить с помощью тумблера в разделе Хранилища атрибутов.



Допустимо удалить внутреннее хранилище, если его не планируется использовать. Для этого необходимо перейти в свойства соответствующего внешнего хранилища и нажать на кнопку Удалить.

Использование нескольких хранилищ может решить задачу входа пользователей, хранящихся в разных LDAP-каталогах или в разных ветках одного каталога. Например, в результате объединения двух компаний можно подключить два каталога к Blitz Identity Provider и обеспечить вход пользователей, не прибегая к настройкам доверия или построению метакаталога.



Подключение хранилища по LDAP

Если в качестве источника учетных записей пользователей используется LDAP хранилище, развернутое в организации, для его настройки необходимо воспользоваться разделом Источники данных консоли управления и выполнить следующие шаги:

- добавить новое хранилище, указать следующие данные:
  - тип добавляемого хранилища – выбрать `LDAP`;
  - адрес хранилища;
  - порт;
  - отметить флажок `Использовать SSL`, если должно использоваться защищенное соединение.
- сконфигурировать LDAP-хранилище, настроив следующие параметры:
  - описание хранилища (опционально);

  - используется ли хранилище только для чтения данных или возможна запись в него;
  - необходимость использования SSL-соединения;
  - необходимость DNS-балансировки вызовов к LDAP-хранилищу – для этого нажать кнопку DNS-балансировка и задать параметры `Доменное имя`, `Порт`, `Использовать SSL`, `Режим работы`, `Время хранения в кэше, мс`;

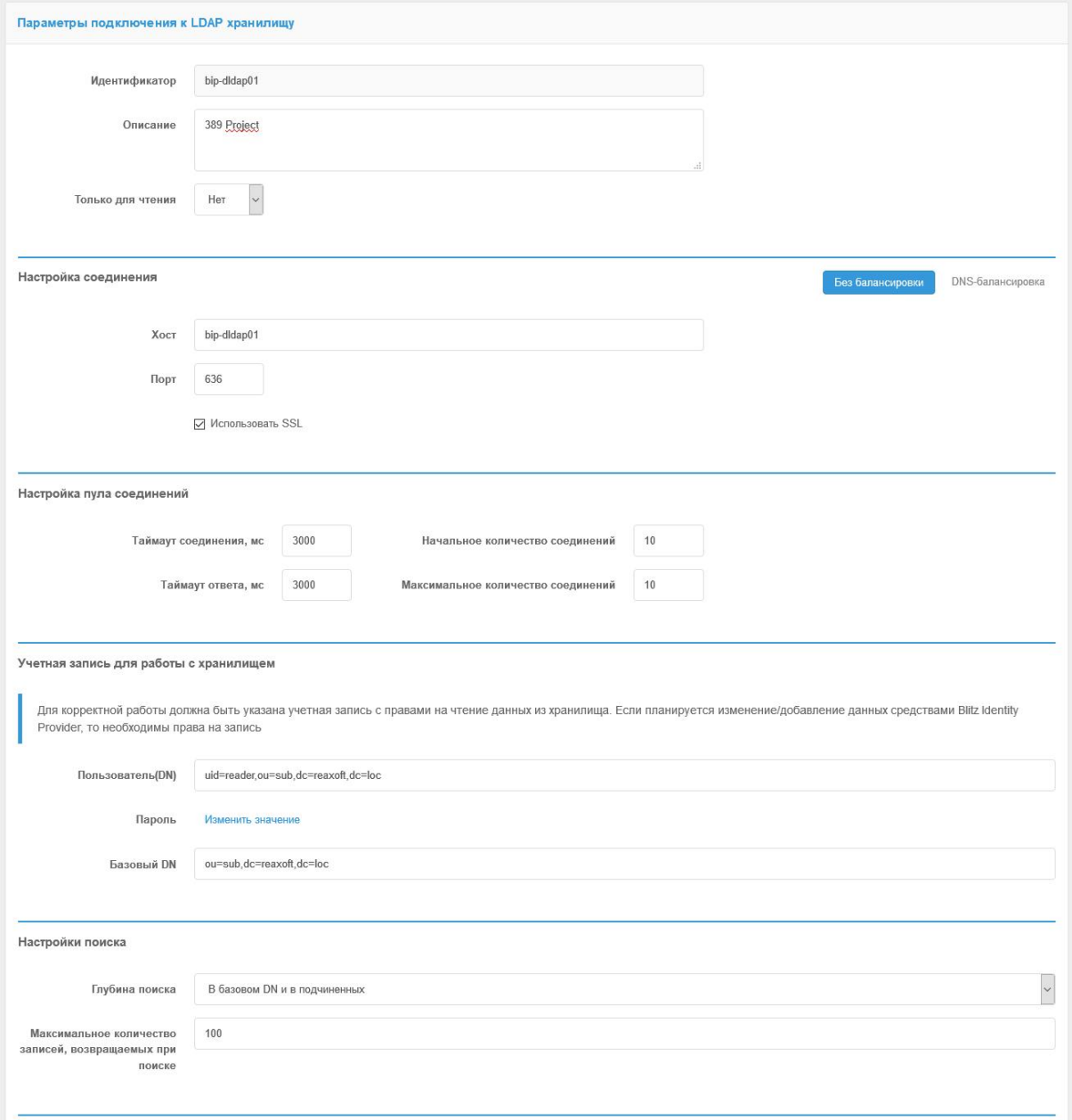
    Примечание: При DNS-балансировке Blitz Identity Provider запрашивает у DNS-сервера по заданному доменному имени LDAP-каталога все адреса подключения. Если в DNS прописано более одного адреса, то в зависимости от выбранного режима работы Blitz Identity Provider устанавливает подключение к первому доступному серверу (режим работы FAILOVER), к случайному серверу (режим работы RANDOM) или к каждому серверу по очереди (режим работы ROUND_ROBIN). Полученный от DNS список серверов хранится в кэше Blitz Identity Provider в течение времени, заданного в настройке `Время хранения в кэше, мс`.

  - настройки пула соединений;

- указать логин и пароль пользователя, от имени которого будет осуществляться работа с LDAP-хранилищем (у этого пользователя должны быть права на чтение и на запись данных), а также базовый DN – раздел каталога с учетными записями пользователей;

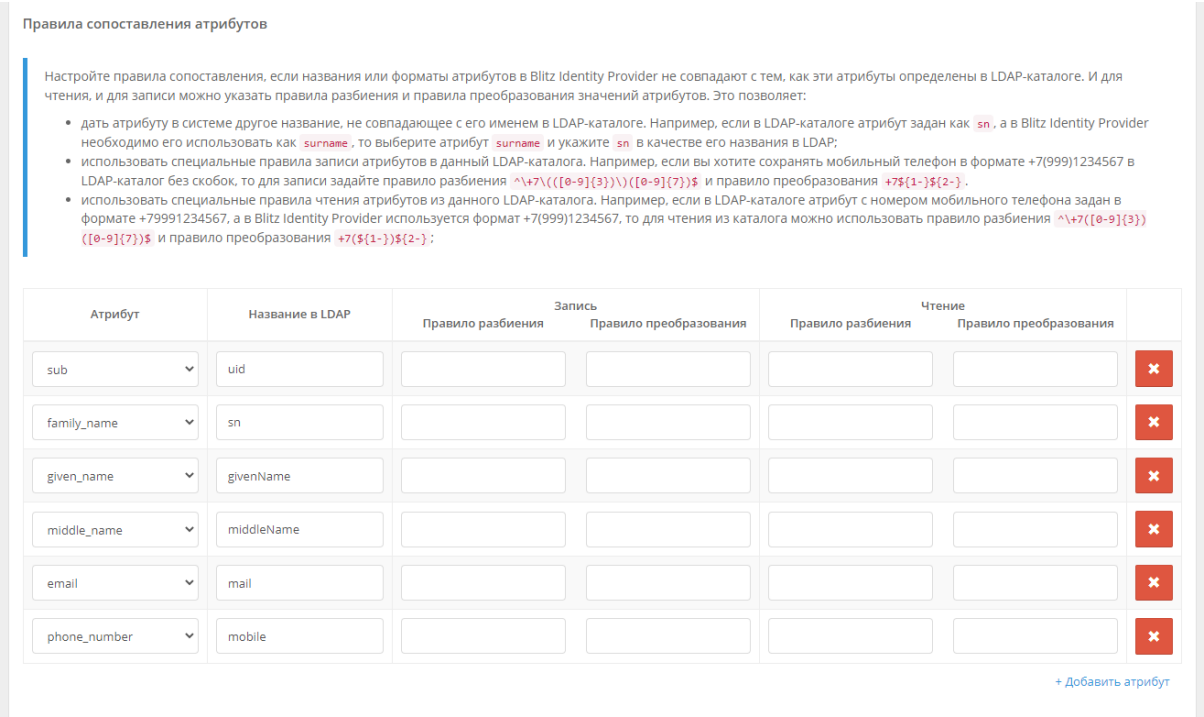
  Примечание: Допустимо указать пользователя только с правами на чтение, если хранилище используется только для чтения.

- указать настройки поиска – глубину поиска и максимальное число возвращаемых учетных записей (это влияет на число пользователей, отображаемых в разделе Пользователи консоли управления).



Далее можно настроить правила сопоставления атрибутов и указать правила разбиения и правила преобразования значений атрибутов. Это позволяет:

- дать атрибуту в системе другое название, не совпадающее с его именем в LDAP-каталоге. Например, если в LDAP-каталоге атрибут задан как `sn`, а в Blitz Identity Provider необходимо его использовать как `family_name`, то выберите атрибут `family_name` и укажите `sn` в качестве его названия в LDAP. Пример такой настройки приведен на рисунке ниже;
- использовать специальные правила записи атрибутов в данный LDAP-каталог. Например, если вы хотите сохранять мобильный телефон в формате `+7(999)1234567` в LDAP-каталог без скобок, то для записи задайте правило разбиения `^\+7\(([0-9]{3})\)([0-9]{7})$` и правило преобразования `+7${1-}${2-}`.
- использовать специальные правила чтения атрибутов из данного LDAP-каталога. Например, если в LDAP-каталоге атрибут с номером мобильного телефона задан в формате `+79991234567`, а в Blitz Identity Provider используется формат `+7(999)1234567`, то для чтения из каталога можно использовать правило разбиения `^\+7([0-9]{3})([0-9]{7})$` и правило преобразования `+7(${1-})${2-}`.

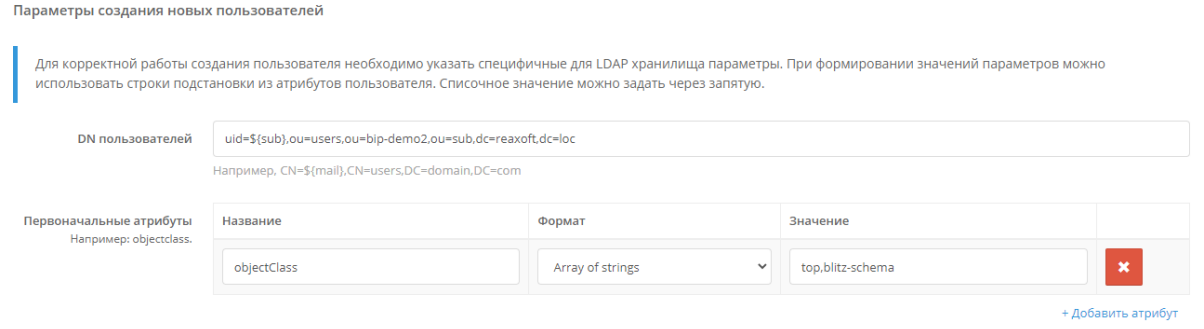


Если хранение *созданного ранее* (страница 73) атрибута в данном хранилище не предполагается, то можно просто удалить атрибут, используя кнопку удаления. В этом случае значение удаленного атрибута будет сохраняться при создании/редактировании учетной записи не в подключаемом внешнем хранилище, а в базе данных Blitz Identity Provider.

Если планируется использовать возможность блокировки учетной записи, то необходимо удалить атрибут, определенный в разделе Источники данных в качестве признака блокировки, из таблицы с правилами сопоставления атрибутов.

Если Blitz Identity Provider используется для регистрации пользователей, причем запись осуществляется в данный каталог, то необходимо указать параметры создания новых пользователей – DN родительского контейнера, внутри которого будут создаваться пользователи, и системные атрибуты, связанные со спецификой хранилища***.

Примечание: Например, `objectclass`, определяющий тип создаваемой учетной записи в LDAP. Для Microsoft Active Directory `objectclass` должен иметь формат `Array of string` и значение - `top, person`.



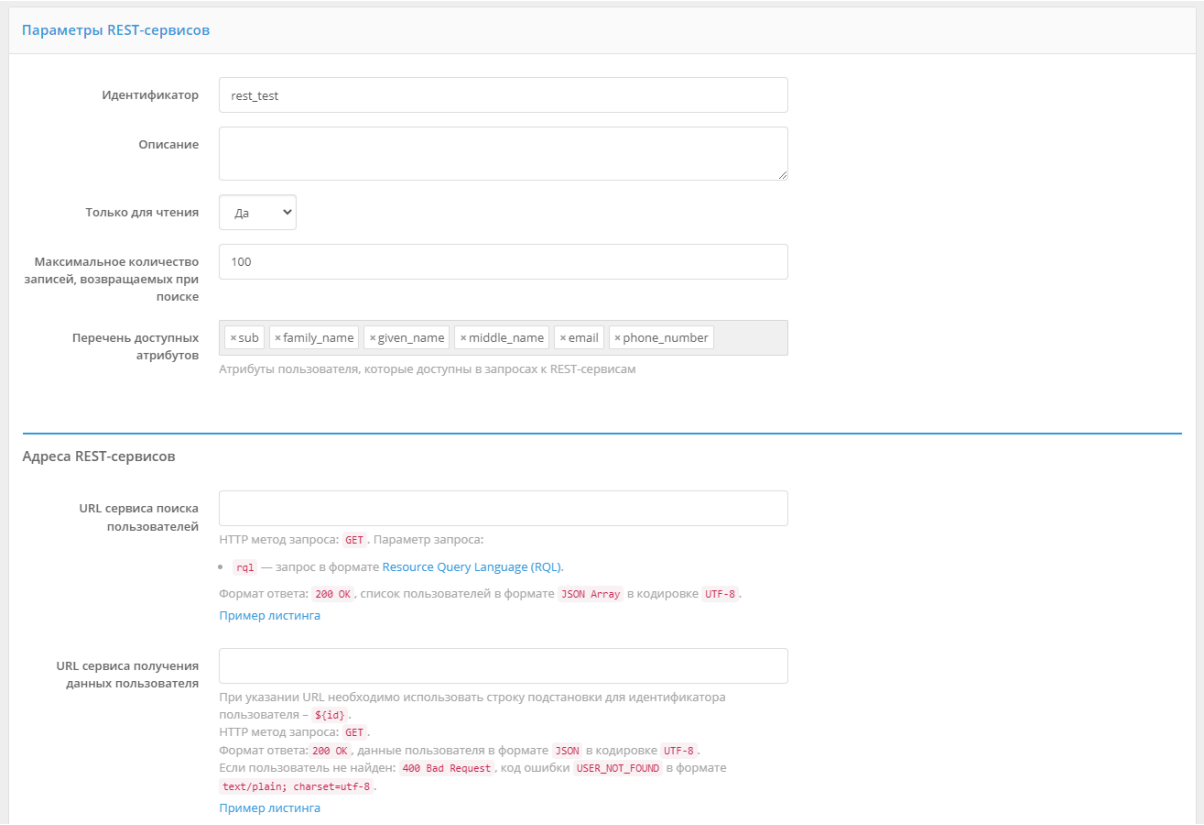
Подключение к хранилищу по REST

Если в качестве источника учетных записей пользователей используется внешняя база данных (не LDAP-хранилище), то для подключения к ней требуется разработать коннектор. Коннектор обеспечивает чтение (или изменение) необходимых данных из базы данных и предоставляет данные в корректном формате в виде REST-сервисов для Blitz Identity Provider.

Для настройки взаимодействия с REST-сервисами необходимо выполнить следующие шаги:

- добавить новое хранилище, указав тип добавляемого хранилища - `REST`;
- указать описание хранилища (опционально);
- указать, используется ли хранилище только для чтения данных или возможна запись в него;
- указать максимальное количество записей, возвращаемых при поиске;
- указать перечень доступных через REST-сервисы атрибутов;
- указать URL следующих сервисов:
  - сервис поиска пользователей;
  - сервис получения данных пользователя;
  - сервис проверки логина и пароля;
  - сервис смены пароля пользователем;
  - сервис добавления нового пользователя;
  - сервис изменения данных пользователя;
  - сервис удаления пользователя.

Скриншот страницы с настройками подключения к хранилищу с использованием REST-сервисов представлен ниже.



В следующих подразделах описаны требования к разработке REST-сервисов, предоставляющих необходимый Blitz Identity Provider доступ к хранилищу учетных записей.

Сервис поиска пользователей

Сервис поиска пользователей должен обрабатывать запросы методом `GET`, где в качестве параметра `rql` указывается поисковый запрос. Запрос имеет формат Resource Query Language (RQL)[25] и должен как минимум поддерживать следующие операции:

- `limit` – количество возвращаемых записей;
- `and` – одновременное выполнение поисковых условий;
- `or` – альтернативное выполнение поисковых условий (например, поиск по разным атрибутам в качестве логина);
- `in` – вхождение значения атрибута в список значений (например, поиск привязанных учетных записей при входе через внешний поставщик идентификации);
- `eq` – проверка условия равенства с возможностью поиска по маске (например, с использованием звездочки (*)).

Например, если в качестве логина в разделе Аутентификация настроен только поиск по атрибуту `email`, то передаваемый при аутентификации RQL-параметр будет иметь вид (где `test@mail.com` – данные, введенные пользователем в качестве логина):

```
rql=and(eq(email,test@mail.com),limit(10))
```

Если выполняется вход через внешний поставщик идентификации, и надо найти связанные с внешней учетной записью учетные записи в хранилище, то передаваемый RQL параметр будет иметь вид:

```
rql=and(in(sub,(7d5fd1d2-e171-4c85-8da6-00368863c396,2b78a2da-241c-4182-ba9b-
↪d810cdb7aa70)),limit(10))
```

Если в качестве логина настроен поиск по атрибуту `email` ИЛИ `sub`, то передаваемый RQL-параметр будет иметь вид:

```
rql=and(or(eq(sub,test@mail.com),eq(email,test@mail.com)),limit(10))
```

Сервис должен возвращать список пользователей и их данные в формате JSON в кодировке UTF-8. По каждому пользователю должны быть возвращены атрибуты:

- `id` – идентификатор пользователя в подключенной базе данных. Предполагается, что этот идентификатор будет неизменным для данного пользователя;
- `attrs` – объект с перечнем возвращаемых данных пользователя. Необходимо возвращать те атрибуты, которые предполагается использовать в системе и которые сконфигурированы в разделе Источники данных.

Пример запроса:

```
GET /users/search?rql=and(eq(sub,BIP*),limit(10)) HTTP/1.1
Host: idstore.identityblitz.com
Content-Type: application/json
Cache-Control: no-cache
```

Пример ответа:

[25] https://github.com/kriszyp/rql

```
[
  {
    "id": "ID123",
    "attrs": {
      "sub": "BIP123",
      "given_name": "Ivan",
      "family_name": "Ivanov",
      "email": "ivanov@test.org",
      "phone_number": "+79991234567"
    }
  },
  {
    "id": "ID456",
    "attrs": {
      "sub": "BIP456",
      "given_name": "Elena",
      "family_name": "Ivanova",
      "email": "ivanova@test.org",
      "phone_number": "+79997654321"
    }
  }
]
```

Сервис получения данных пользователя

В ряде случаев Blitz Identity Provider запрашивает данные конкретного пользователя. Сервис получения данных пользователя должен обрабатывать запросы методом `GET`, в котором в URL указывается атрибут `id` – внутренний идентификатор пользователя в подключенной базе данных. При задании URL этого сервиса в консоли управления необходимо использовать строку подстановки для идентификатора пользователя – `${id}`, например:

```
https://idstore.identityblitz.com/users/${id}
```

Если пользователь найден, то сервис должен отвечать `200 OK` и возвращать данные пользователя в формате JSON в кодировке UTF-8. Если пользователь не найден: `400 Bad Request`, код ошибки `USER_NOT_FOUND` в формате `text/plain; charset=utf-8`.

Пример запроса:

```
GET /users/ID123 HTTP/1.1
Host: idstore.identityblitz.com
Content-Type: application/json
Cache-Control: no-cache
```

Пример ответа, если пользователь найден:

```
HTTP/1.1 200 OK
Date: Mon, 18 Jul 2016 12:28:59 GMT
Content-Type: application/json; charset=utf-8

{
  "id": "ID123",
  "attrs": {
    "sub": "BIP123",
    "given_name": "Ivan",
    "family_name": "Ivanov",
    "email": "ivanov@test.org",
    "phone_number": "+79991234567"
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  }
}
```

Ответ для случая, если пользователь не найден:

```
HTTP/1.1 400 Bad Request
Date: Mon, 18 Jul 2016 12:28:59 GMT
Content-Type: text/plain; charset=utf-8

USER_NOT_FOUND
```

Сервис проверки логина и пароля

Сервис проверки логина и пароля должен обрабатывать запросы методом `POST`, в теле которых указаны следующие параметры (в формате `application/x-www-form-urlencoded`):

- `id` – внутренний идентификатор пользователя в подключенной базе данных;
- `password` – пароль.

В случае успеха сервис должен вернуть ответ `200 OK`.

При невозможности провести аутентификацию сервис должен вернуть `400 Bad Request` с одной из следующих ошибок:

- `INVALID_CREDENTIALS` – неверный логин или пароль пользователя;
- `UNWILLING_TO_PERFORM` – пользователь заблокирован;
- `INAPPROPRIATE_AUTHENTICATION` — пользователь не может быть аутентифицирован по паролю;
- `PASSWORD_EXPIRED` — пароль пользователя устарел.

Пример запроса:

```
POST /users/bind HTTP/1.1
Host: idstore.identityblitz.com
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

id=ivanov&password=12345678
```

Пример ответа (успешная проверка логина и пароля):

```
HTTP/1.1 200 OK
Date: Mon, 18 Jul 2016 12:38:53 GMT
Content-Type: application/json; charset=utf-8
```

Пример ответа (неверный логин и/или пароль):

```
HTTP/1.1 400 Bad Request
Date: Mon, 18 Jul 2016 12:38:53 GMT
Content-Type: text/plain; charset=utf-8

INVALID_CREDENTIALS
```

Сервис смены пароля пользователем

Сервис смены пароля пользователем должен обрабатывать запросы методом `POST`, в теле которых указаны следующие параметры (в формате `application/x-www-form-urlencoded`):

- `id` – идентификатор пользователя, полученный по результату операции проверки пароля пользователя;
- `old_password` – старый пароль;
- `new_password` – новый пароль.

В случае успеха сервис должен вернуть ответ `200 OK`.

В случае ошибки сервис должен вернуть `400 Bad Request` с одной из следующих ошибок:

- `INVALID_CREDENTIALS` — пользователь с данным идентификатором и паролем не найден;
- `UNWILLING_TO_PERFORM` — пользователь заблокирован;
- `CONSTRAINT_VIOLATION` — новый пароль не соответствует политикам безопасности.

Остальные возвращаемые ошибки должны быть аналогичны операции по проверке логина и пароля.

Пример запроса:

```
POST /users/changePassword HTTP/1.1
Host: idstore.identityblitz.com
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

id=ivanov&old_password=12345678&new_password=0987654321
```

Пример ответа:

```
HTTP/1.1 400 Bad Request
Date: Mon, 18 Jul 2016 12:43:23 GMT
Content-Type: text/plain; charset=utf-8

CONSTRAINT_VIOLATION
```

Сервис добавления нового пользователя

Сервис добавления нового пользователя должен обрабатывать запросы методом `PUT`, в теле которых указаны следующие параметры (в формате `application/json`):

- `password` – пароль пользователя (опционально);
- `attrs` – атрибуты пользователя.

В случае успеха сервис должен вернуть данные пользователя в формате JSON в кодировке UTF-8.

Если пароль не удовлетворяет политикам безопасности, сервис должен вернуть `400 Bad Request` с ошибкой `CONSTRAINT_VIOLATION`.

Если такой пользователь уже существует, сервис должен вернуть `400 Bad Request` с ошибкой `USER_ALREADY_EXISTS` и уточнением, что пользователь с данным идентификатором уже существует.

Пример запроса:

```
PUT /users HTTP/1.1
Host: idstore.identityblitz.com
Content-Type: application/json
Cache-Control: no-cache
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{
    "password":"********",
    "attrs": {
            "sub": "ivanov@test.org",
            "email": "ivanov@test.org"
    }
}
```

Пример ответа (пользователь создан):

```
HTTP/1.1 200 OK
Date: Mon, 18 Jul 2016 12:28:53 GMT
Content-Type: application/json; charset=utf-8

{
    "id": "ID678",
    "attrs": {
        "sub": "ivanov@test.org",
        "email": "ivanov@test.org"
    }
}
```

Пример ответа (учетная запись уже зарегистрирована):

```
HTTP/1.1 400 Bad Request
Date: Mon, 18 Jul 2016 12:43:23 GMT
Content-Type: text/plain; charset=utf-8

USER_ALREADY_EXISTS:ivanov@test.org
```

Сервис изменения данных пользователя

Сервис изменения данных пользователя должен обрабатывать запросы методом `POST`, в URL вызываемого сервиса указывается атрибут `id` – внутренний идентификатор пользователя в подключенной базе данных. При задании URL этого сервиса в консоли управления необходимо использовать строку подстановки для идентификатора пользователя – `${id}`, например:

```
http://idstore.identityblitz.com/users/${id}
```

В теле запроса на изменение данных указаны следующие параметры (в формате `application/json`):

- `password` – новое значение пароля пользователя (если пароль не передан, то он не должен измениться);
- `replaced` – новые значения атрибутов пользователя, которые нужно заменить или добавить;
- `deleted` – список названий удаляемых атрибутов.

В случае успеха сервис должен вернуть данные пользователя в формате JSON в кодировке UTF-8.

Если новый пароль не удовлетворяет политикам безопасности, сервис должен вернуть `400 Bad Request` с ошибкой `CONSTRAINT_VIOLATION`.

Если такой пользователь не существует, сервис должен вернуть `400 Bad Request` с ошибкой `USER_NOT_FOUND`.

Пример запроса:

```
POST /users/ID123 HTTP/1.1
Host: idstore.identityblitz.com
Content-Type: application/json
Cache-Control: no-cache

{
    "replaced": {
        "email": "ivanov@domain.org"
    },
    "deleted": ["family_name"],
    "password": "########"
}
```

Пример ответа:

```
HTTP/1.1 200 OK
Date: Mon, 18 Jul 2016 12:38:53 GMT
Content-Type: application/json; charset=utf-8

{
    "id": "ID123",
    "attrs": {
        "sub": "BIP123",
        "given_name": "Ivan",
        "email": "ivanov@domain.org"
    }
}
```

Сервис удаления пользователя

Сервис удаления учетной записи пользователя должен обрабатывать запросы методом `DELETE`, в URL вызываемого сервиса указывается атрибут `id` – внутренний идентификатор пользователя в подключенной базе данных. При указании URL этого сервиса необходимо использовать строку подстановки для идентификатора пользователя – `${id}`, например:

```
http://idstore.identityblitz.com/users/${id}
```

В случае успеха сервис должен вернуть статус `200 OK`.

Если пользователь не существует, сервис должен вернуть `400 Bad Request` с ошибкой `USER_NOT_FOUND`.

Пример запроса:

```
DELETE /users/ID123 HTTP/1.1
Host: idstore.identityblitz.com
Content-Type: application/json
Cache-Control: no-cache
```

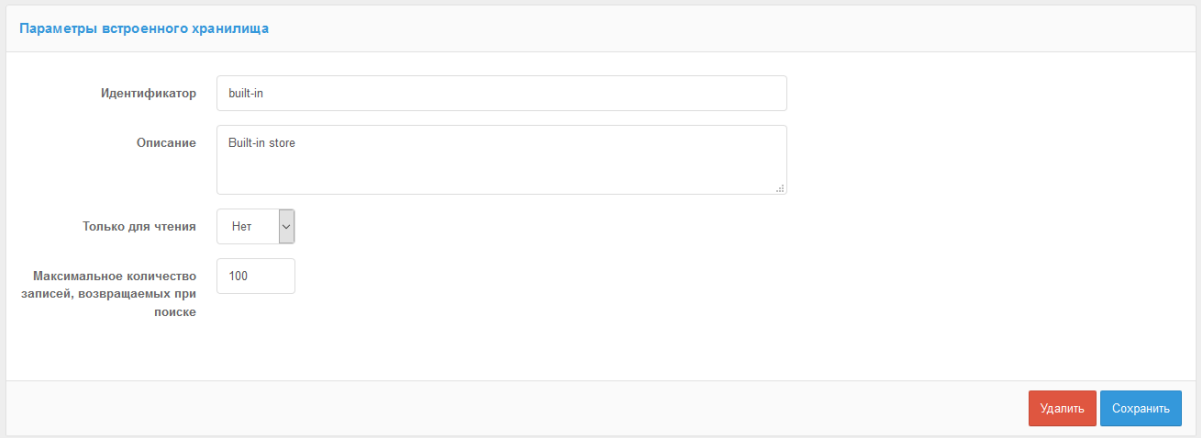
Пример ответа:

```
HTTP/1.1 200 OK
Date: Mon, 18 Jul 2016 12:28:53 GMT
Content-Type: application/json; charset=utf-8
```

Настройка внутреннего хранилища

Если в качестве источника учетных записей пользователей используется база данных Blitz Identity Provider, то необходимо выполнить следующие шаги:

- добавить новое хранилище, указав тип добавляемого хранилища – `BUILT-IN`;
- указать идентификатор хранилища;
- дать описание хранилища;
- определить, используется ли хранилище только для чтения или нет;
- указать максимальное число возвращаемых учетных записей при поиске.



Примечание: В случае если в качестве СУБД используется PostgreSQL, то можно хранить любое число учетных записей. В случае если в качестве СУБД используется Couchbase Server, то внутреннее хранилище можно использовать для хранения небольшого числа учетных записей.

2.2.2 Аутентификация

Настройки аутентификации задаются в разделе Аутентификация консоли управления. Описание работы с настройками приведено в следующих разделах.

Как работать с настройками аутентификации

Настройки аутентификации задаются в разделе Аутентификация консоли управления. Настройки разделены по вкладкам:

Общие настройки

Общие настройки, определяющие прохождение аутентификации пользователями

Парольные политики

Настройки парольных политик

Ключи безопасности

Настройки ключей безопасности

Первый фактор

Настройки методов аутентификации, применяемых при первичной идентификации и аутентификации

Второй фактор

Настройки методов аутентификации, применяемых для подтверждения входа

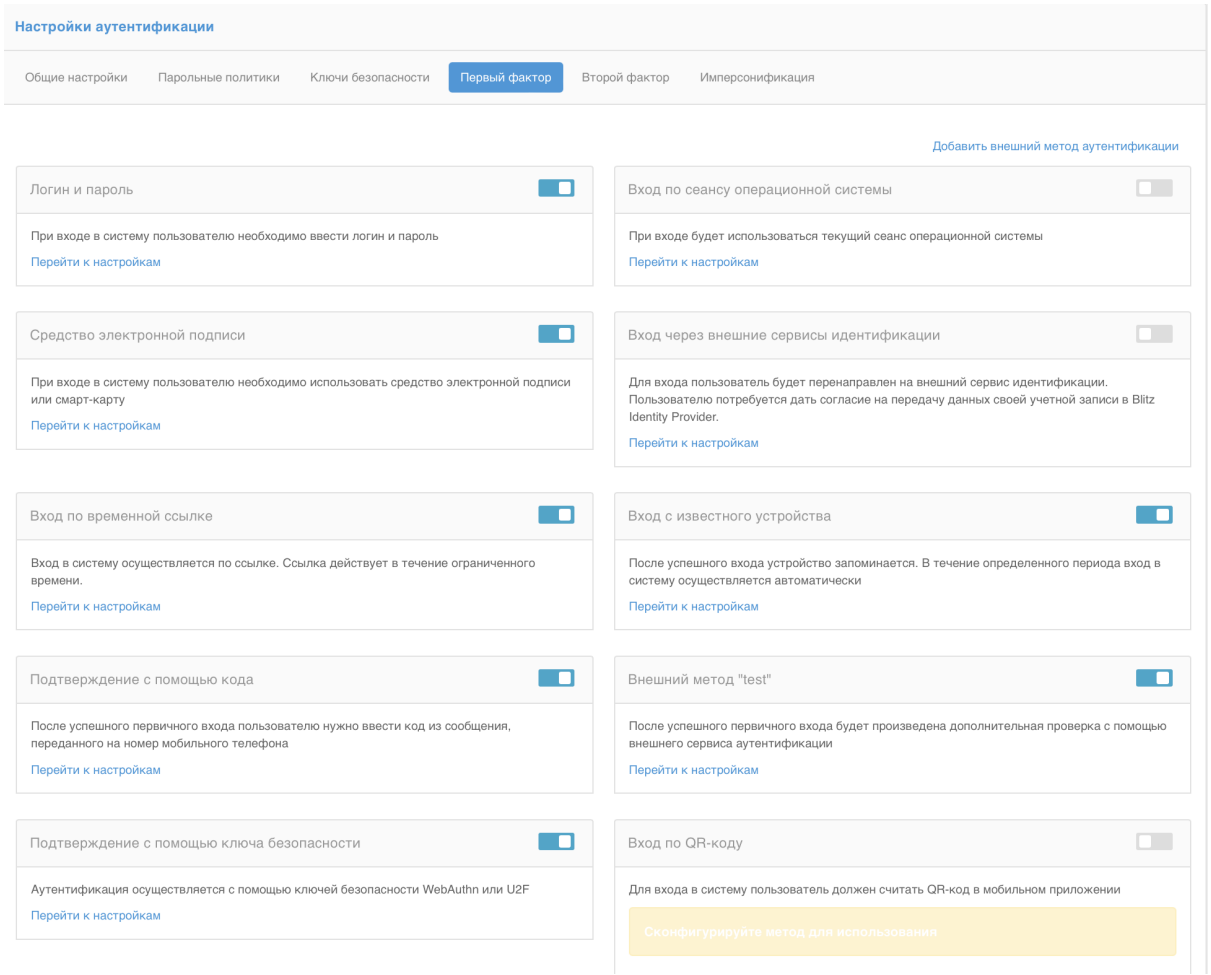
Третий фактор

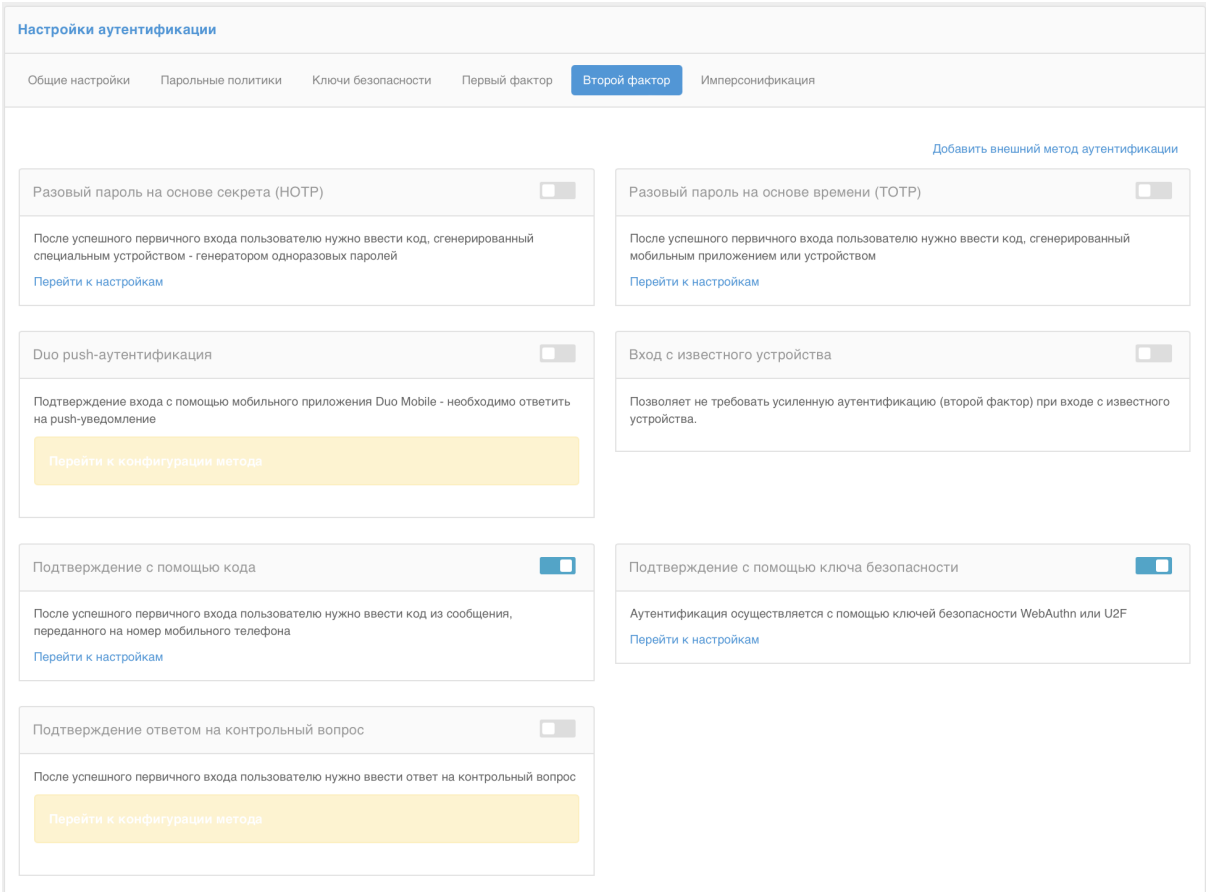
Опциональная вкладка, отображается, только если сконфигурировано наличие метода аутентификации, применяемого дополнительно после прохождения проверок первого и второго фактора

Методы аутентификации сгруппированы к первому и второму фактору. Чтобы включить метод аутентификации, его нужно сначала настроить.

Примечание: Второй фактор используется для «усиления» первого фактора, например, пользователю в дополнение к паролю требуется ввести специальный код, сгенерированный мобильным приложением

Набор методов может отличаться в зависимости от типа используемой лицензии. Для перехода к настройкам метода нужно нажать кнопку Перейти к конфигурации метода (при первичной настройке метода) либо ссылку Перейти к настройкам (для корректировки текущих заданных настроек).





Руководства по настройкам каждого метода приведены в последующих разделах. Для включения или отключения метода аутентификации необходимо установить переключатель в требуемое положение.

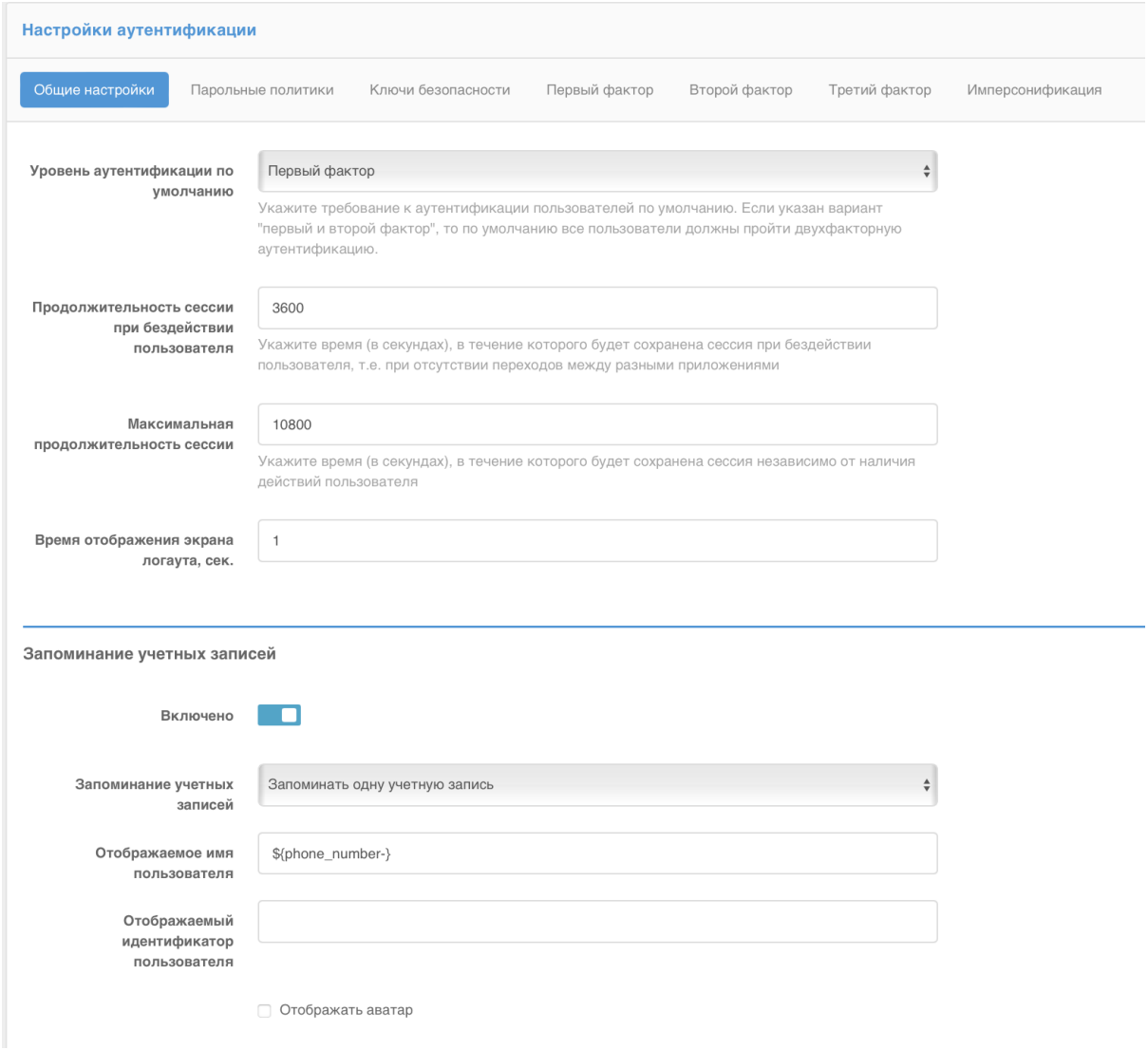
Общие настройки

На вкладке Общие настройки раздела Аутентификация можно задать:

- `Уровень аутентификации по умолчанию`: укажите Первый фактор, чтобы у пользователей запрашивалась только проверка первого фактора аутентификации (кроме пользователей, в настройках которых включена необходимость проверки второго фактора). Укажите Первый и второй фактор, чтобы для пользователей дополнительно к первому фактору требовалась проверка второго фактора аутентификации.
- Параметры продолжительности сессии:
  - `Продолжительность сессии при бездействии пользователя`: укажите время (в секундах), в течение которого будет сохранена сессия при бездействии пользователя, т.е. при отсутствии переходов между разными приложениями.
  - `Максимальная продолжительность сессии`: укажите время (в секундах), в течение которого будет сохранена сессия независимо от наличия действий пользователя.

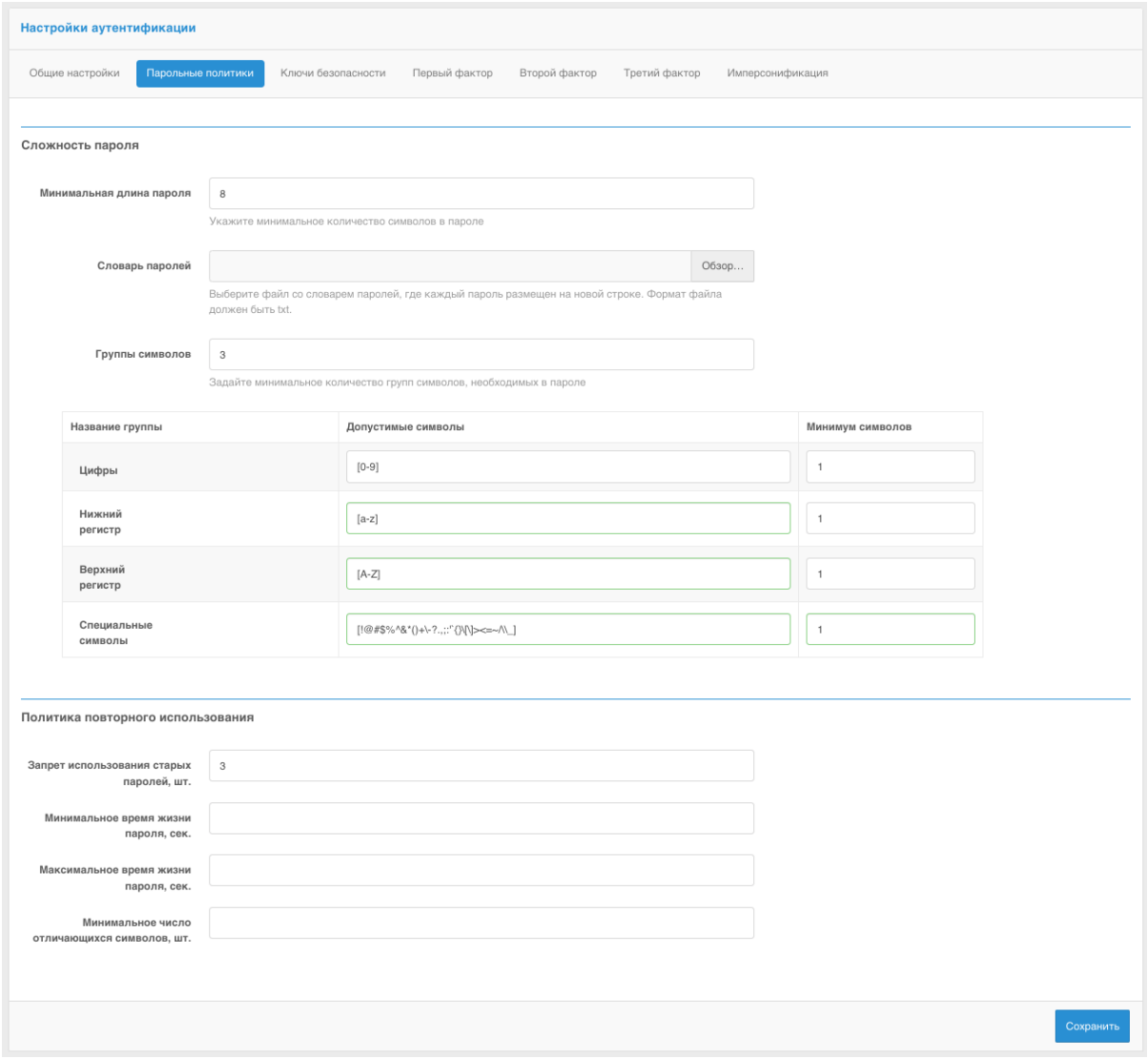
> **Внимание:** На продолжительность SSO-сессии пользователя также может влиять срок действия cookie `blc` на стороне Blitz Identity Provider. По умолчанию срок действия cookie `blc` составляет 10800 секунд. Если максимальная продолжительность сессии превышает данное значение, у пользователя может быть запрошен повторный вход, как только срок действия cookie истечет, даже при активной SSO-сессии. В этом случае необходимо *внести изменения* (страница 363) в файл конфигурации.

- Время отображения экрана логаута, сек.: время в секундах, в течение которого пользователю будет показываться экран выхода до автоматического перенаправления пользователя на страницу перехода в приложение после логаута.
- Настройте запоминание учетных записей:
  - По умолчанию запоминание учетных записей включено. При необходимости отключите его.
  - Запоминание учетных записей: режим запоминания учетных записей. Укажите Запоминать одну учетную запись, чтобы каждый вход новой учетной записью в браузере перезаписывал запомненный вход предыдущей учетной записи или Запоминать все учетные записи, чтобы каждый вход новой учетной записи добавлял к списку запомненных учетных записей в браузере еще одну.
  - Отображаемое имя пользователя: имя пользователя, которое отображается на странице входа. Задается в виде регулярного выражения, например: ${family_name-} ${given_name-}. Такое регулярное выражение позволяет отображать фамилию и имя пользователя, сохраненные в атрибутах family_name и given_name.
  - Отображаемый идентификатор пользователя: идентификатор учетной записи, который отображается второй строчкой на странице входа. Задается в виде регулярного выражения, например: ${email-$phone_number}. Такое регулярное выражение позволяет отображать один из контактов, сохраненных в атрибутах email или phone_number (если имеются оба, то отображается email). При настройке можно использовать маскирование значений. Например, правило ${phone_number&maskInMiddle(3,3)} будет отображать средние числа номера телефона в виде *.
  - Отображать аватар: признак необходимости отображать аватар на странице входа.



Парольные политики

Парольные политики настраиваются на вкладке Парольные политики раздела Аутентификация консоли управления.



Предусмотрены следующие настройки:

- Минимальная длина пароля – число символов в пароле (рекомендуется не менее 8);
- Словарь паролей – указывается текстовый файл, содержащий список запрещенных паролей. Каждый пароль должен быть на отдельной строке. В случае использования больших файлов рекомендуется загружать их непосредственно на сервер, и задавать путь к файлу в настройке `dicPath` в блоке настроек `blitz.prod.local.idp.password-policy` в файле `blitz.conf`.
- Группа символов – задает минимально необходимое количество групп символов в пароле. По каждой группе символов можно задать настройки в таблице групп символов:
  - Допустимые символы – с помощью регулярного выражения задается множество символов группы. Например, можно расширить допустимые символы цифр, изменив регулярное выражение на следующее – `[0-9٠-٩]`, можно расширить допустимые наборы символов букв – `[a-zа-я]` и `[A-ZA-Я]`, добавить или убрать допустимые спецсимволы – `[!@#$%^&*()+-?.,;:'`“{}[]><=~/_]`.
  - Минимум символов – сколько минимум символов из группы должно использоваться в пароле, что считалось, что группа задействована в пароле.
- Запрет использования старых паролей – настройка указывает, какое количество старых паролей должно запоминаться, чтобы при задании нового пароля не допускать ввод пароля из истории использованных паролей.
- Минимальное время жизни пароля – минимальное время жизни пароля, в секундах; пока это вре-

мя не истекло, пользователю не будет разрешено поставить новый пароль. Если такую проверку не следует выполнять, то нужно задать пустое значение настройки.

- Максимальное время жизни пароля – максимальное время жизни пароля, в секундах; как только это время истечет, пользователю потребуется задать новый пароль. Если такую проверку не следует выполнять, то нужно задать пустое значение настройки.
- Минимальное число отличающихся символов – сколько измененных символов должно быть в новом пароле по сравнению с предыдущим (для случаев, когда пользователь меняет текущий пароль на новый). Если такую проверку не следует выполнять, то нужно задать пустое значение настройки.

Работа с ключами безопасности

Настройка ключей безопасности

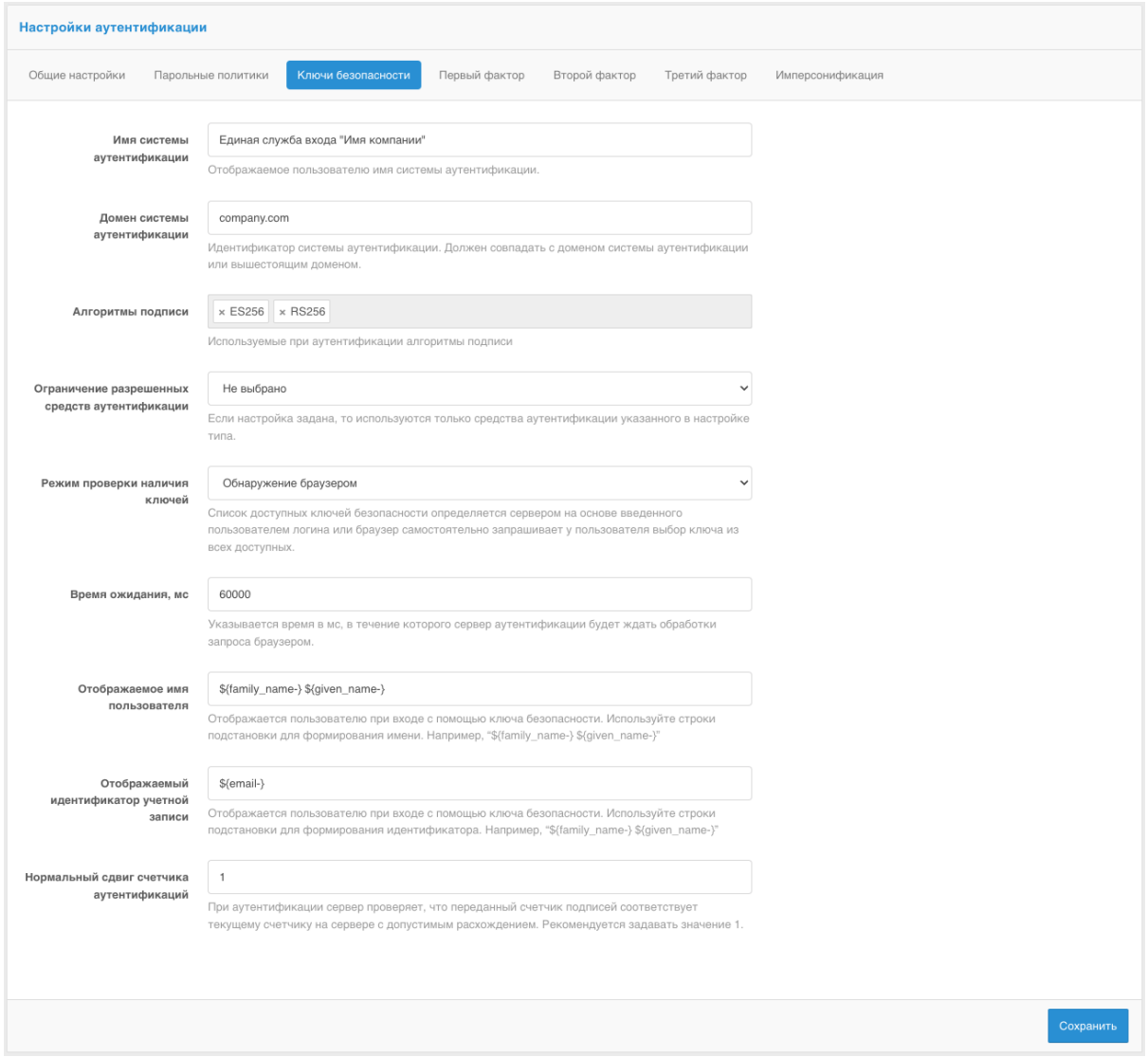
Blitz Identity Provider позволяет использовать для идентификации и аутентификации ключи безопасности (WebAuthn, Passkey, FIDO2, U2F). Для взаимодействия с ключами безопасности используется спецификация WebAuthn[26].

Поддерживаются следующие типы ключей:

- **Внешние ключи** – представляют собой аппаратные устройства в виде USB-ключей или брелоков, подключаемые к ПК, планшету и телефону с помощью USB-порта, Bluetooth или NFC. Для использования ключей не требуется установка на устройство драйверов, плагинов – взаимодействие с ключами осуществляется через встроенные возможности браузеров.
- **Встроенные ключи** – встроенные в устройстве и операционной системе механизмы аутентификации, поддерживающие WebAuthn:
  - Windows Hello – можно входить с помощью ПИН-кода Windows, проверки отпечатка пальца или распознавания лица;
  - Touch ID или пароль на MacBook;
  - Touch ID или Face ID на мобильном телефоне iOS или проверки отпечатка пальца или распознавания лица в Android.

Ключи безопасности настраиваются на вкладке Ключи безопасности раздела Аутентификация консоли управления.

[26] https://fidoalliance.org/fido2/



Предусмотрены следующие настройки:

- Имя системы аутентификации – необходимо задать подходящее для отображения пользователям имя системы аутентификации или имя приложения.
- Домен системы аутентификации – должен совпадать с доменом, используемым системой аутентификацией или быть вышестоящим доменом. На этот домен будут выпускаться ключи безопасности.
- Алгоритмы подписи – рекомендуется как минимум указать алгоритмы ES256 и RS256, чтобы обеспечивалась работа с Passkey, Windows Hello и большинством распространенных аппаратных FIDO2 и U2F ключей безопасности.
- Ограничение разрешенных средств аутентификации – при значении «Не выбрано» средства аутентификации не ограничиваются. Если выбрать «Переносные», то будут работать только аппаратные ключи безопасности (подключаемые по USB, Bluetooth или NFC). Если выбрать «Встроенные в платформу», то будут работать только встроенные в устройства ключи безопасности (Windows Hello, Touch ID на MacBook, Touch ID и Face ID в мобильных телефонах, а также использование телефона как средства аутентификации с подключением по Bluetooth).
- Режим проверки наличия ключа – при выборе «Обнаружение браузером» пользователю будут показываться все доступные на его устройстве для домена системы аутентификации ключи безопасности. При выборе «Обнаружение сервером» у пользователя будет запрошен логин, после чего будут показаны только те ключи, которые доступны на устройстве и привязаны к учетной записи пользователя на сервере.

- Время ожидания – указывается время в миллисекундах, в течение которого система аутентификации будет ожидать от браузера ответа на запрос обращения к ключу безопасности.
- Отображаемое имя пользователя – задает шаблон со строками подстановки, в соответствии с которым на странице входа по ключу безопасности в системе аутентификации отображается имя запомненного пользователя (актуально при использовании режиме «Обнаружение сервером»).
- Отображаемый идентификатор учетной записи – задает шаблон со строками подстановки, в соответствии с которым на устройстве пользователю показывается имя ключа безопасности.
- Нормальный сдвиг счетчика аутентификации – настройка, которая определяет, что сервер аутентификации будет сравнивать счетчик количества аутентификаций на устройстве со счетчиком количества аутентификаций этим же ключом на сервере и в случае расхождения более чем на число, указанное в счетчике, запретит использование ключа безопасности (защита от клонирования ключа).

Сервер аутентификации Blitz Identity Provider стандартно сконфигурирован таким образом, что в нем настроено доверие ко всем известным на момент выпуска текущей версии Blitz Identity Provider корневым и промежуточным сертификатам TPM модулей, FIDO, а также актуальным сертификатам Apple и Google, необходимым для проверки подписи аттестационных объектов FIDO2 и U2F. При необходимости *скорректируйте* (страница 348) разрешенные аттестационные сертификаты.

Использование ключей безопасности на первом и втором факторе описано в следующих разделах.

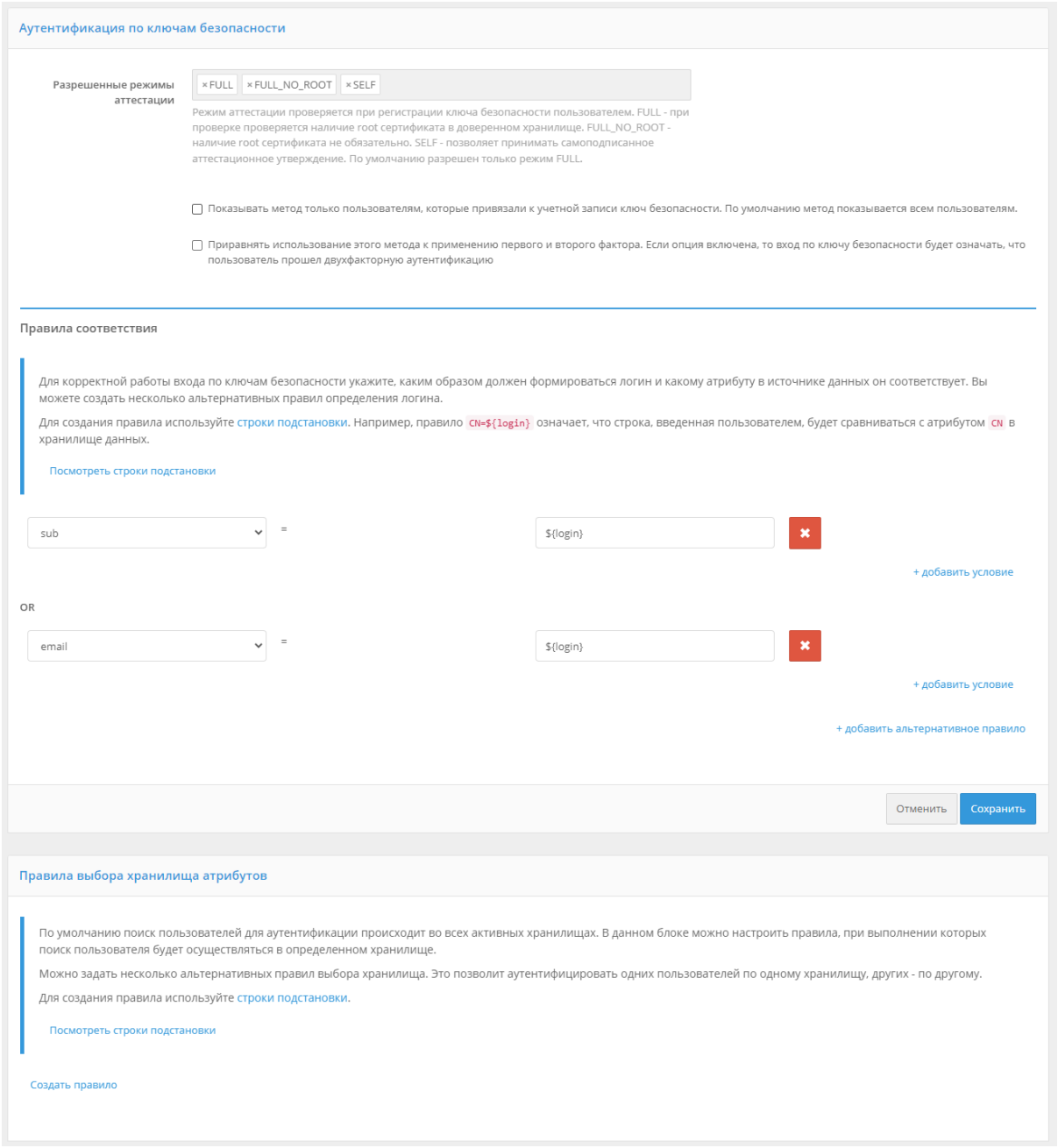
Вход с помощью WebAuthn, Passkey, FIDO2

Существует возможность использовать ключи безопасности (WebAuthn, Passkey, FIDO2[27]) для входа в Blitz Identity Provider.

Для настройки входа с помощью ключей безопасности необходимо задать следующие настройки на вкладке Первый фактор:

- Разрешенные режимы аттестации – использование только режимов `FULL` и `FULL_NO_ROOT` повысит безопасность, но не позволит использовать для входа некоторые ключи, а также ПИН-код Windows, так как при регистрации таких ключей аттестационный объект приходит без подписи производителя чипсета или ключа или с использованием самоподписанного ключа. Использование режима `SELF` позволяет атакующему реализовать атаку «человек по середине» на подмену ключа в момент регистрации, в случае если устройство пользователя контролируется атакующим.
- Показывать метод только пользователям, которые привязали к учетной записи ключ безопасности – если Blitz Identity Provider уже идентифицировал пользователя, то он уже знает, настроены ли для учетной записи пользователя ключи безопасности. Если ключи безопасности не настроены, то можно настроить, чтобы пользователю метод входа с помощью ключа безопасности не показывался.
- Приравнять использование этого метода к применению первого и второго фактора – если опция включена, то вход по ключу безопасности будет означать, что пользователь прошел двухфакторную аутентификацию.
- Правила соответствия – при входе по ключу безопасности пользователя просят ввести логин. Настройка правил соответствия позволяет указать правила поиска соответствия учетной записи введенному логину. Для найденной учетной записи будет запрошена проверка входа по ключу безопасности. Для создания правила используется строка подстановки: `${login}` – это строка, введенная пользователем в поле «логин». В результате, например, правило `email=${login}` означает, что строка, введенная пользователем, будет сравниваться с атрибутом `email` в хранилище данных.
- Правила выбора хранилища атрибутов – как и в случае входа по логину и паролю, по умолчанию поиск пользователей для аутентификации происходит во всех активных хранилищах. В блоке Правила выбора хранилища атрибутов можно *настроить правила* (страница 99), при выполнении которых поиск пользователя будет осуществляться в определенном хранилище.

[27] https://fidoalliance.org/fido2/



Подтверждение входа с помощью WebAuthn, Passkey, FIDO2, U2F

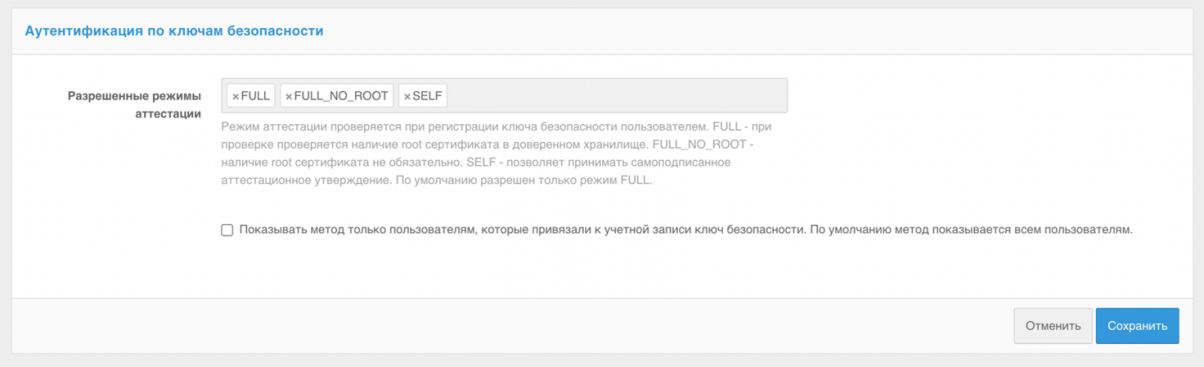
Существует возможность использовать ключи безопасности (WebAuthn, Passkey, FIDO2[28], U2F) для подтверждения входа в Blitz Identity Provider.

Для настройки подтверждения входа с помощью ключей безопасности необходимо задать следующие настройки на вкладке Второй фактор:

- Разрешенные режимы аттестации – использование только режимов `FULL` и `FULL_NO_ROOT` повысит безопасность, но не позволит использовать для входа некоторые ключи, а также ПИН-код Windows, так как при регистрации таких ключей аттестационный объект приходит без подписи производителя чипсета или ключа или с использованием самоподписанного ключа. Использование режима `SELF` позволяет атакующему реализовать атаку «человек по середине» на подмену ключа в момент регистрации, в случае если устройство пользователя контролируется атакующим.

[28] https://fidoalliance.org/fido2/

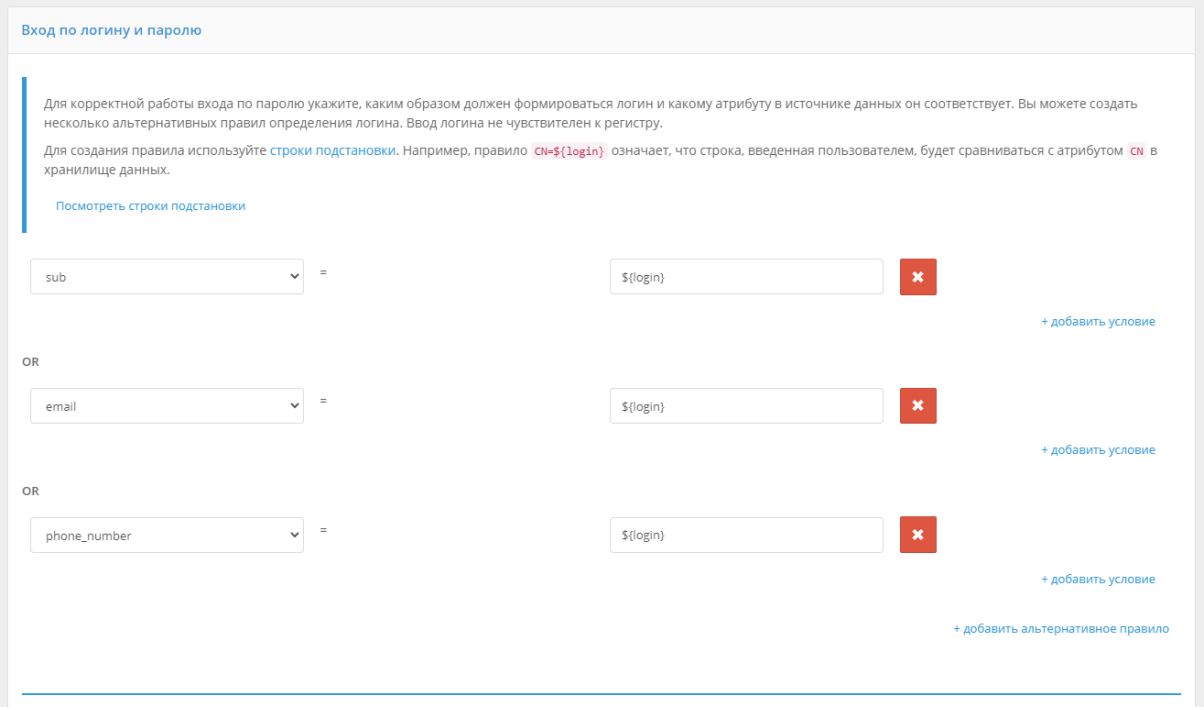
- Показывать метод только пользователям, которые привязали к учетной записи ключ безопасности – если ключи безопасности не настроены, то можно настроить, чтобы пользователю метод подтверждения входа с помощью ключа безопасности не показывался.



Вход по логину и паролю

Для использования входа по логину и паролю необходимо задать правила соответствия – каким образом определять, как введенный логин соотносится с пользователями в хранилище данных.

Для создания правила используется строка подстановки: `${login}` – это строка, введенная пользователем в поле «логин». В результате, например, правило «`email=${login}`» означает, что строка, введенная пользователем, будет сравниваться с атрибутом `email` в хранилище данных;



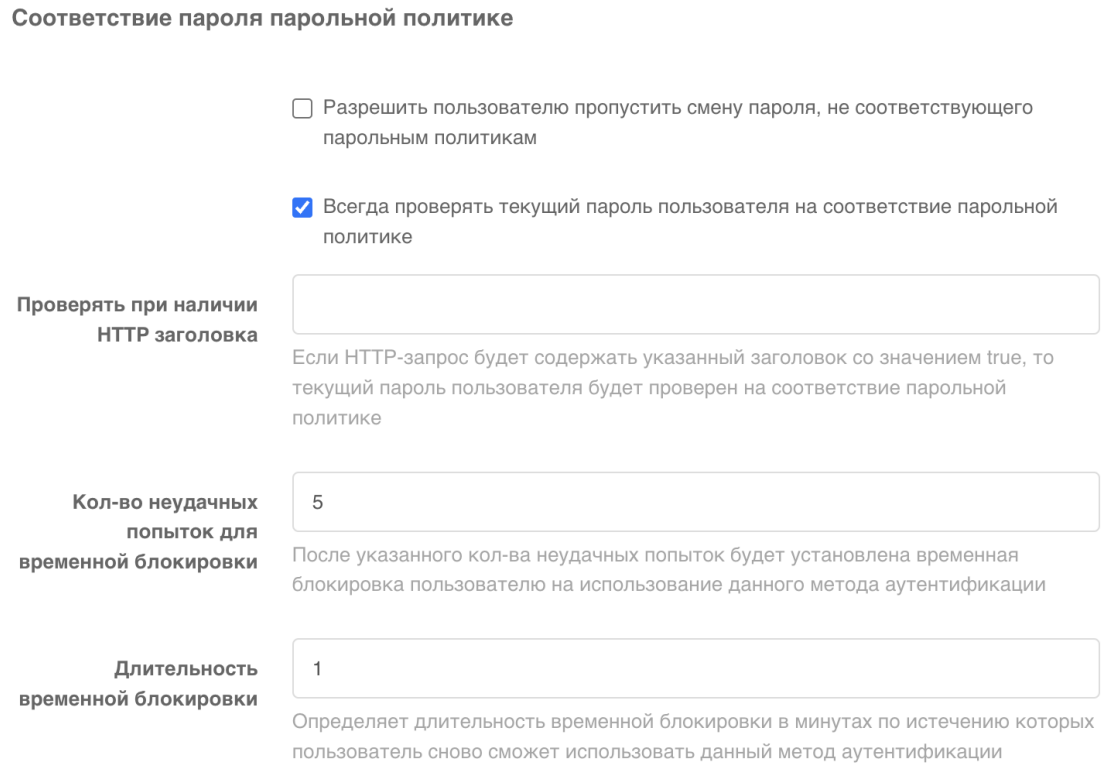
В настройках входа по логину и пароля можно включить проверку на соответствие пароля *парольной политике* (страница 312). Вводимый пользователем пароль будет в момент входа проверяться на соответствие парольной политике. В случае несоответствия пароля требованиям политики пользователь сможет задать новый пароль или пропустить этот шаг.

Для настройки проверки на соответствие пароля парольной политике при входе необходимо:

- выбрать опцию `Всегда проверять текущий пароль пользователя на соответствие парольной политике` или вписать имя некоторого заголовка в поле `Проверять при наличии HTTP заголовка` (в этом случае, если HTTP-запрос будет содержать указанный заголовок со

значением `true`, то текущий пароль пользователя будет проверен на соответствие парольной политике);

- опция `Разрешить пользователю пропустить смену пароля, не соответствующего парольным политикам` позволяет пользователю отказаться от смены пароля при входе;
- указать количество неудачных попыток для временной блокировки. После указанного количества неудачных попыток будет установлена временная блокировка пользователю на использование данного метода аутентификации;
- длительность временной блокировки (в минутах).



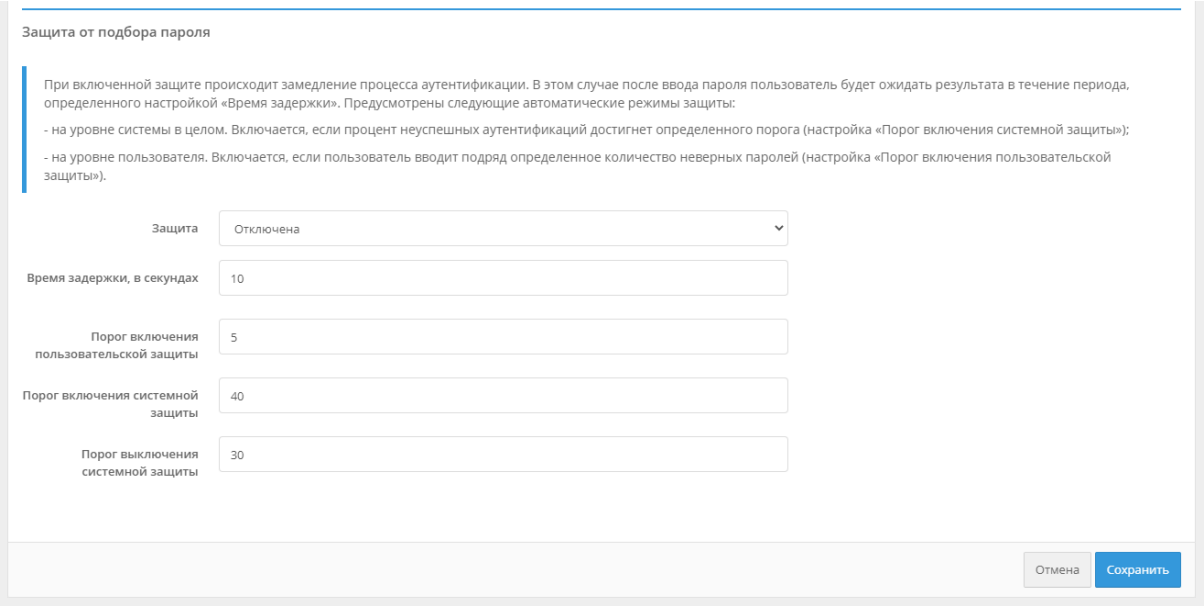
В настройках входа по логину и пароля можно управлять защитой от перебора пароля. При включенной защите замедляется проверка пароля. После ввода пароля пользователь будет ожидать проверки в течение заданного периода `Время задержки` (в секундах).

Администратор в настройке `Защита` может выбрать следующие режимы защиты:

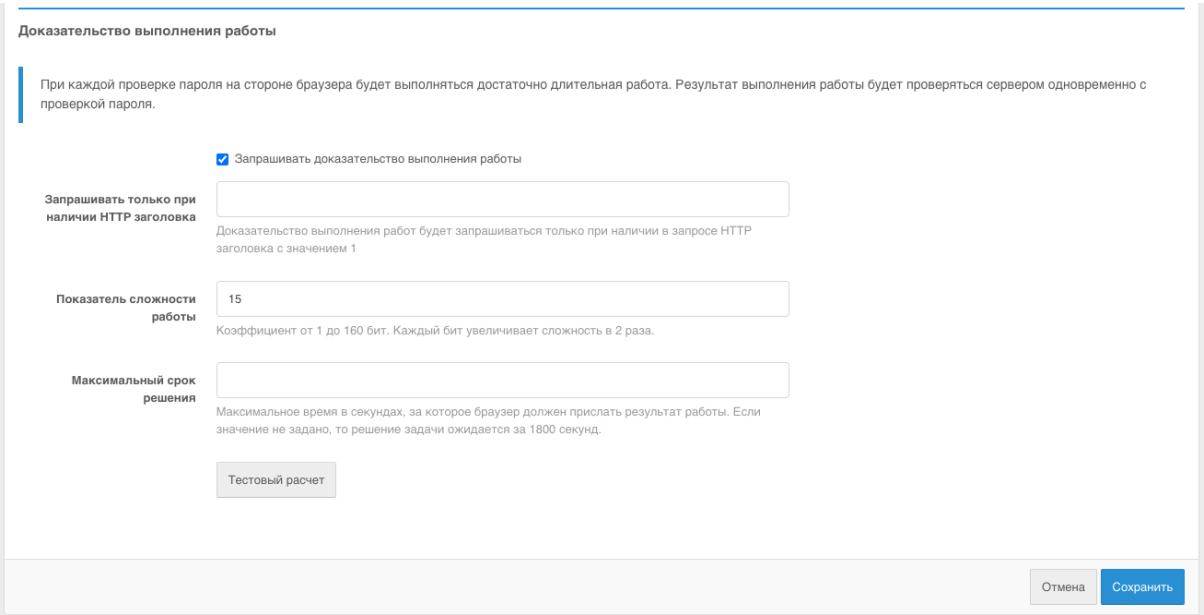
- `Автоматический режим на уровне системы и пользователей` – защита включится для всех пользователей, если доля неуспешных аутентификаций превысит «Порог включения системной защиты», и выключится, если доля неуспешных аутентификаций станет ниже «Порог выключения системной защиты»;
- `Автоматический режим на уровне пользователей` – защита сработает в отношении пользователей, по которым будет превышено число неуспешных проверок пароля, заданное настройкой «Порог включения пользовательской защиты»;
- `Задержка аутентификации для всех пользователей` – защита будет включена для всех пользователей;
- `Отключена` – защита будет выключена.

Параметры `Порог включения системной защиты` и `Порог выключения системной защиты` задаются в процентах, соответствующих доле неуспешных аутентификаций в общем числе попыток аутентификации.

Пример настройки защиты от подбора пароля представлен ниже.



Для усложнения автоматического подбора пароля можно включить в Blitz Identity Provider настройки `Доказательство выполнения работы`. Тогда при каждом входе по логину и паролю браузер пользователя должен будет выполнить вычислительно сложную задачу. Если не предоставить решение, предоставить неправильное решение или предоставить решение не вовремя, то Blitz Identity Provider вернет ошибку. В итоге нельзя будет понять, правильные ли логин и пароль.



В блоке настроек Доказательство выполнения работы можно настроить следующее:

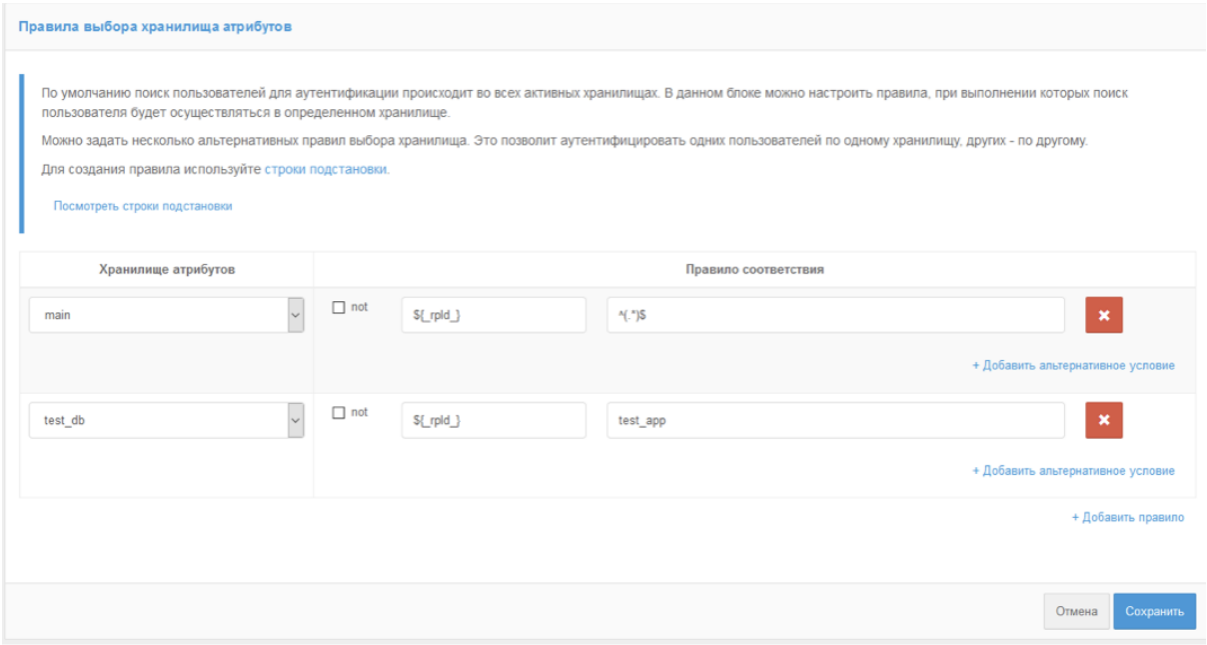
- включить настройку `Запрашивать доказательство выполнения работы`.
- при необходимости задать настройку `Запрашивать только при наличии HTTP заголовка` – это полезно, если нужно оставить возможность автотестам выполнять вход по паролю без необходимости прохождения проверки. В этом случае на веб-сервере нужно для пользовательских запросов настроить установку заголовка из этой настройки, а для запросов, приходящих от автотестов, заголовок не устанавливать.

- установить `Показатель сложности работы` – задается значение коэффициента от 1 до 160 бит. Каждый бит увеличивает сложность в 2 раза. Рекомендуется значение 15 бит.
- `Максимальный срок решения` – время в секундах, за которое браузер должен прислать результат работы. Если значение не задано, то решение задачи ожидается за 1800 секунд. Время отсчитывается с момента генерации задачи сервером в момент отображения страницы входа.

После установки настройки перед сохранением рекомендуется нажать кнопку Тестовый расчет, чтобы получить примерное представление о времени выполнения работы на текущем устройстве.

В блоке Правила выбора хранилища атрибутов можно настроить правила, при выполнении которых поиск пользователя будет осуществляться только в указанном хранилище. По умолчанию поиск пользователей для аутентификации происходит во всех активных хранилищах атрибутов. Можно задать несколько альтернативных правил выбора хранилища. Это позволит аутентифицировать одних пользователей через одно хранилище, а других – через другое. Для создания правила используются строки подстановки.

Например, на скриншоте ниже выполнена настройка, что при запросе входа приложением с идентификатором `test_app` логин и пароль пользователя будет проверяться по хранилищу `test_db`. Вход во все иные приложения будет производиться через хранилище `main`.



Вход с помощью средства электронной подписи

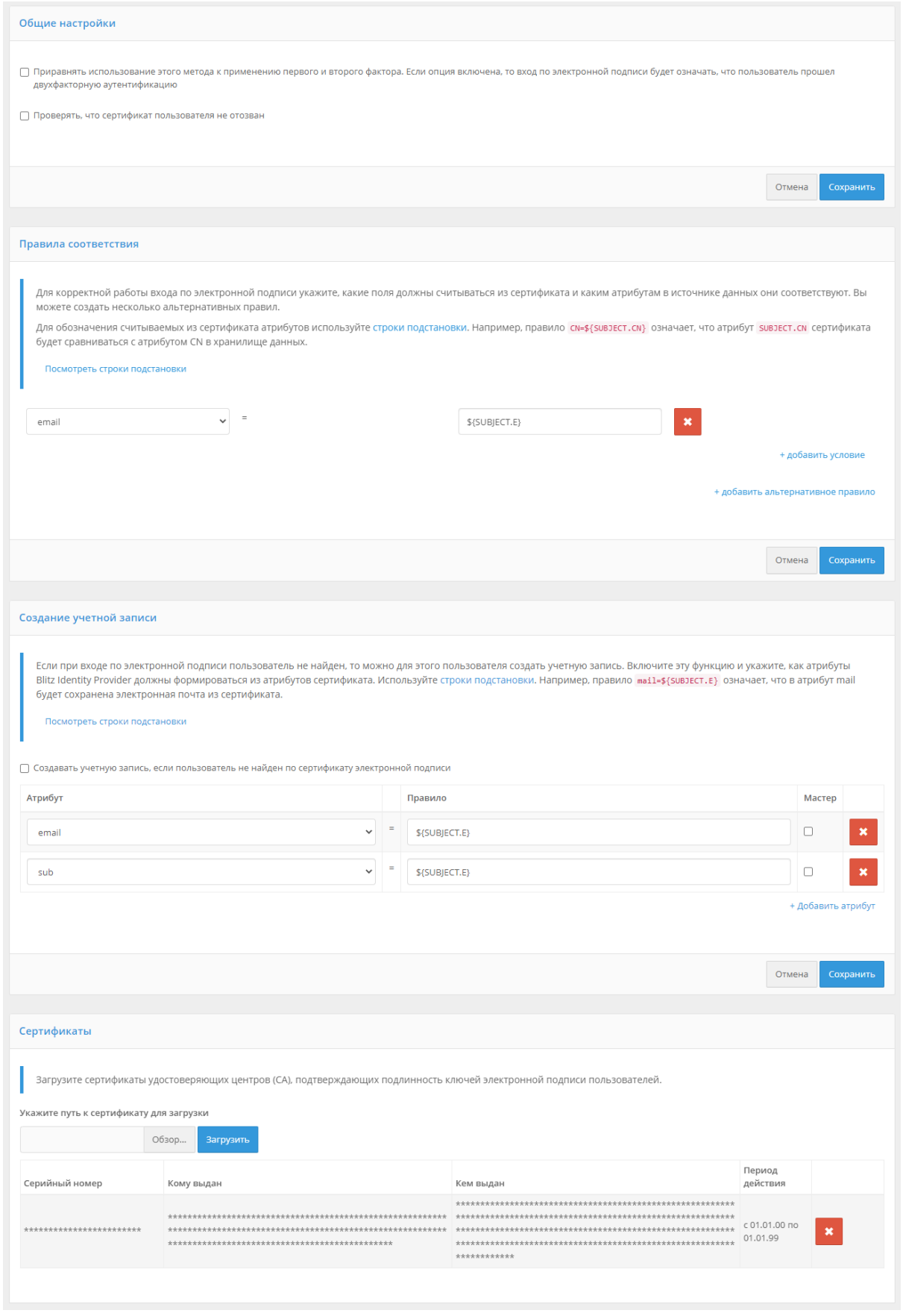
Настройка метода аутентификации в консоли управления

При использовании для аутентификации средства электронной подписи необходимо:

- в блоке настроек Сертификаты загрузить сертификаты удостоверяющих центров, подтверждающих подлинность сертификатов ключей электронной подписи или *настроить* (страница 316) взаимодействие с внешним сервисом проверки электронной подписи.
- настроить в блоке Правила соответствия параметры сопоставления учетной записи пользователя в хранилище по его атрибутам из сертификата электронной подписи. В правилах сопоставления используются строки подстановки. Например, правило `cn=${SUBJECT.CN}` означает, что атрибут `SUBJECT.CN` сертификата будет сравниваться с атрибутом `cn` в хранилище данных. Возможно указание нескольких условий одновременно, а также указание альтернативных правил.

При конфигурировании входа по электронной подписи можно также указать:

- следует ли этот метод использовать в качестве первого и второго фактора. Если да, то пользователь, прошедший аутентификацию по электронной подписи, будет считаться прошедшим двухфакторную аутентификацию (пример настройки на рисунке ниже);
- следует ли проверять действительность сертификата. В этом случае Blitz Identity Provider, используя указанную в сертификате точку распределения списка отзыва (CRL), будет проверять, не был ли сертификат отозван. Для активации этой возможности следует отметить чекбокс `Проверять, что сертификат пользователя не отозван`;
- следует ли создавать (регистрировать) учетную запись при первом входе по электронной подписи. В этом случае, если пользователь не найден по определенным правилам соответствия, то ему будет предложено зарегистрировать учетную запись. Чтобы включить эту функцию, следует отметить чекбокс `Создавать учетную запись, если пользователь не найден по сертификату электронной подписи` и настроить правила регистрации пользователя – каким образом заполнять атрибуты в хранилище из атрибутов сертификата. Для задания правил следует использовать строки подстановки. Например, правило `email=${SUBJECT.E}` означает, что в атрибут `email` будет сохранена электронная почта из сертификата электронной подписи пользователя.


Общие настройки
Приравнять использование этого метода к применению первого и второго фактора. Если опция включена, то вход по электронной подписи будет означать, что пользователь прошел двухфакторную аутентификацию
Проверять, что сертификат пользователя не отозван
Отмена
Сохранить
Правила соответствия
Для корректной работы входа по электронной подписи укажите, какие поля должны считываться из сертификата и каким атрибутам в источнике данных они соответствуют. Вы можете создать несколько альтернативных правил.
Для обозначения считываемых из сертификата атрибутов используйте строки подстановки. Например, правило CN=${SUBJECT.CN} означает, что атрибут SUBJECT.CN сертификата будет сравниваться с атрибутом CN в хранилище данных.
Посмотреть строки подстановки
email
=
${SUBJECT.E}
+ добавить условие
+ добавить альтернативное правило
Отмена
Сохранить
Создание учетной записи
Если при входе по электронной подписи пользователь не найден, то можно для этого пользователя создать учетную запись. Включите эту функцию и укажите, как атрибуты Blitz Identity Provider должны формироваться из атрибутов сертификата. Используйте строки подстановки. Например, правило mail=${SUBJECT.E} означает, что в атрибут mail будет сохранена электронная почта из сертификата.
Посмотреть строки подстановки
Создавать учетную запись, если пользователь не найден по сертификату электронной подписи
Атрибут
Правило
Мастер
email
=
${SUBJECT.E}
sub
=
${SUBJECT.E}
+ Добавить атрибут
Отмена
Сохранить
Сертификаты
Загрузите сертификаты удостоверяющих центров (CA), подтверждающих подлинность ключей электронной подписи пользователей.
Укажите путь к сертификату для загрузки
Обзор...
Загрузить
Серийный номер
Кому выдан
Кем выдан
Период действия
с 01.01.00 по 01.01.99

Использование и обновление плагина

Для корректной работы входа по электронной подписи на компьютерах пользователей используется специальный плагин – Blitz Smart Card Plugin. При первом входе по электронной подписи пользователю будет предложено установить плагин. После загрузки файла и его запуска пользователю следует пройти все шаги установки плагина. При повторном входе с данного устройства не потребуется устанавливать плагин заново.

Blitz Identity Provider поставляется вместе с версией плагина, позволяющей работать со средством электронной подписи в качестве метода аутентификации.

При необходимости обновить версию Blitz Smart Card Plugin следует заменить дистрибутивы плагина – они размещены в директории `assets` с установкой Blitz Identity Provider, в архиве `assets.zip`. Структура архива имеет следующий вид:

```
plugins/sc/deb/BlitzScPlugin.deb
plugins/sc/rpm/BlitzScPlugin.rpm
plugins/sc/win/BlitzScPlugin.msi
plugins/sc/mac/BlitzScPlugin.pkg
plugins/sc/mac/BlitzScPlugin-10.14.pkg
...
```

Необходимо распаковать архив `assets.zip`, заменить файлы с дистрибутивом плагина и заархивировать обратно файлы в `assets.zip`.

Также можно при необходимости *скорректировать* (страница 317) настройки использования плагина электронной подписи.

Вход через внешние сервисы идентификации

Перечень доступных внешних сервисов идентификации зависит от редакции Blitz Identity Provider и приобретенных опций.

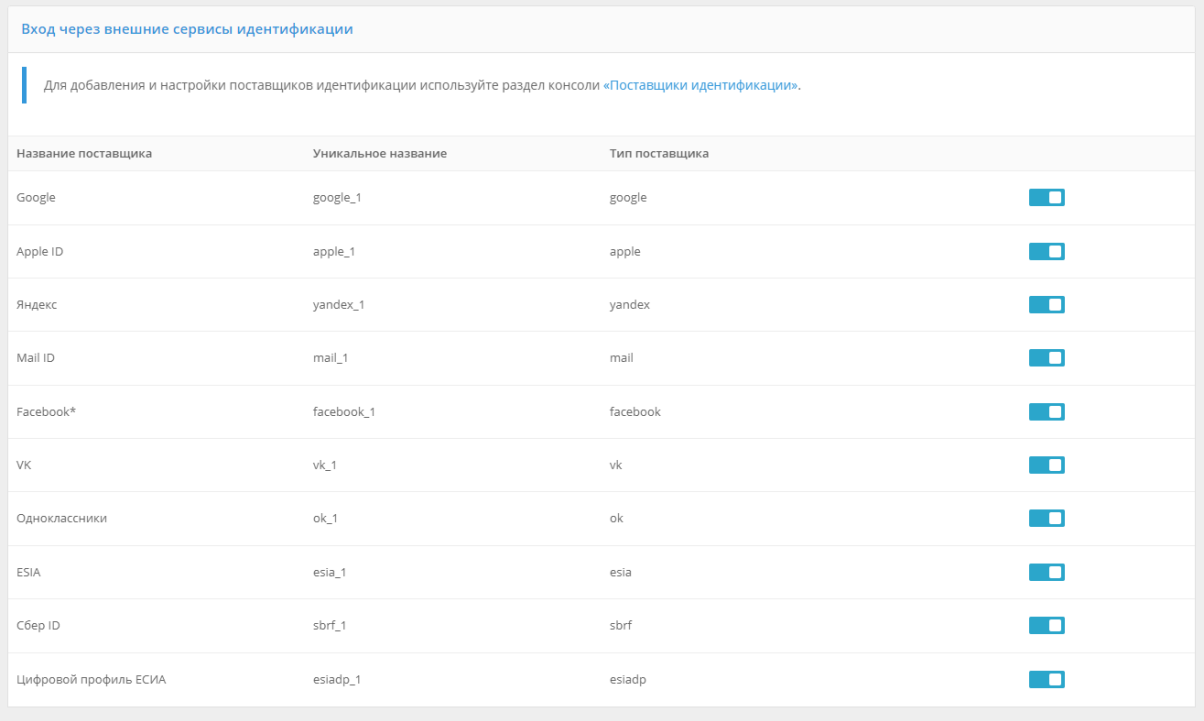
Возможен вход с использованием следующих внешних сервисов идентификации:

- Apple ID;
- Facebook[1];
- ВКонтакте;
- Яндекс;
- Google;
- Одноклассники;
- Mail ID;
- VK ID;
- ЕСИА (gosuslugi.ru);
- ЕСИА в режиме Цифровой профиль (gosuslugi.ru);
- Сбер ID;
- T-ID;
- ВТБ ID;
- СберБизнес ID;
- Альфа ID;
- Mos ID (СУДИР);
- поставщики идентификации, работающие по OpenID Connect;
- поставщики идентификации, работающие по SAML.

Подключения к внешним сервисам идентификации должны быть предварительно *сконфигурированы* (страница 143) в консоли управления на вкладке Поставщики идентификации.

В разделе настроек Вход через внешние сервисы идентификации необходимо выбрать, какие из настроенных поставщиков идентификации должны использоваться при входе.

[1] Meta признана экстремистской организацией и запрещена в России, деятельность принадлежащих ей соц.сетей Facebook и Instagram также запрещена в РФ.



Вход с помощью прокси-аутентификации

Прокси-аутентификация (аутентификация с помощью прокси-сервера) производится по данным, передаваемым в HTTP-заголовках.

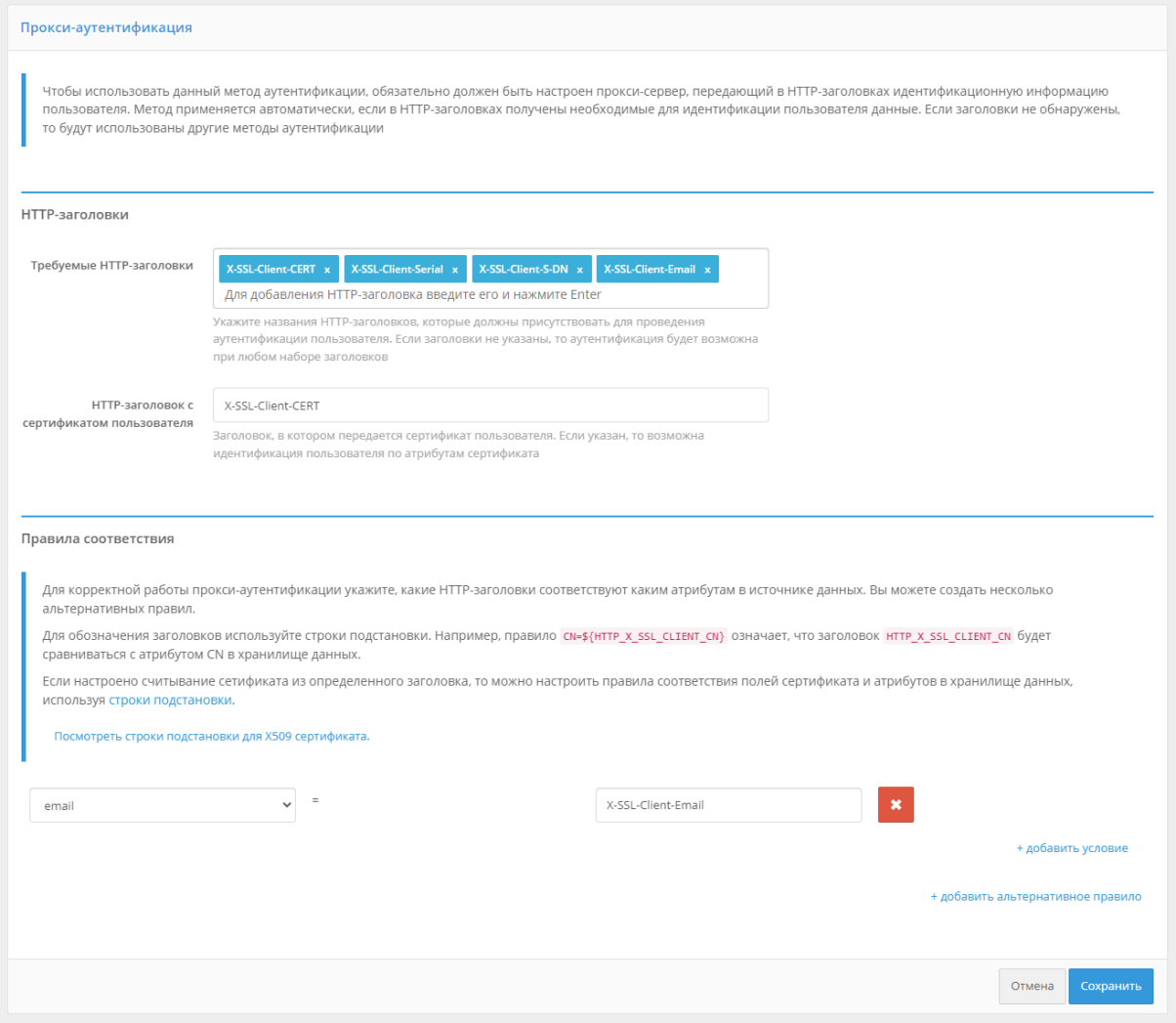
Важно: При включенной прокси-аутентификации Blitz Identity Provider производит только идентификацию пользователя, тогда как аутентификацию (в результате проверки сертификата) осуществляет прокси-сервер. Включение данного метода аутентификации допустимо в тех случаях, когда все пользователи обращаются к Blitz Identity Provider через прокси-сервер.

Для корректной работы метода необходимо указать:

- требуемые HTTP-заголовки – перечень HTTP-заголовков, которые должны присутствовать для прохождения прокси-аутентификации пользователя,
- HTTP-заголовок с сертификатом пользователя (опциональный параметр) – заголовок, содержащий x.509 сертификат пользователя,
- соответствие значений HTTP-заголовков и идентификационных данных пользователя в хранилище атрибутов.

Возможна настройка маппинга атрибутов сертификата, передаваемого в HTTP-заголовке, и данных пользователя в хранилище.

Пример настроек входа с помощью прокси-аутентификации представлен ниже:



Вход с помощью сеанса операционной системы

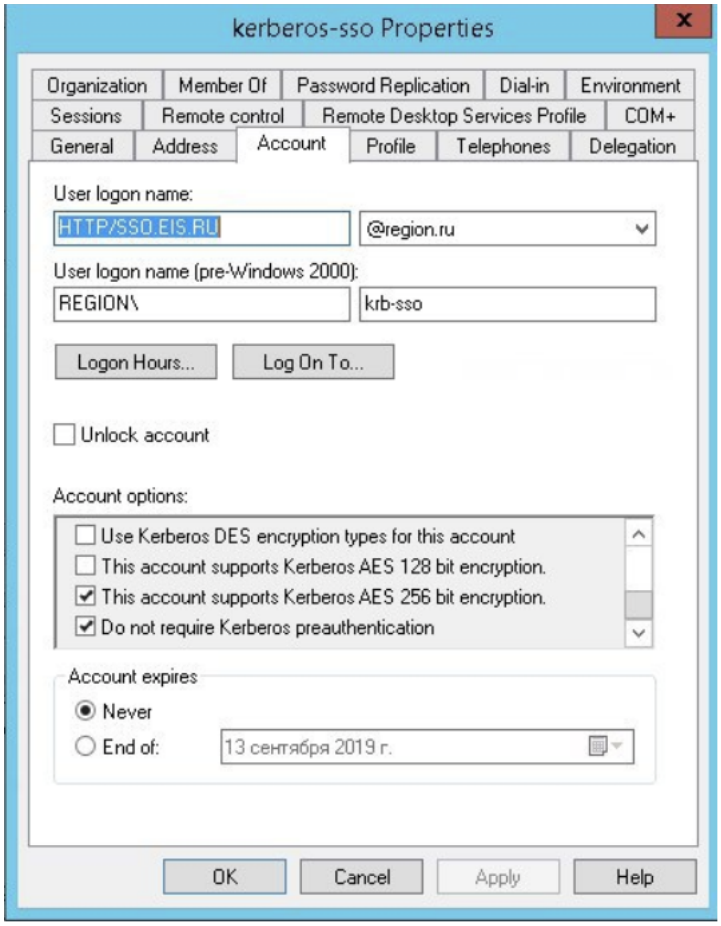
Способ входа с использованием сеанса операционной системы позволяет пользователям не проходить дополнительно идентификацию и аутентификацию в Blitz Identity Provider, если они ранее вошли со своего ПК в сеть организации и прошли идентификацию и аутентификацию в операционной системе (вошли в сетевой домен). Такие пользователи получат возможность сквозной идентификации при доступе ко всем приложениям, подключенным к Blitz Identity Provider.

Для входа с помощью сеанса операционной системы в организации должен быть развернут Kerberos-сервер (отдельно или в составе контроллера домена организации) и выполнены описанные ниже настройки.

Настройки контроллера домена (Kerberos-сервера)

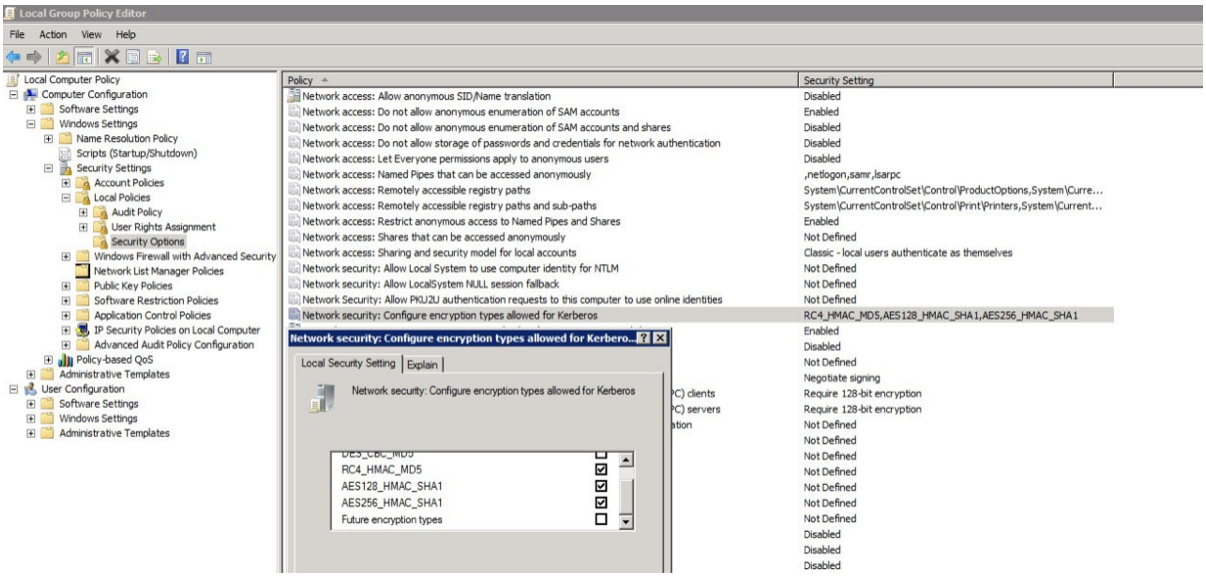
В контролере домена необходимо зарегистрировать учетную запись для сервера Blitz Identity Provider. Для созданной учетной записи нужно на странице Account в блоке Account options оснастки контроллера домена включить настройки `User cannot change password` и `Password never expires`.

Также следует отметить опции `This account supports Kerberos AES 256 bit encryption` и запретить предварительную аутентификацию `Do not require Kerberos preauthentication`.



В оснастке управления групповыми политиками следует настроить политику `Configure encryption types allowed for Kerberos`, указав следующие возможные значения: `RC4_HMAC_MD5`, `AES128_HMAC_SHA1` и `AES256_HMAC_SHA1`.

Пример настройки:



Далее необходимо создать Service Principal Name (SPN) для идентификации сервера Blitz Identity Provider сервером Kerberos. Это выполняется с помощью следующей команды:

```
ktpass -princ HTTP/idp.company.ru@DOMAIN.LOC -mapuser DOMAIN\blitzidpsrv -out C:\
↪temp\spnego_spn.keytab -mapOp set -crypto ALL -ptype KRB5_NT_PRINCIPAL /pass␣
↪SecretPassword
```

Параметры команды ktpass:

- значение параметра `mapuser` – имя созданной в домене учетной записи сервера Blitz Identity Provider, например, `DOMAIN\blitzidpsrv`;
- значение параметра `princ` – имя SPN сервера с Blitz Identity Provider для идентификации в среде Kerberos. Это имя состоит из имени хоста сервера с Blitz Identity Provider, имени Kerberos Realm в верхнем регистре (обычно совпадает с именем домена) и используемого транспортного протокола (HTTP). Пример значения SPN – `HTTP/idp.company.ru@DOMAIN.LOC`. Важно, чтобы `HTTP/` в начале имени SPN указывалось именно большими буквами, как в примере.
- параметр `mapOp` – если задан в значение `add`, то новый SPN будет добавлен к существующим. Если задано значение `set`, то SPN будет перезаписан.
- параметр `out` – задает путь к генерируемому keytab-файлу. Например, `C:\temp\spnego_spn.keytab`;
- параметр `/pass` – значение пароля от учетной записи сервера Blitz Identity Provider в домене.
- параметры `crypto` и `ptype` задают ограничения на используемые алгоритмы и тип генерируемой Kerberos-службы. Рекомендуется задать параметры как в указанном примере `-crypto ALL -ptype KRB5_NT_PRINCIPAL`.

Сгенерированный keytab-файл необходимо сохранить. Он будет необходим для последующей настройки в консоли управления Blitz Identity Provider.

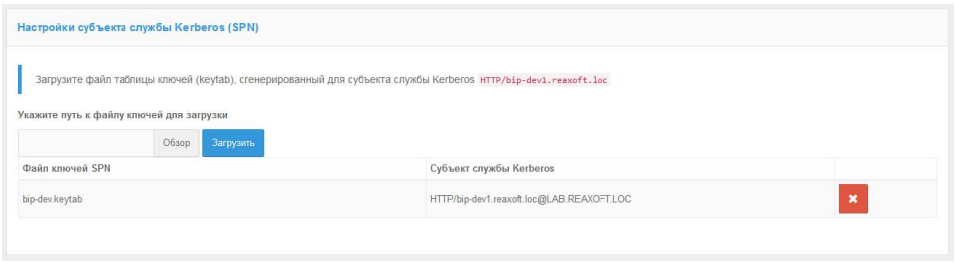
Настройки в консоли управления Blitz Identity Provider

Необходимо перейти в консоли управления в разделе Аутентификация к настройкам способа входа Вход по сеансу операционной системы. В открывшемся окне необходимо загрузить сгенерированный ранее keytab файл. Имя SPN при этом будет задано автоматически в соответствии с загруженным файлом.

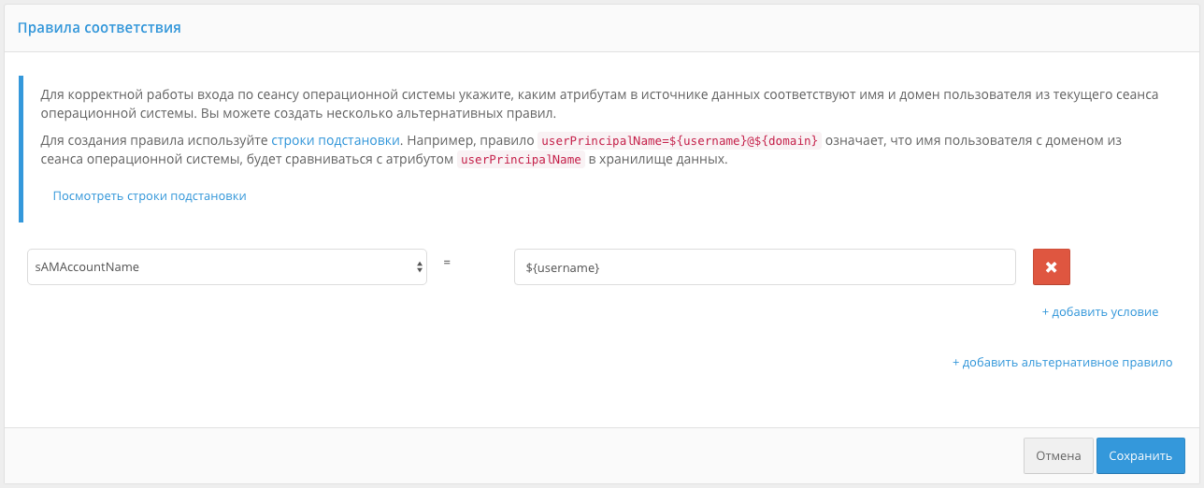
По результатам загрузки keytab-файла будет отображаться информация о соответствующей Kerberos-службе.

При необходимости можно:

- удалить загруженный keytab-файл;
- загрузить еще keytab-файлы, в случае подключения Blitz Identity Provider к нескольким контроллерам домена.

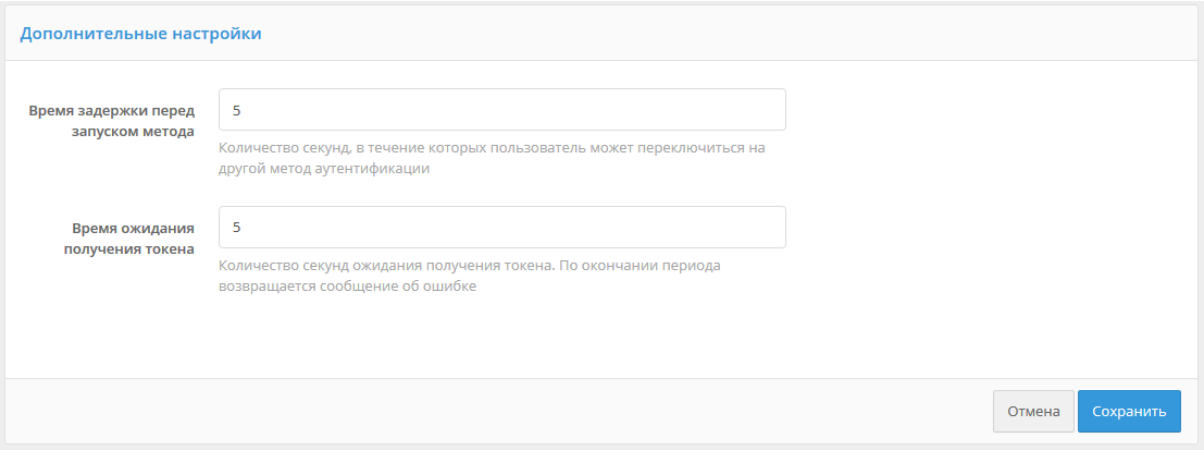


Далее необходимо определить параметры соответствия Kerberos-токена (TGS) и учетной записи в Blitz Identity Provider.



Например, можно задать соответствие, что получаемый из Kerberos-токена идентификатор пользователя (username) должен соответствовать атрибуту `sAMAccountName`, получаемому из LDAP-каталога (Microsoft Active Directory).

Далее необходимо установить параметры задержек при использовании метода входа с использованием сеанса операционной системы.



Blitz Identity Provider предоставляет два возможных сценария использования входа по сеансу операционной системы:

Основной сценарий. Пользователи входят в операционную систему, и после этого должны сквозным образом входить во все приложения, подключенные к Blitz Identity Provider. Предоставлять пользователям возможность войти в приложения под другой учетной записью не требуется. В этом случае нужно установить `Время задержки перед запуском метода`, равное `0` секунд. При обращении к приложению сразу будет произведена попытка сквозного входа по сеансу операционной системы.

Дополнительный сценарий. Пользователи не всегда имеют возможность войти в домен операционной системы, либо пользователям в некоторых случаях необходима возможность войти в приложения под другой учетной записью чем та, что они использовали для входа в домен. В этом случае нужно установить `Время задержки перед запуском метода` такое, чтобы пользователю хватило времени для возможности отменить автоматический вход с использованием сеанса операционной системы.

`Время ожидания получения токена` нужно установить достаточным, чтобы Kerberos сервер успевал предоставить ответ Blitz Identity Provider. Обычно достаточно установить 5 секунд.

Как и в случае входа по логину и паролю, по умолчанию поиск пользователей для аутентификации происходит во всех активных хранилищах. В блоке Правила выбора хранилища атрибутов можно настроить

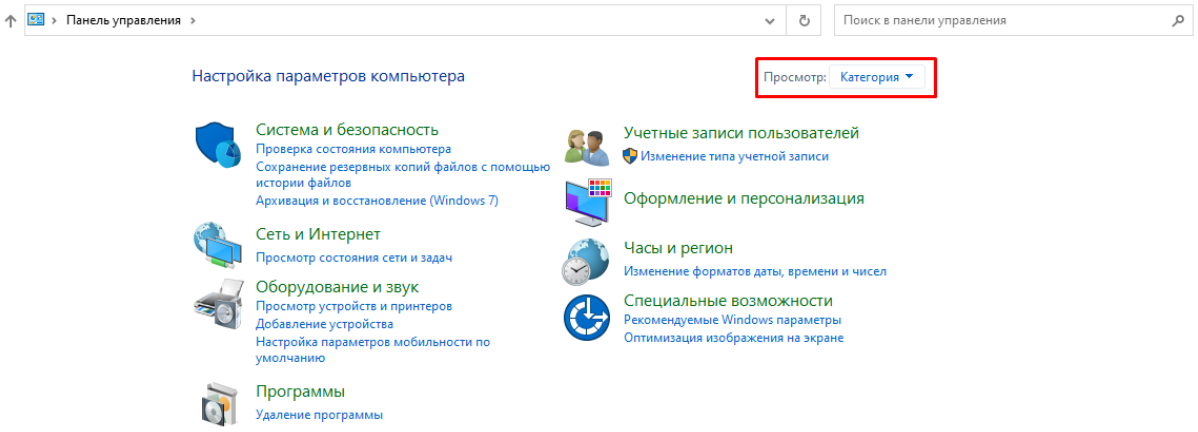
правила, при выполнении которых поиск пользователя будет осуществляться в *определенном хранилище* (страница 99).

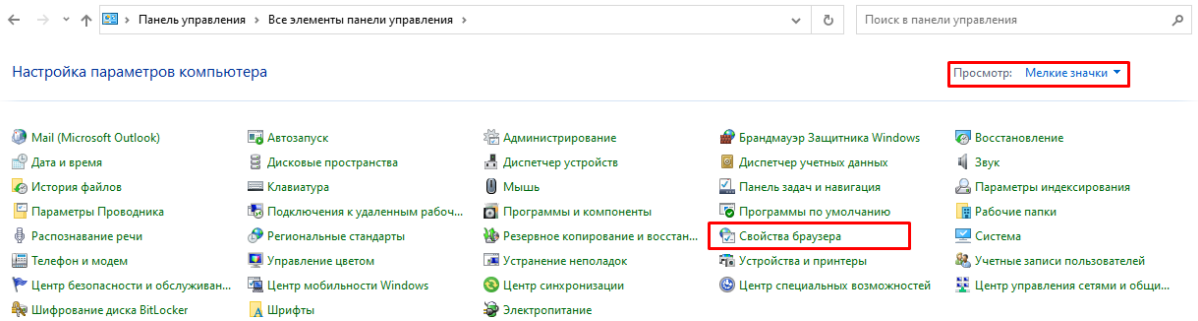
Настройки браузеров пользователей

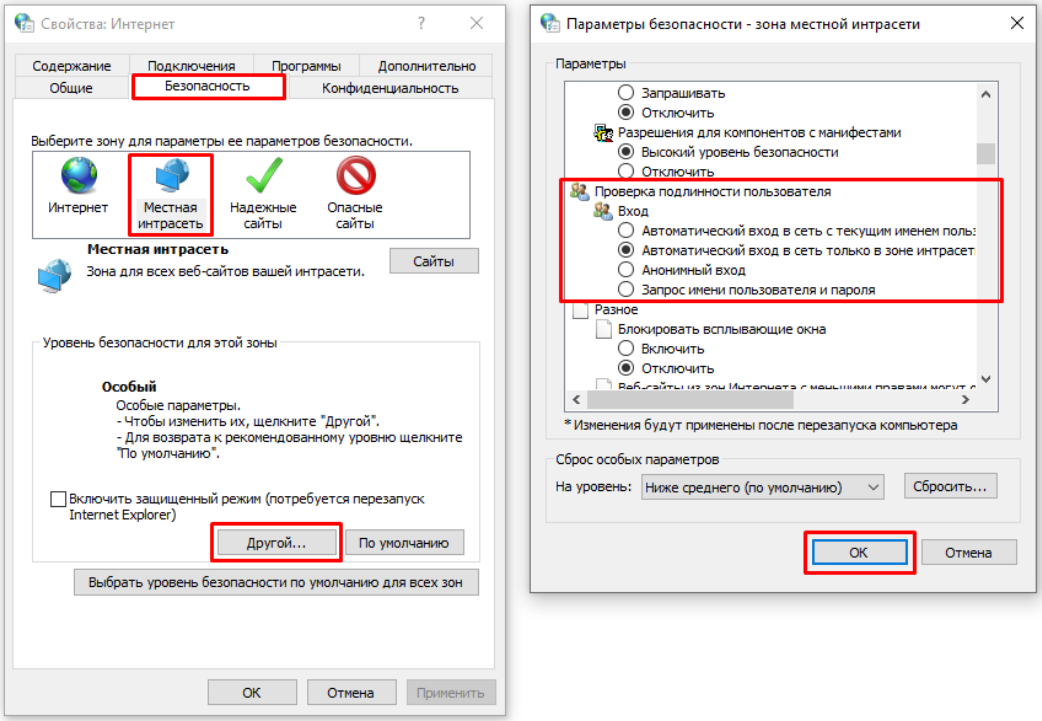
В зависимости от используемого пользователем браузера может потребоваться его дополнительная настройка для поддержки Kerberos-идентификации.

Для браузеров под операционной системой Windows нужно задать следующие настройки:

- открыть Пуск → Панель управления, изменить вариант просмотра с `Категория` на `Мелкие значки`, в открывшихся настройках выбрать Свойства браузера;
- в новом окне выбрать Безопасность → Местная интрасеть и нажать кнопку Сайты. В открывшемся окне нажать кнопку Дополнительно и внести сайт с Blitz Identity Provider в список сайтов Местная интрасеть, нажав Добавить;
- в окне Свойства: Интернет → Безопасность → Местная интрасеть нажать кнопку Другой…. В открывшемся окне найти настройку `Проверка подлинности пользователя` → `Вход`. Установить ее в значение `Автоматический вход в сеть только в зоне интрасети`.







Можно не задавать для операционной системы Windows описанные выше настройки и в качестве альтернативы для возможности входа по сеансу операционной системы в браузере Google Chrome тогда можно запускать браузер со следующими параметрами запуска:

```
Chrome.exe -auth-server-whitelist="idp.domain.ru" -auth-negotiate-
↪delegatewhitelist="idp.domain.ru"  -auth-schemes="digest,ntlm,negotiate"
```

Где в качестве `idp.domain.ru` нужно указать URL сайта Blitz Identity Provider.

Также можно задать следующие настройки в реестр Windows, чтобы запускать браузер Google Chrome без параметров запуска.

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Google]

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Google\Chrome]
"AuthNegotiateDelegateWhitelist"="idp.domain.ru"
"AuthSchemes"="basic,digest,ntlm,negotiate"
"AuthServerWhitelist"="idp.domain.ru"
```

Для Mozilla Firefox нужно задать следующие настройки (для любых операционных систем):

- в адресной строке браузера ввести `about:config` и нажать `Enter`. В следующем окне ввести `network.nego` в поле `Фильтры`. Дважды нажать на найденной записи `network.negotiate-auth.trusted-uris` и установить в ней значение URL сайта с Blitz Identity Provider, например, `idp.domain.ru`. При указании адресов можно использовать звездочку (`*`) и указать несколько URL через запятую, например: `https://*.idp.domain.ru,http://*.idp.domain.ru`. Закрыть всплывающее окно кнопкой OK.
- дважды нажать на найденной записи `network.negotiate-auth.delegation-uris` и установить в ней значение URL сайта с Blitz Identity Provider, например, `idp.domain.ru`. При указании адресов можно использовать звездочку (`*`) и указать несколько URL через запятую, например:

`https://*.idp.domain.ru,http://*.idp.domain.ru`. Закрыть всплывающее окно кнопкой ОК.

- открыть параметр `network.auth-sspi`, установить его значение в `true`;
- перезапустить браузер.

Для Google Chrome в macOS и в Linux нужно осуществлять запуск Google Chrome специальным образом:

```
"/Applications/Google Chrome.app/Contents/MacOS/Google Chrome" --args --auth-
↪server-whitelist="idp.domain.ru" --auth-negotiate-delegate-whitelist="idp.domain.
↪ru"
```

Где в качестве `idp.domain.ru` нужно указать URL сайта Blitz Identity Provider.

Для Apple Safari в macOS отдельная настройка не требуется.

Настройки запуска приложений Blitz Identity Provider

У пользователей могут возникнуть проблемы при входе по сеансу операционной системы, если они используют браузер Internet Explorer, и если в домене их учетная запись включена во многие группы безопасности, либо если DN учетной записи достаточно длинный. Чтобы избежать такой ситуации, необходимо при запуске приложения сервиса аутентификации `blitz-idp` задать специальный JAVA-параметр, определяющий большой допустимый размер HTTP-заголовка. Для этого необходимо отредактировать файл `/etc/default/blitz-idp`. В параметр JAVA_OPTS добавить ключ:

```
-Dakka.http.parsing.max-header-value-length=16K
```

Настройки веб-сервера

У пользователей могут возникнуть проблемы при входе по сеансу операционной системы, если они используют браузер Internet Explorer, и если в домене их учетная запись включена во многие группы безопасности, либо если DN учетной записи достаточно длинный. Чтобы избежать такой ситуации, необходимо скорректировать настройки веб-сервера, определяющие допустимый размер буферов заголовков.

Рекомендуемые значения буферов для nginx приведены ниже:

```
proxy_buffer_size 16k;
proxy_buffers 4 16k;
proxy_busy_buffers_size 16k;
client_body_buffer_size 16K;
client_header_buffer_size 16k;
client_max_body_size 8m;
large_client_header_buffers 2 16k;
```

Отладка проблем с входом по сеансу операционной системы

Если при выполненных настройках у пользователей все же не работает вход по сеансу операционной системы, то рекомендуется на компьютере пользователя в командной строке выполнить следующую команду:

```
klist
```

Если команда успешно вернет TGS мандаты для SPN, настроенного для Blitz Identity Provider, значит нужно проверять корректность настроек на стороне браузера пользователя и в Blitz Identity Provider. Если TGS мандаты для Blitz Identity Provider отсутствуют, то можно их запросить, используя следующую команду (необходимо указать правильные SPN и имя домена компании):

```
klist get HTTP/idp.company.ru@DOMAIN.LOC
```

Если команда не вернет полученных TGS мандатов, значит нужно проверять корректность настроек на Kerberos-сервере.

Вход с помощью электронной почты

Blitz Identity Provider позволяет выполнять вход с использованием электронной почты в качестве первого фактора аутентификации. В этом случае для первичного входа пользователю требуется ввести код, отправленный на адрес электронной почты. Для настройки метода выполните описанную ниже последовательность действий.

Шаг 1. Добавление метода в blitz.conf

Для того, чтобы в методах аутентификации на вкладке Первый фактор появился метод аутентификации Вход с помощью электронной почты, выполните следующие действия:

1. Откройте файл `/usr/share/identityblitz/blitz-config/blitz.conf`.

   ```
   sudo vim /usr/share/identityblitz/blitz-config/blitz.conf
   ```

2. В блоке настроек `blitz.prod.local.idp.login.factors` в первом списке добавьте блок настроек с методом `email`:

   ```
   "login" : {
       "factors" : [
           [
               …
           ],
           [
               {
                   "enabled" : false,
                   "method" : "email"
               },
               …
           ]
       ],
       …
   }
   ```

3. Перезапустите сервисы.

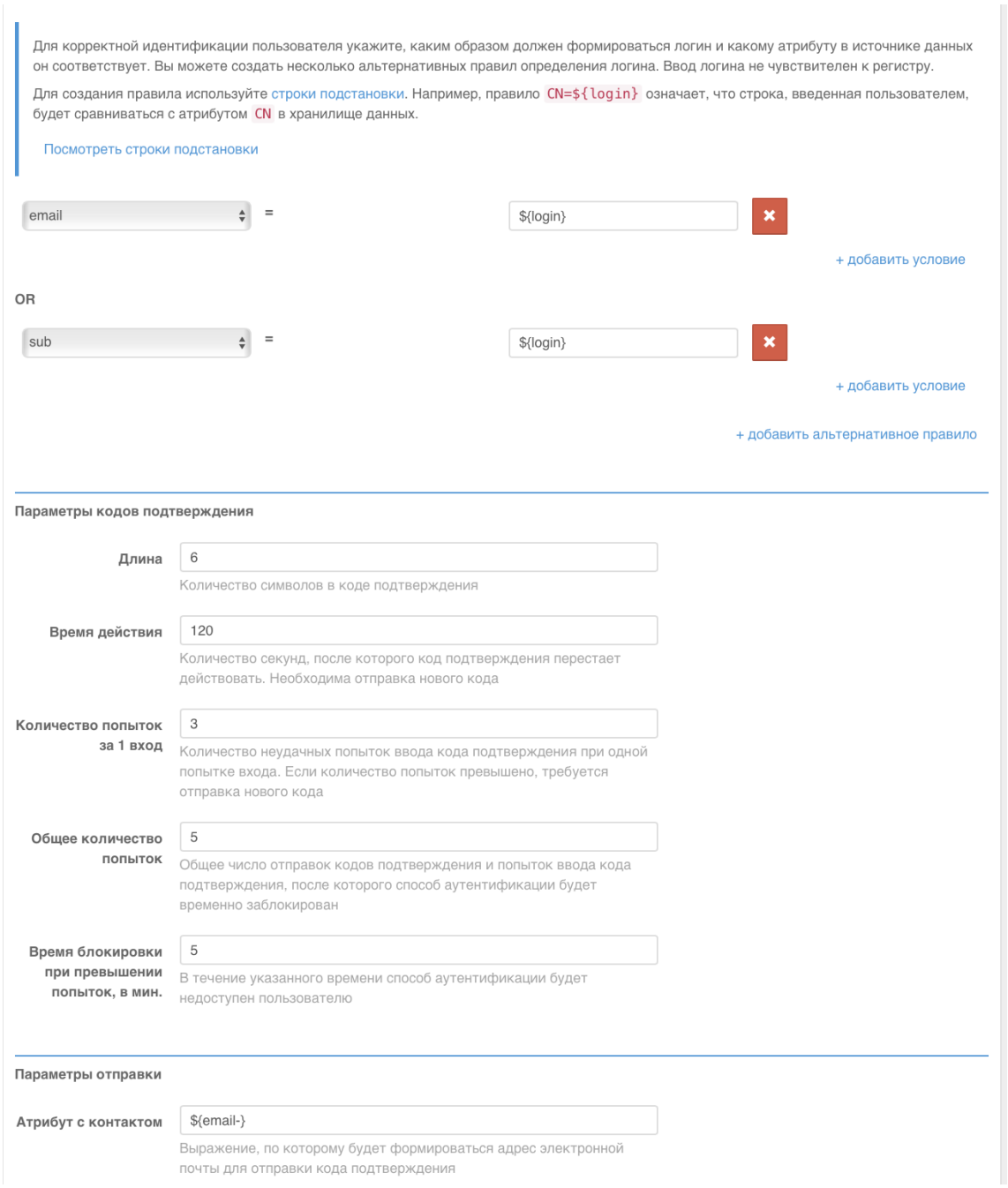
   ```
   sudo systemctl restart blitz-idp blitz-console blitz-recovery
   ```

Шаг 2. Настройка метода в консоли

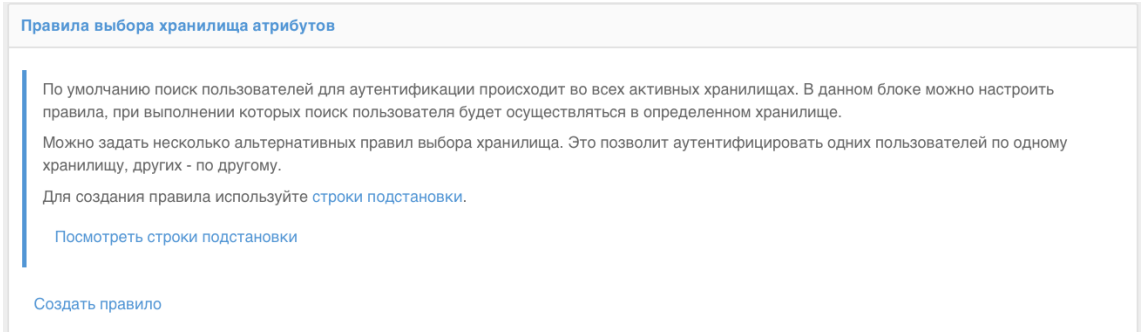
В консоли управления выполните следующие действия:

1. На вкладке Вход с помощью электронной почты задайте следующие настройки:
   - Способ идентификации учетной записи – задайте регулярное выражение. Например, правило `email=${login}` означает, что введенное пользователем значение в форме входа будет сопоставлено с атрибутом `email`.
   - Длину кода подтверждения.
   - Время его действия.

- Количество попыток ввода кода подтверждения за 1 вход.
- Общее количество попыток (число отправок кодов и попыток ввода кода, после чего для пользователя будет временно заблокирован данный способ аутентификации).
- Время блокировки при превышении попыток (в минутах).
- Способ отправки: в виде выражения укажите атрибут, в котором сохранен адрес электронной почты пользователя, например, `${email}`.



- Задайте правило выбора хранилища атрибутов для поиска введенного пользователем адреса электронной почты.



2. Включите метод Вход с помощью электронной почты в списке методов на вкладке Аутентификация -> Первый фактор.
3. Настройте подключение Blitz Identity Provider к *SMTP-сервису* (страница 217).

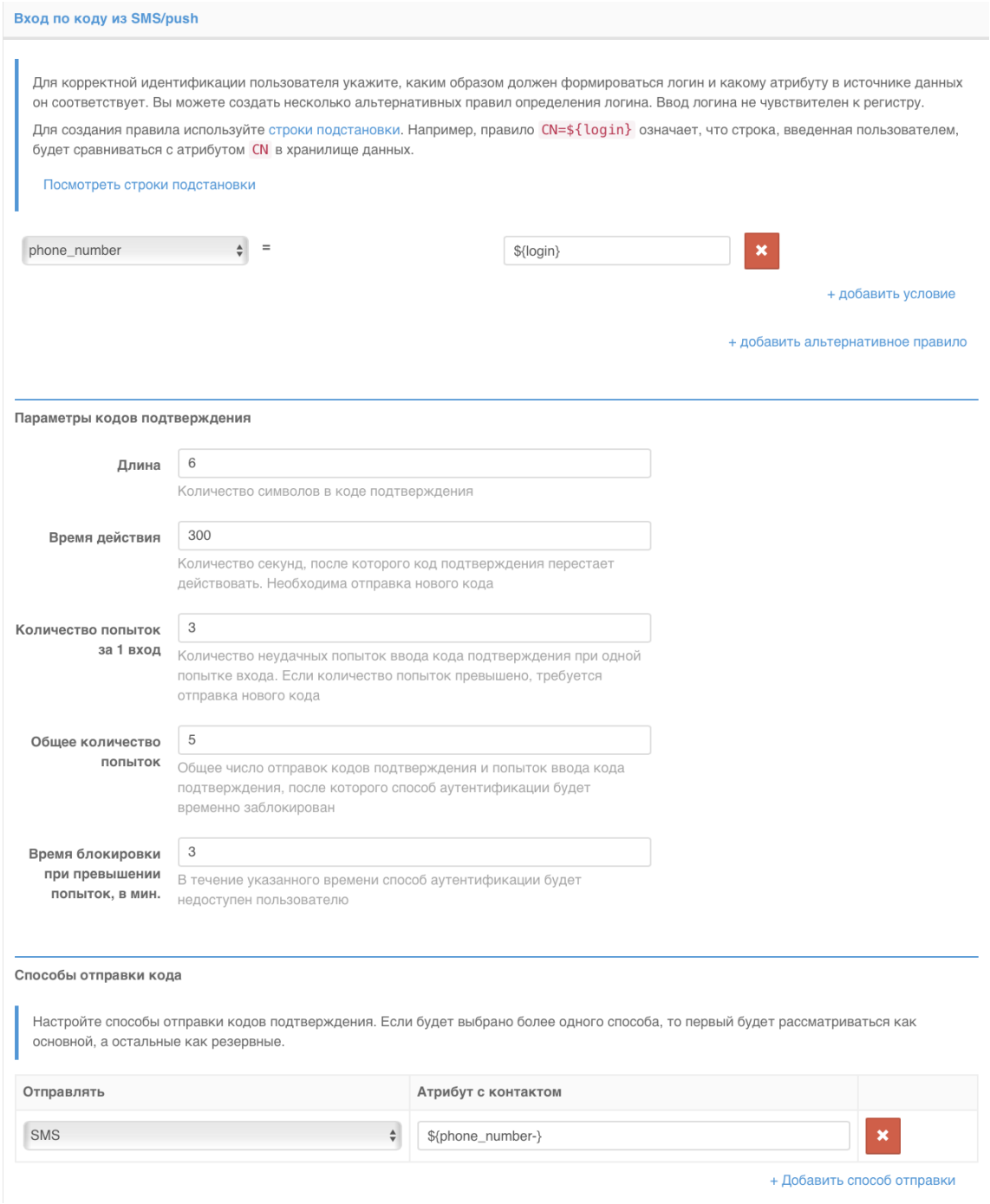
Вход с помощью кодов подтверждения

Для входа по первому фактору аутентификации можно использовать отправляемые в мобильное приложение push-уведомления или SMS-сообщения.

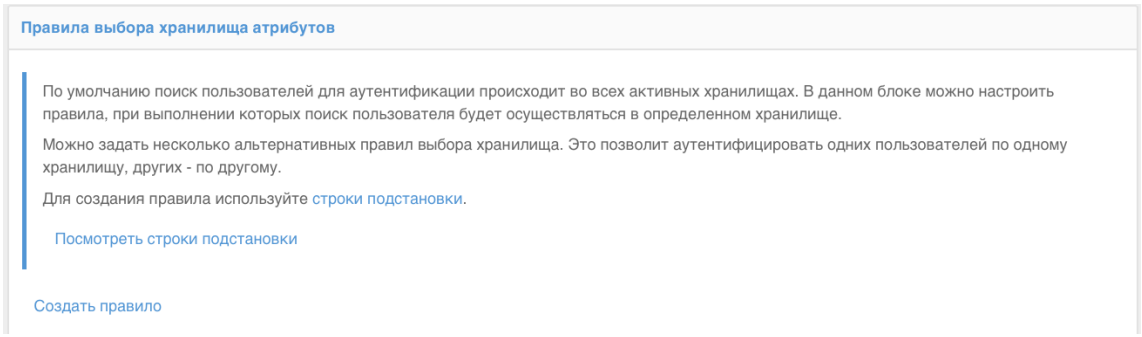
Внимание: Если у пользователя не задан номер мобильного телефона, то он не сможет использовать способ подтверждения входа с помощью кода подтверждения, отправляемого по SMS.

Для использования кодов подтверждения необходимо:

- настроить и включить включить метод аутентификации Вход по коду из SMS/push. Необходимо настроить:
  - способ идентификации учетной записи – задать регулярное выражение. Например, правило `phone_number=${login}` означает, что введенное пользователем значение в форме входа будет сопоставлено с атрибутом `phone_number`;
  - длину кода подтверждения;
  - время действия кода подтверждения;
  - количество попыток ввода кода подтверждения за 1 вход;
  - общее количество попыток (число отправок кодов и попыток ввода кода, после чего для пользователя будет временно заблокирован данный способ аутентификации);
  - время блокировки при превышении попыток (в минутах);
  - способы отправки кода:
    * отправлять push-уведомление – нужно указать атрибут с номером мобильного телефона или иным необходимым сервису идентификатором пользователя, например, `${phone_number}`;
    * отправлять SMS – указать атрибут с номером мобильного телефона пользователя, например, `${phone_number}`;



– правило выбора хранилища атрибутов для поиска введенного пользователем телефона.



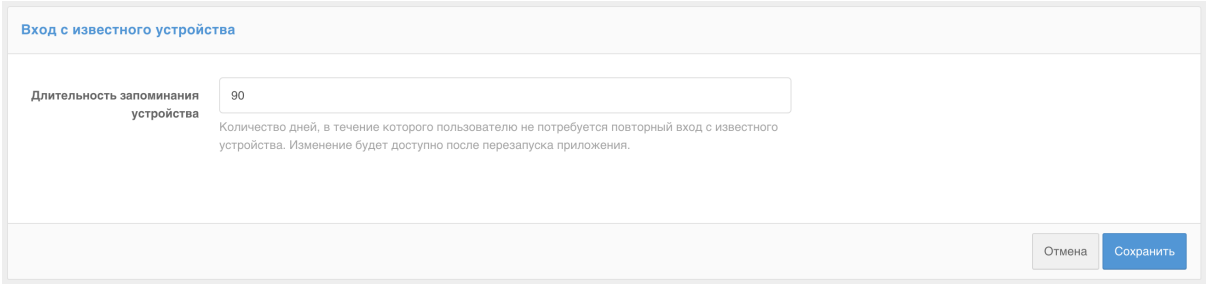
- настроить подключение Blitz Identity Provider к *SMS-шлюзу и сервису отправки push-уведомления*

(страница 217).

Вход с известного устройства

Вход с известного устройства позволяет не запрашивать идентификацию и аутентификацию пользователя (метод первого фактора), если пользователь, в течение определенного времени, уже осуществлял вход с данного устройства и браузера. Иными словами, пользователь может входит без аутентификации после перезапуска браузера.

Настройка метода включает в себя указание длительности запоминания устройства. Также можно установить, что при входе с запомненного устройства не будет требоваться двухфакторная аутентификация (опция *«Приравнять использование этого метода к применению первого и второго фактора»*). Если эта опция включена, то вход с известного устройства будет означать, что пользователь прошел двухфакторную аутентификацию.

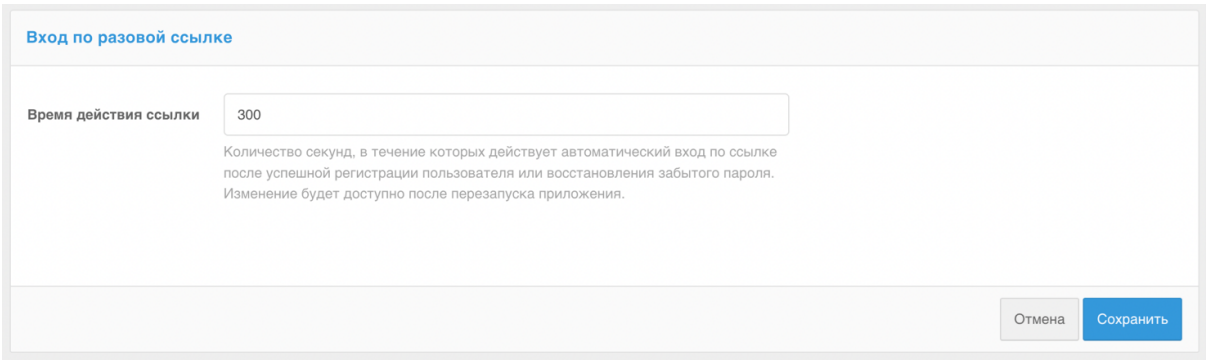


Вход по разовой ссылке

Вход по разовой ссылке используется для обеспечения автоматического входа после самостоятельной регистрации пользователем учетной записи, восстановлении забытого пароля или при использовании специального режима входа при открытии веб-браузера из мобильного приложения, в которое предварительно вошел пользователь.

> **Примечание:** Подробно этот сценарий описан *здесь* (страница 416).

Настройка метода включает в себя указание времени действия ссылки, используемой для автоматического входа. Чтобы сработал автоматический вход, с момента выработки ссылки (после успешного окончания регистрации или восстановления пароля или получения параметра `css` мобильным приложением) до момента инициирования входа пользователя прошло не больше указанного в настройке времени, и что ссылка ранее не была использована.



Вход по QR-коду

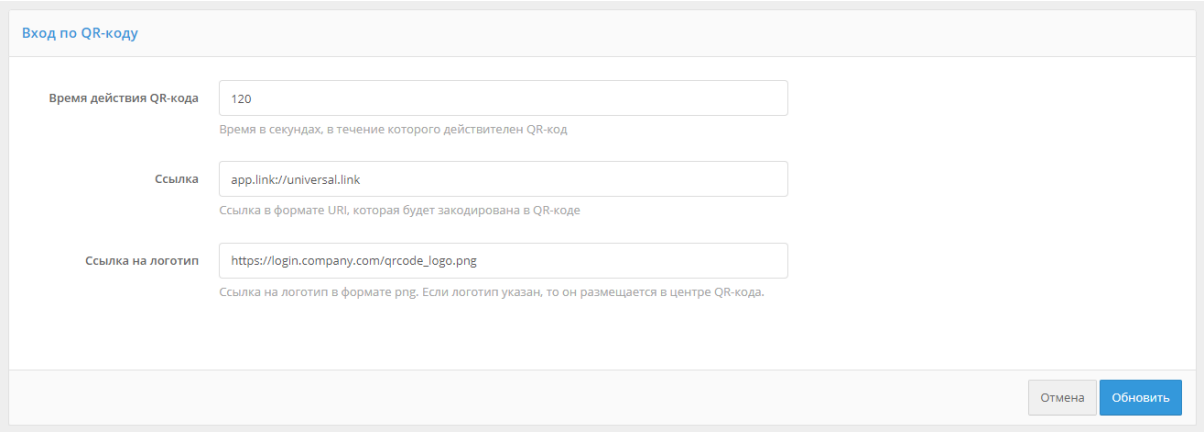
В Blitz Identity Provider предусмотрена возможность настроить вход в веб-приложение по QR-коду в качестве первого фактора аутентификации.

Процесс входа устроен следующим образом:

- Пользователь в браузере инициирует вход в веб-приложение. В Blitz Identity Provider отображается страница входа. На странице входа пользователь выбирает «Войти по QR коду».
- Blitz Identity Provider отображает на странице входа пользователю QR-код и инструкцию. QR-код имеет ограниченный срок действия (пользователю показывается таймер со сроком действия QR-кода).
- Пользователь запускает мобильное приложение компании, в которое встроена поддержка режима входа по QR-коду, и считывает с помощью этого приложения QR код.
- Мобильное приложение показывает пользователю детальную информацию о входе, полученную от Blitz Identity Provider (имя приложения, в которое осуществляется вход, IP-адрес, браузер и имя операционной системы устройства, на котором осуществляется вход).
- Пользователь в мобильном приложении принимает решение, разрешить или запретить вход.
- В зависимости от решения пользователя на компьютере происходит успешный вход пользователя в приложение или запрос входа отклоняется.

Настройка метода включает в себя указание следующих параметров:

- время действия QR-кода – в течение этого срока пользователь должен считать QR-код и принять решение по входу;
- ссылка, которая будет закодирована в QR-коде – указывает, какое приложение или веб страницу нужно запустить в случае считывания QR-кода стандартным приложением «Камера». В качестве параметра в ссылку будет передан закодированный QR-код (ссылка будет иметь вид `QR_URL?code=b0671081-cb73-4839-8bc1-8cf020457228`);
- ссылка на логотип (опционально) – данный логотип будет отображаться в центре QR кода.



Автоматическая идентификация пользователя по свойствам сессии

Blitz Identity Provider может выполнять автоматическую идентификацию пользователя и предоставление доступа по предварительно вычисленным свойствам сессии. Поддерживаются любые свойства сессии, которые могут быть определены средствами Заказчика и предоставлены в Blitz Identity Provider.

Совет: Частным случаем использования метода является вход пользователя по номеру мобильного телефона, автоматически определенного по его IP-адресу Заказчиком-оператором сотовой связи.

Внимание: Автоматическая идентификация возможна только для первого фактора.

Для использования данного метода аутентификации выполните описанную ниже последовательность действий.

Шаг 1. Создание процедуры входа

Для использования автоматической идентификации необходимо *создать процедуру входа* (страница 260), выполняемую до прохождения первого фактора аутентификации, которая будет запрашивать свойства сессии от сервиса Заказчика. Например, в частном случае при входе по автоматически определенному номеру телефона процедура должна выполнять следующие действия:

1. Определение IP-адреса пользователя. В случае если IP-адрес лежит в установленном диапазоне, производится вызов сервиса Заказчика-оператора сотовой связи для определения номера мобильного телефона.
2. После получения номера телефона процедура запрашивает у Blitz Identity Provider вход методом автоматической идентификации.

Шаг2. Добавление метода в blitz.conf

Для того чтобы метод автоматической идентификации отображался на вкладке Аутентификация -> Первый фактор, выполните следующие действия:

1. Откройте файл конфигурации `/usr/share/identityblitz/blitz-config/blitz.conf`.

```
sudo vim /usr/share/identityblitz/blitz-config/blitz.conf
```

2. Добавьте метод в список доступных методов первого фактора блока `blitz.prod.local.idp.login.factors` по аналогии с примером ниже. Методы первого фактора задаются в первой секции блока. Название метода должно состоять из префикса `sprop_` и идентификатора: например, у метода `sprop_msisdn` из примера идентификатор `msisdn`.

Примечание: Можно добавить несколько методов.

```
"login" : {
    "factors" : [
        [
            {
                "enabled" : false,
                "method" : "sprop_msisdn"
            },
            …
        ],
        [
            …
        ]
    ],
    …
}
```

3. Перезагрузите сервисы.

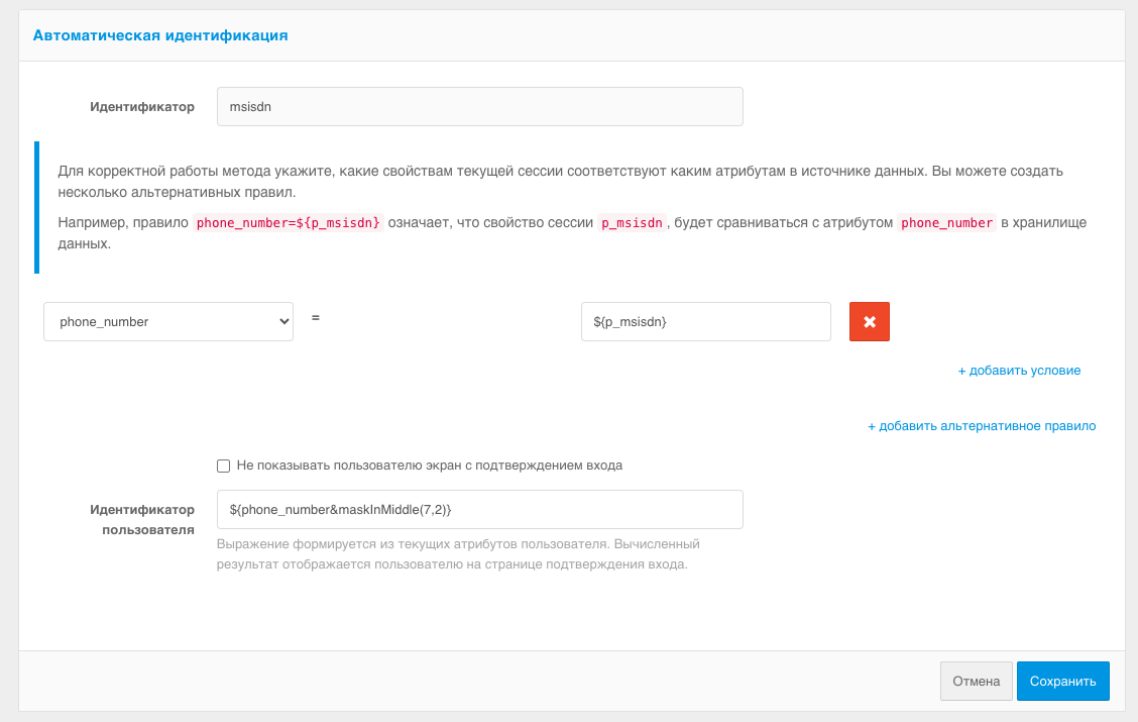
```
sudo systemctl restart blitz-idp blitz-console
```

Шаг 3. Настройка метода в консоли

Конфигурация метода в консоли управления выполняется следующим образом:

1. В консоли управления перейдите Аутентификация -> Первый фактор -> настройки метода Автоматическая идентификация.
2. Выполните маппинг атрибута, хранящегося в источнике данных Blitz Identity Provider, на свойство сессии, получаемое от сервиса Заказчика при выполнении процедуры входа. После получения свойства сессии Blitz Identity Provider выполнит поиск его значения среди значений указанного атрибута и в случае успеха разрешит вход по соответствующей учетной записи. Например, маппинг `phone_number=${p_msisdn}` означает, что свойство сессии `p_msisdn` будет сравниваться с атрибутом `phone_number` в хранилище данных.

Совет: Вы можете добавить несколько условий для поиска среди атрибутов, которые должны выполниться одновременно, чтобы пользователь был идентифицирован, а также ввести альтернативное правило.



3. По умолчанию после автоматической идентификации пользователя на его экране отображается его идентификатор и запрос на подтверждение входа. Задайте правило для формирования идентификатора пользователя из его атрибутов в виде строки подстановки. Это может быть замаскированный номер телефона, имя пользователя и др.

   Для того чтобы деактивировать подтверждение входа, поставьте флажок Не показывать пользователю экран с подтверждением входа.

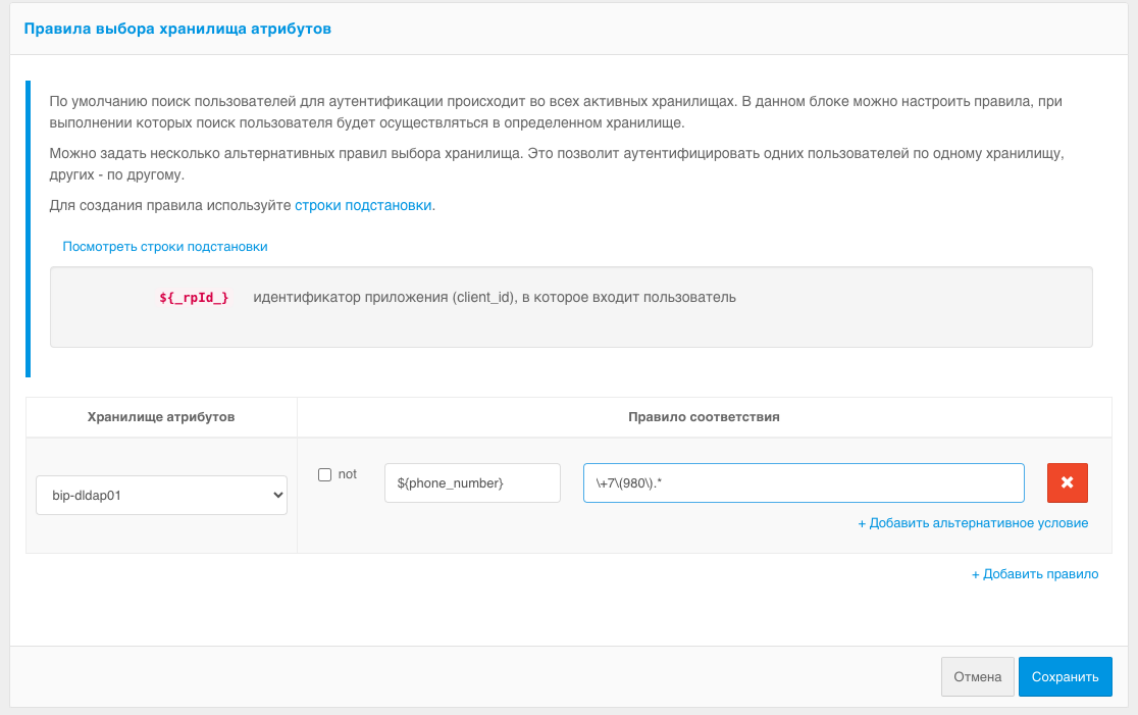
4. Нажмите Сохранить.
5. По умолчанию поиск пользователей для аутентификации происходит во всех активных хранилищах. В блоке Правила выбора хранилища атрибутов можно настроить правила, при выполнении которых поиск пользователя будет осуществляться в определенном хранилище. Можно задать несколько альтернативных правил выбора хранилища. Это позволит аутентифицировать одних пользователей по одному хранилищу, других – по другому.

   Для создания правила используйте следующие компоненты:

   - флажок `not`: признак инвертирования условия;

- первый столбец: проверяемое выражение, например, атрибут учетной записи, идентификатора приложения и пр.;
- второй столбец: условие выбора в виде регулярного выражения, например, значение атрибута пользователя, значение идентификатора приложения и пр.

Например, для того чтобы аутентифицировать всех пользователей, номер телефона которых содержит код `980`, в указанном хранилище, создайте правило, как показано на рисунке ниже.



6. Нажмите Сохранить.

Шаг 4. Кастомизация текстов

Если вы используете несколько методов *автоматической идентификации* (страница 119), следует провести кастомизацию текстов интерфейса для каждого из них, руководствуясь *алгоритмом* (страница 301).

В идентификатор текстовой строки понадобится включить имя метода или идентификатор метода. Имя метода *определено* (страница 120) в файле конфигурации `/usr/share/identityblitz/blitz-config/blitz.conf` и состоит из префикса `sprop_` и идентификатора метода: например, у метода `sprop_msisdn` идентификатор `msisdn`.

Для кастомизации используются следующие способы и строки:

Форма входа

Кастомизация с помощью имени метода `<sprop_id>`:

```
login.methods.sprop.head.title.<sprop_id>=Подтверждение входа по номеру телефона
login.methods.sprop.info.<sprop_id>=Ваш номер<br><strong>{0}</strong>.
login.methods.sprop.btn.consent.<sprop_id>=Войти
login.methods.sprop.btn.refuse.<sprop_id>=Войти под другим номером
```

Отображение метода в списке доступных методов при аутентификации

Кастомизация с помощью идентификатора метода <id>:

```
login.methods.switcher.title.sprop.<id>=Автовход по номеру телефона
login.methods.switcher.label.sprop.<id>=Автовход по номеру телефона
```

Отображение метода в списке методов в консоли управления

Кастомизация с помощью имени метода <sprop_id>:

```
page.authn.<sprop_id>.title=Автовход по номеру телефона
page.authn.<sprop_id>.info=Для идентификации пользователя используется свойство␣
↪сессии p_msisdn, которое вычисляется и сохраняется при старте процедуры входа.
```

Форма настройки метода в консоли

Кастомизация с помощью имени метода <sprop_id>:

```
page.method.sprop.title.<sprop_id>=Автовход по номеру телефона
page.method.sprop.info.<sprop_id>=<p>Для корректной работы автовхода укажите,␣
↪какие свойствам текущей сессии соответствуют каким атрибутам в источнике данных.␣
↪Вы можете создать несколько альтернативных правил. </p>Например, правило <code>
↪phone_number=$'{p_msisdn}'</code> означает, что свойство сессии <code>p_msisdn</
↪code>, будет сравниваться с атрибутом <code>phone_number</code> в хранилище␣
↪данных.</p>
```

Результат выполнения метода на вкладке События консоли управления

- Успешный вход: добавьте строку audit.method.<sprop_id>.
- Вход не выполнен: добавьте строку console.audit.type.auth_failed.<sprop_id>.

```
audit.method.<sprop_id>=Автовход по номеру телефона
console.audit.type.auth_failed.sprop_msisdn=Ошибка автовхода по номеру телефона
```

Отображение события неуспешного входа в Личном кабинете

Для отображения неуспешного входа в Личном кабинете пользователя добавьте строку profile.audit.type.auth_failed.<sprop_id>.

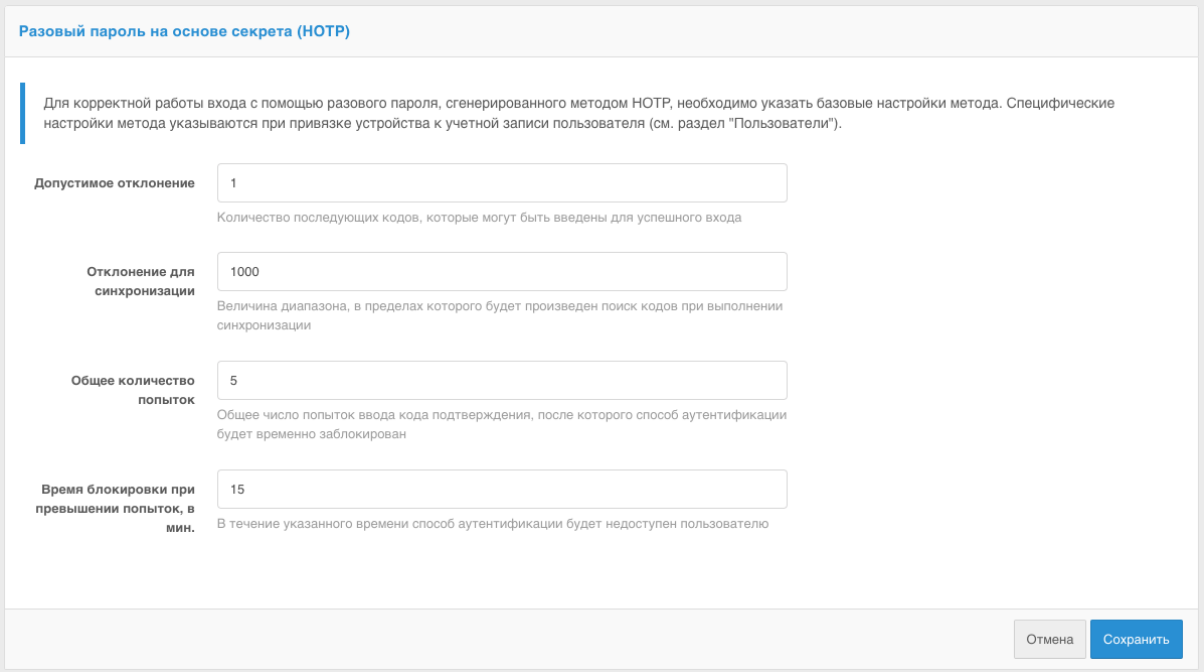
```
profile.audit.type.auth_failed.<sprop_id>=Ошибка автовхода по номеру телефона
```

Подтверждение входа разовым паролем на основе состояния (HOTP)

Для проверки второго фактора аутентификации с использованием метода аутентификации «Разовый пароль на основе секрета (HOTP)» можно использовать любой аппаратный брелок, совместимый со стандартом RFC4226 «HOTP: An HMAC-Based One-Time Password Algorithm»[29].

Для использования HOTP необходимо:

- настроить и включить данный метод аутентификации;
- загрузить в Blitz Identity Provider файл с описаниями HOTP-устройств. Файл с описаниями предоставляет поставщик HOTP-устройств. Для загрузки файла с описанием используется раздел меню *«Устройства»* в консоли управления Blitz Identity Provider;
- привязать HOTP-устройство к учетной записи пользователя и выдать HOTP-устройство пользователю. Привязку можно выполнить двумя способами – либо администратор привязывает устройство по серийному номеру к учетной записи пользователя в консоли управления в меню *«Пользователи»*, либо пользователь привязывает устройство к своей учетной записи самостоятельно с использованием веб-приложения *«Личный кабинет»*.



Для настройки метода аутентификации «Разовый пароль на основе секрета (HOTP)» необходимо задать:

- максимальное допустимое отклонение при проверке кода — количество последующих кодов (например, если пользователь случайно нажал кнопку генерирования нового пароля и не использовал его в процессе аутентификации), при котором аутентификация пройдет успешно. При этом при вводе пользователем правильного кода Blitz Identity Provider автоматически восстановит синхронизацию с устройством;
- отклонение для синхронизации – если пользователь многократно будет нажимать на устройстве кнопку выработки кода и не будет использовать код для подтверждения входа, то устройство перестанет быть синхронизированным с сервером. В этом случае при очередном входе пользователя в Blitz Identity Provider ему на странице входа будет предложено пройти процедуру сверки устройства. Для этого пользователь введет три последовательно выработанных устройством кода подтверждения. Далее в соответствии с заданной настройкой *«Отклонение для синхронизации»* Blitz Identity Provider проверит, встречается ли введенная пользователем последовательность кодов, и восстановит синхронизацию с устройством в случае успеха;

[29] https://tools.ietf.org/html/rfc4226

- общее количество попыток – число попыток ввода кода подтверждения, после которого данный способ подтверждения будет заблокирован;
- время блокировки при превышении попыток (в минутах).

Подтверждение входа разовым паролем на основе времени (TOTP)

Для проверки второго фактора аутентификации с использованием метода аутентификации «Разовый пароль на основе времени (TOTP)» можно использовать любые устройства и программы, совместимые со стандартом RFC6238 «TOTP: Time-Based One-Time Password Algorithm»[30]. В качестве таковых могут быть:

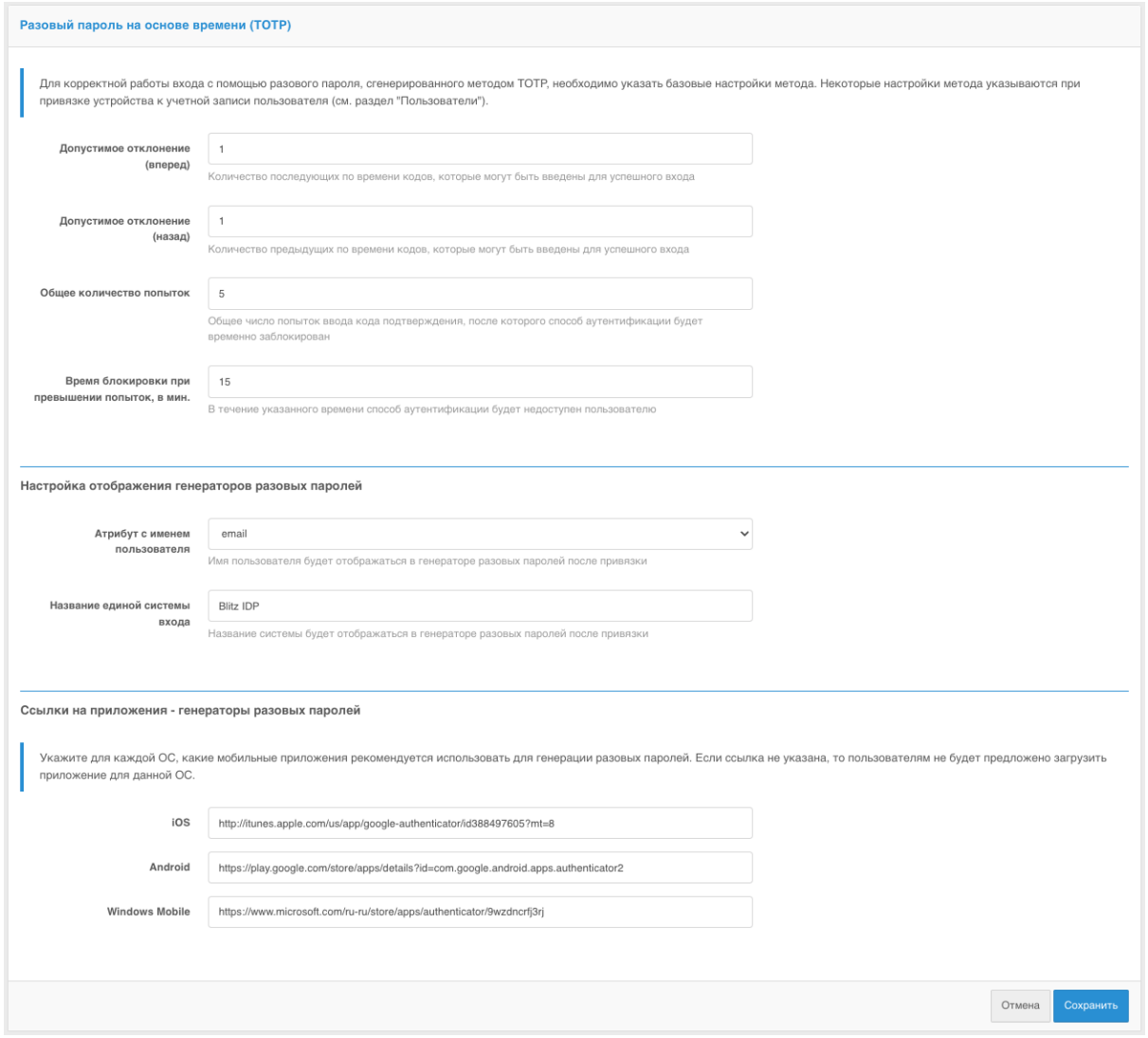
- аппаратные брелоки (генераторы разовых паролей) на основе времени;
- мобильные приложения.

> **Примечание:** Наиболее известные приложения для выработки TOTP-кодов: Google Authenticator, Twilio Authy, FreeOTP Authenticator, Microsoft Authenticator, Яндекс.Ключ.

В настройках метода аутентификации *«Разовый пароль на основе времени (TOTP)»* необходимо указать:

1. Допустимое отклонение при проверке кода (количество предыдущих / последующих кодов). По умолчанию оба значения равны `1`: пользователь при входе может ввести как текущий код подтверждения, так и следующий или предыдущий (сгенерированный в соседних временных интервалах). Такая необходимость может возникнуть, например, для компенсации возможной незначительной рассинхронизации серверного времени и времени на TOTP-устройствах пользователей.
2. Общее количество попыток – число попыток ввода кода подтверждения, после которого данный способ подтверждения будет заблокирован.
3. Время блокировки при превышении попыток (в минутах).
4. Настройка отображения генераторов разовых паролей, которая включает в себя *«Атрибут с именем пользователя»* и *«Название единой системы входа»*. Эти параметры будут отображаться в мобильном приложении после привязки учетной записи пользователя.
5. Ссылки на приложения-генераторы разовых паролей. Следует указать ссылки на приложения, которые рекомендуется использовать пользователям. Эти ссылки будут предложены пользователю в веб-приложении *«Личный кабинет»*.

[30] https://tools.ietf.org/html/rfc6238

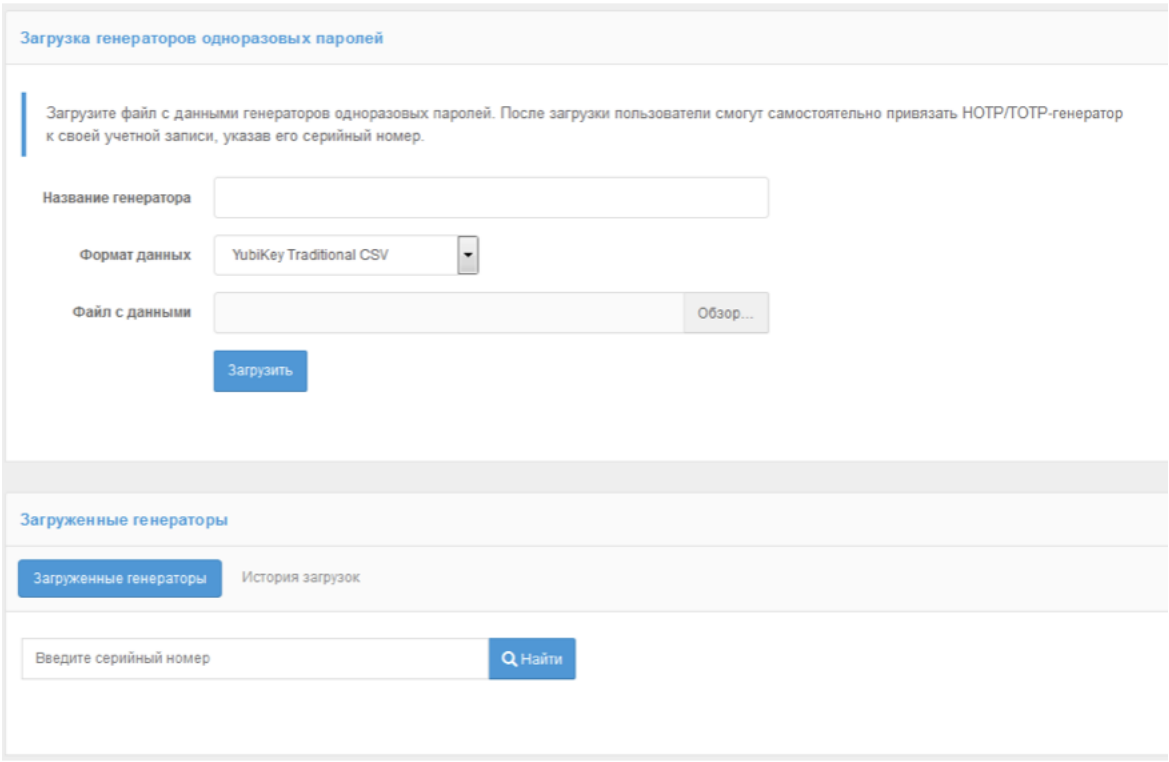


Привязка устройств к учетным записям пользователей

Привязка HOTP и TOTP устройств через консоль управления отличается в зависимости от того, используются аппаратные брелоки или мобильные приложения.

Привязка аппаратных брелоков

Для возможности использования аппаратных HOTP и TOTP устройств в качестве средств аутентификации администратор должен предварительно загрузить в консоли управления в меню *«Устройства»* файл с описаниями партии устройств, полученной от их поставщика. Файл содержит сведения о серийном номере устройства, векторе инициализации и ряд других настроек. Blitz Identity Provider поддерживает загрузку файлов распространенных форматов (специализированные XML-файлы, CSV-файлы) файлов с описаниями устройств от различных производителей устройств.



Для выполнения загрузки файла нужно задать имя для загружаемый генераторов (это может быть, например, имя устройства), формат данных, а также путь к файлу с описаниями устройств. По нажатии кнопки «Загрузить» Blitz Identity Provider сообщит, сколько записей устройств было загружено или отброшено (если их описание в файле было некорректно, либо запись об устройстве уже присутствуют в системе).

Пример загружаемого файла формата `Aladdin/SafeNet XML` для HOTP устройств с алгоритмом SHA-1 с минимальным набором параметров:

```
<?xml version="1.0" encoding="utf-8"?>
<Tokens>
  <Token serial="SN123">
    <Applications>
      <Application>
        <Seed>7bba106e428231c4d4e78361375d161c2d59b40b</Seed>
        <MovingFactor>0</MovingFactor>
      </Application>
    </Applications>
  </Token>
</Tokens>
```

Пояснения по значениям параметров в файле:

- `serial` – серийный номер устройства.
- `Seed` – ключ устройства в шестнадцатеричном (hex) формате.

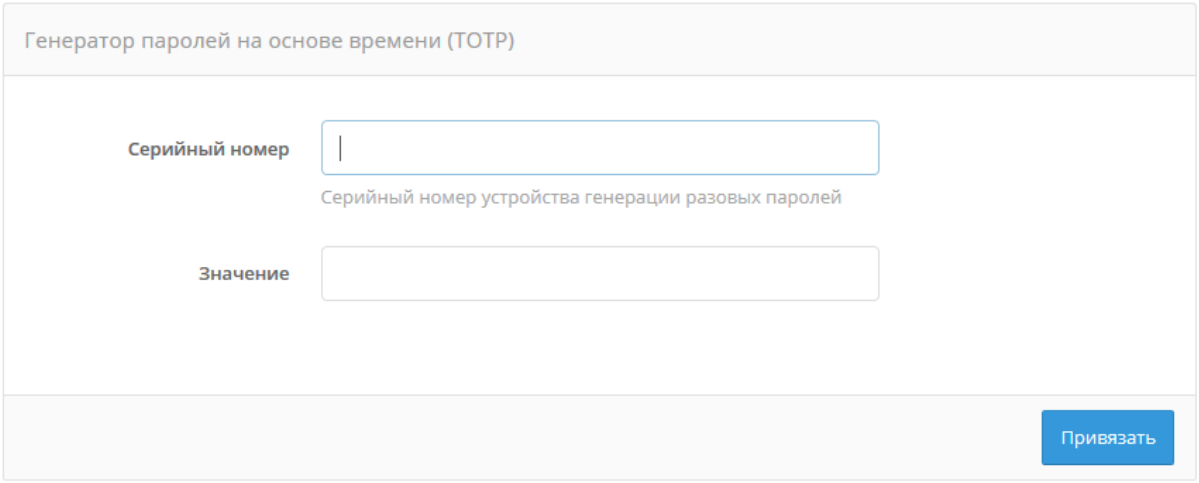
  Примечание: Если для эмуляции HOTP-устройства используется программный генератор одноразовых кодов, то обычно в программном генераторе в качестве секрета вводится строка в формате Base32. В этом случае значение из Seed нужно из hex перекодировать в Base32, и полученное значение использовать в программном генераторе.

- `MovingFactor` – начальное значение генератора (обычно 0).

В разделе «Устройства» также можно выполнить поиск устройства по серийному номеру, посмотреть, было ли привязано и к какой учетной записи найденное устройство.

После загрузки файла следует:

- перейти к учетной записи пользователя, которому необходимо привязать устройство (меню *«Пользователи»*, см. *Привязка устройств для 2FA по разовому паролю* (страница 207));
- найти раздел *«Генератор паролей на основе времени (TOTP)»* или *«Генератор паролей на основе секрета (HOTP)»*;
- выбрать *«Другой тип»*;
- ввести серийный номер необходимого устройства и текущий разовый код подтверждения.

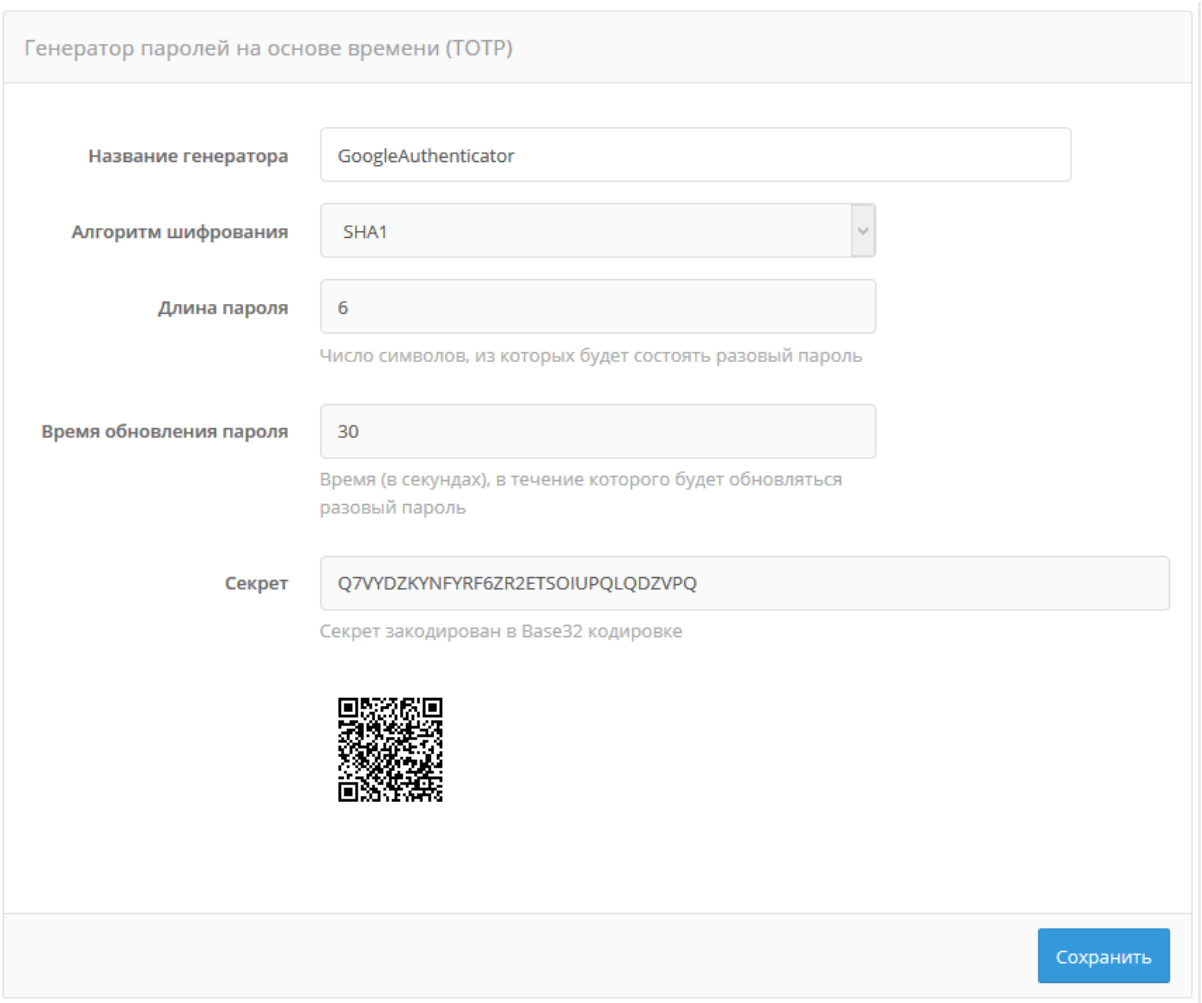


Привязка мобильного приложения

Для привязки мобильного приложения следует:

- перейти к учетной записи пользователя, которому необходимо привязать мобильное приложение (меню *«Пользователи»*, см. *Привязка устройств для 2FA по разовому паролю* (страница 207));
- найти раздел *«Генератор паролей на основе времени (TOTP)»*;
- выбрать *«GoogleAuthenticator»*;
- при необходимости отредактировать название мобильного приложения;
- с помощью мобильного приложения сфотографировать отображаемый QR-код или ввести в приложение строчку-секрет.

Также пользователь может самостоятельно привязать мобильное приложение, генерирующее TOTP-коды, в веб-приложении *«Личный кабинет»*.



Коды подтверждения, отправляемые в SMS и push-уведомлениях

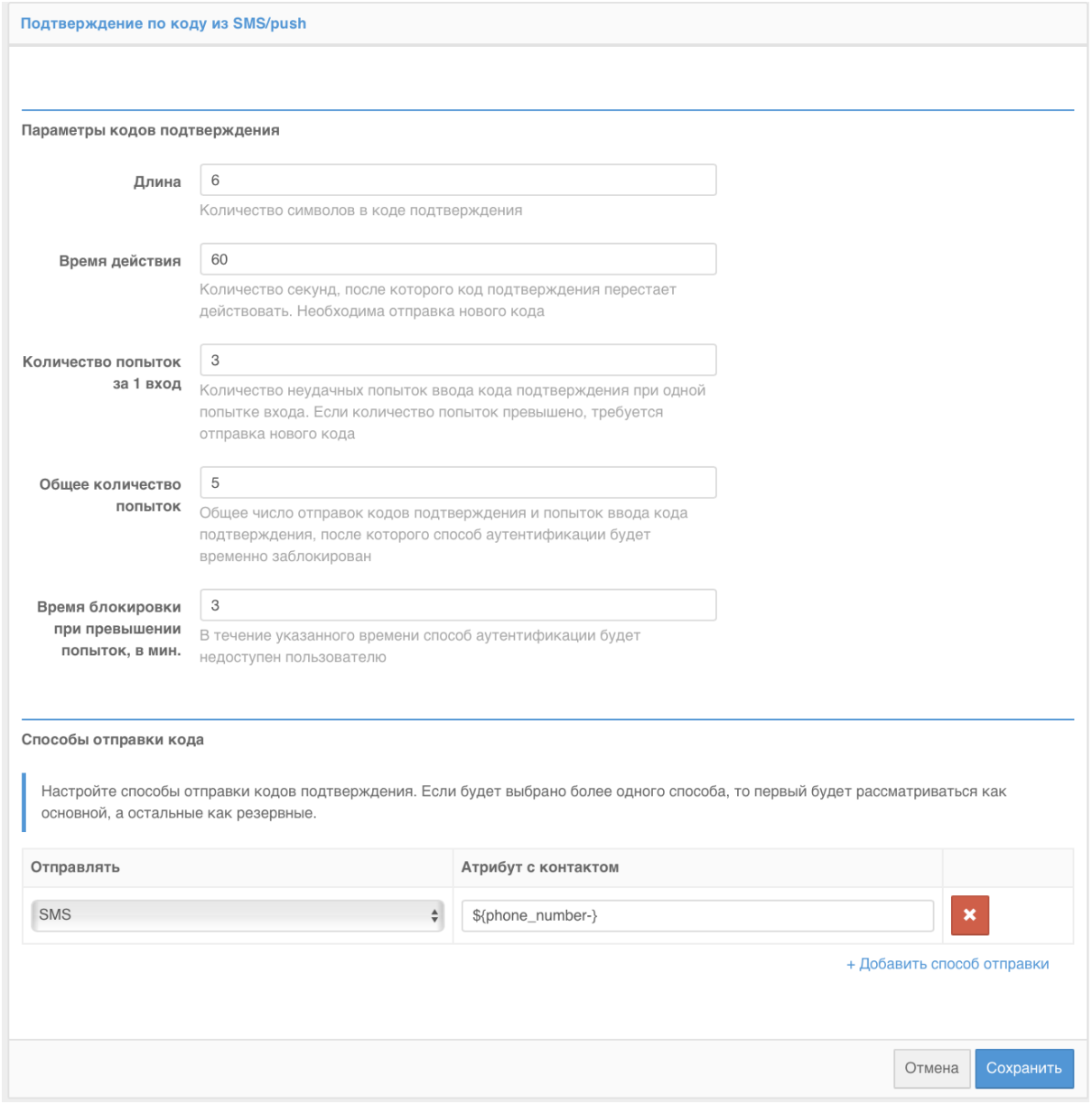
Для подтверждения входа (второго фактора аутентификации) можно использовать отправляемые в мобильное приложение push-уведомления или SMS-сообщения.

Для использования кодов подтверждения необходимо:

- настроить и включить метод аутентификации Подтверждение по коду из SMS/push. Для корректной работы метода необходимо определить:
  - длину кода подтверждения;
  - время его действия;
  - количество попыток ввода кода подтверждения за 1 вход;
  - общее количество попыток (число отправок кодов и попыток ввода кода, после чего для пользователя будет временно заблокирован данный способ аутентификации);
  - время блокировки при превышении попыток (в минутах);
  - сконфигурировать способы отправки:
    - отправлять push-уведомление – нужно указать атрибут с номером мобильного телефона или иным необходимым сервису идентификатором пользователя, например, `${phone_number}`;
    - отправлять SMS – указать атрибут с номером мобильного телефона пользователя, например, `${phone_number}`;

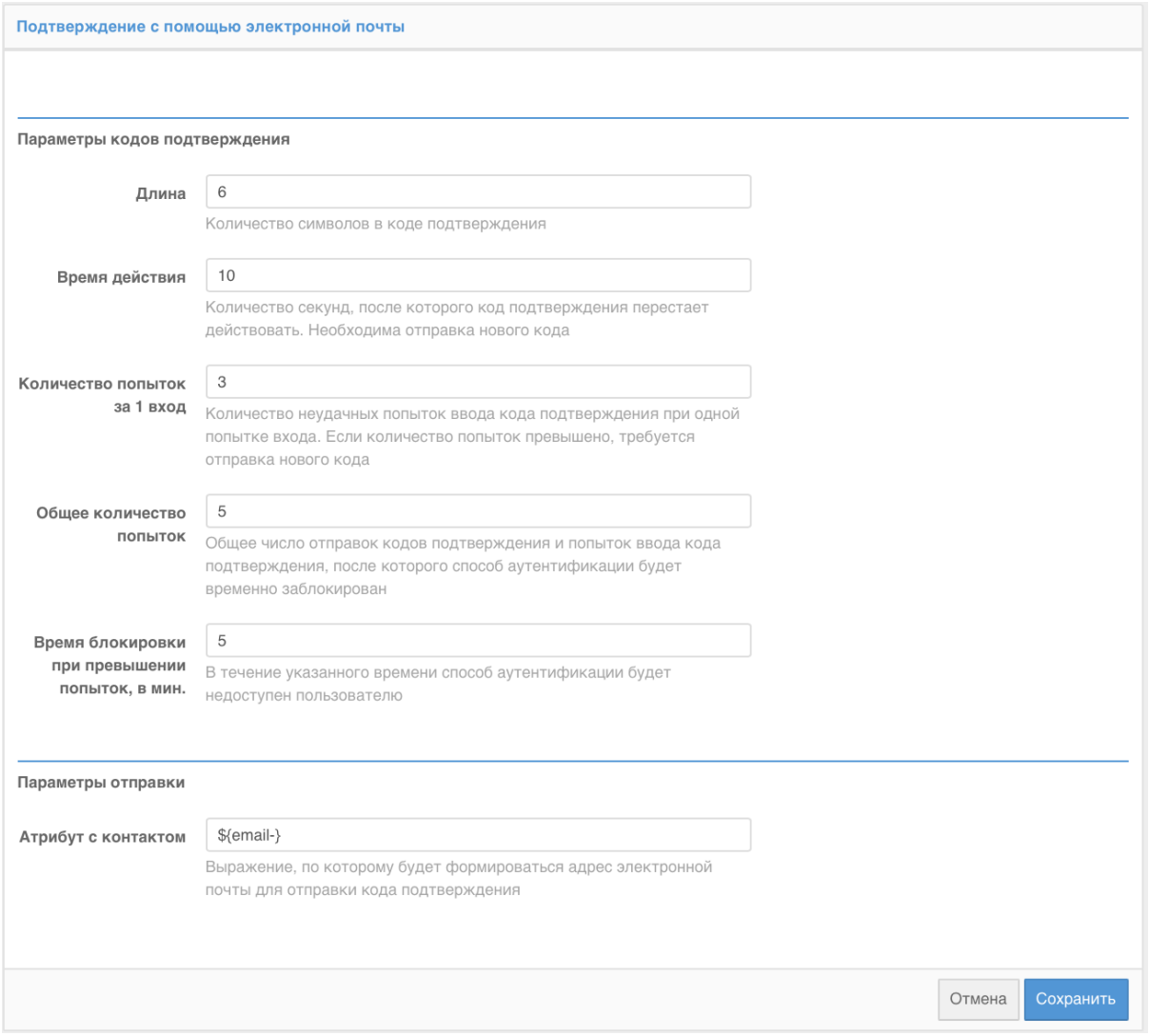
- настроить подключение Blitz Identity Provider к SMS-шлюзу и сервису отправки push-уведомления (см. *Уведомления и отправка сообщений* (страница 217)).

Внимание: Если у пользователя не задан номер мобильного телефона, то он не сможет использовать способ подтверждения входа с помощью кода подтверждения, отправляемого по SMS.



Коды подтверждения, отправляемые по электронной почте

Для подтверждения входа можно использовать отправляемые по электронной почте коды подтверждения.



Для этого необходимо:

- настроить и включить этот метод аутентификации. Для корректной работы метода необходимо определить:
  - длину кода подтверждения;
  - время его действия;
  - количество попыток ввода кода подтверждения за один вход;
  - общее количество попыток (число отправок кодов и попыток ввода кода, после чего для пользователя будет временно заблокирован данный способ аутентификации);
  - время блокировки при превышении попыток (в минутах);
  - сконфигурировать способ отправки: указать атрибут, в котором сохранен адрес электронной почты пользователя, например, `${email}`;
- *настроить* (страница 217) подключение Blitz Identity Provider к SMTP-сервису.

Подтверждение входа с помощью Duo Mobile

Можно использовать мобильное приложение Duo Mobile[31] (компания Cisco) для подтверждения входа (второго фактора аутентификации).

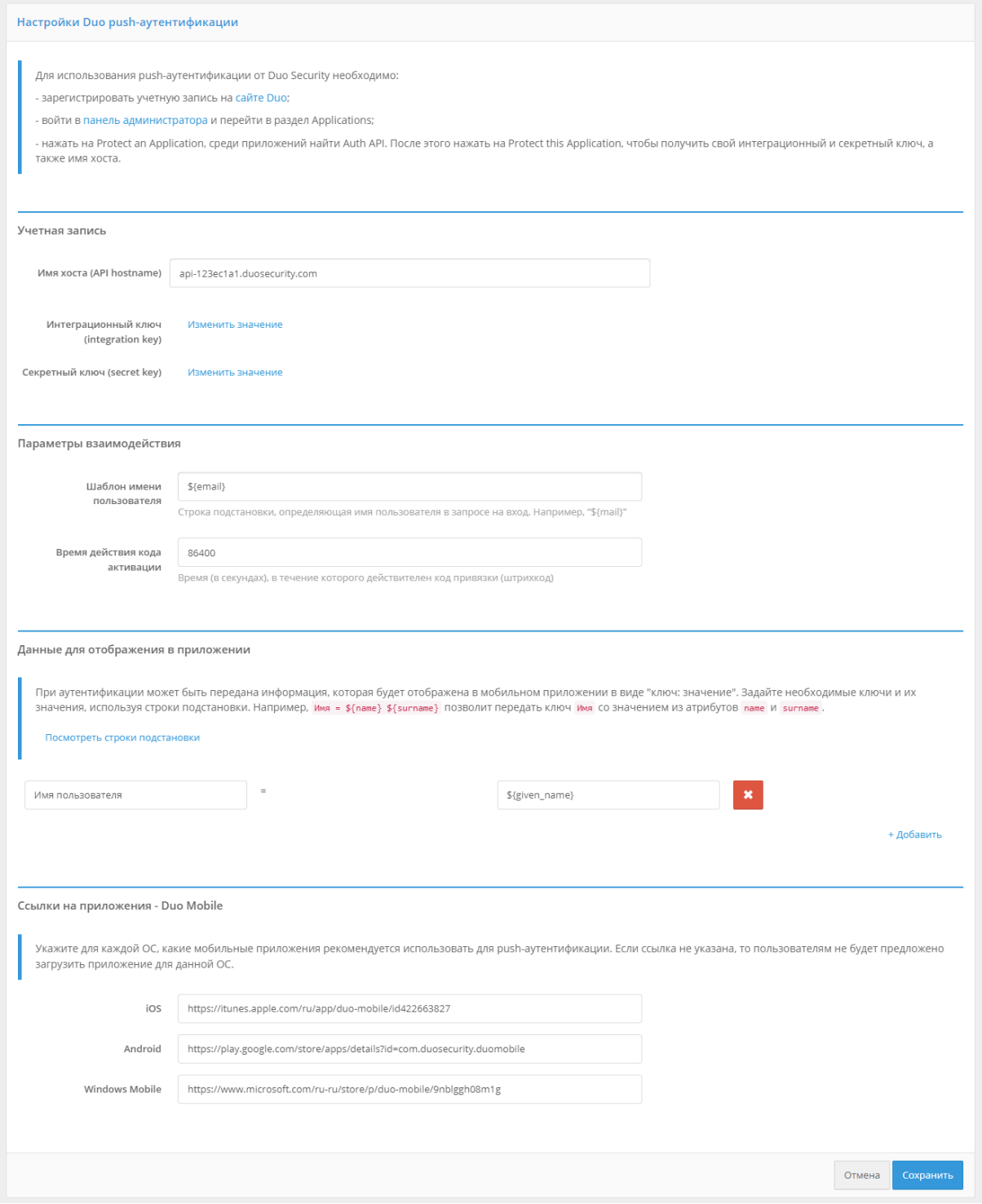
Для этого необходимо выполнить настойки на стороне сервиса Duo Security:

- зарегистрировать учетную запись на сайте Duo[32];
- войти в панель администратора[33] и перейти в раздел Applications;
- нажать на Protect an Application, среди приложений найти `Auth API`. После этого нажать на Protect this Application, чтобы получить свой интеграционный и секретный ключ, а также имя хоста.

После выполнения этих операций нужно провести настройки в консоли управления Blitz Identity Provider.

- сконфигурировать метод аутентификации Duo push-аутентификация. Необходимо указать:
  - параметры учетной записи Duo (имя хоста, интеграционный и секретный ключ);
  - параметры взаимодействия:
    - имя пользователя (задается с помощью строки подстановки) – это имя будет отображено в Duo Mobile в качестве имени учетной записи;
    - время действия кода активации (в секундах) – время, в течение которого действителен код привязки (QR-код);
  - данные для отображения в приложении – информация, отображаемая пользователю в Duo Mobile в виде «ключ: значение». Здесь можно передать значение пользовательского атрибута или какое-то фиксированное значение. В качестве значения также можно указать строку `${app}` – это позволит отобразить имя приложения, куда пользователь входит;
  - ссылки на загрузку приложения Duo Mobile.
- включить метод Duo push-аутентификации в разделе Аутентификация.

[31] https://duo.com/product/multi-factor-authentication-mfa/duo-mobile-app
[32] https://signup.duo.com/
[33] https://admin.duosecurity.com/



Привязка приложения Duo Mobile к учетной записи пользователя возможна следующими способами:

- пользователем самостоятельно через веб-приложение Личный кабинет;
- администратором через консоль управления.

В веб-приложении Личный кабинет пользователь должен перейти в раздел Безопасность / Подтверждение входа и выполнить следующие шаги:

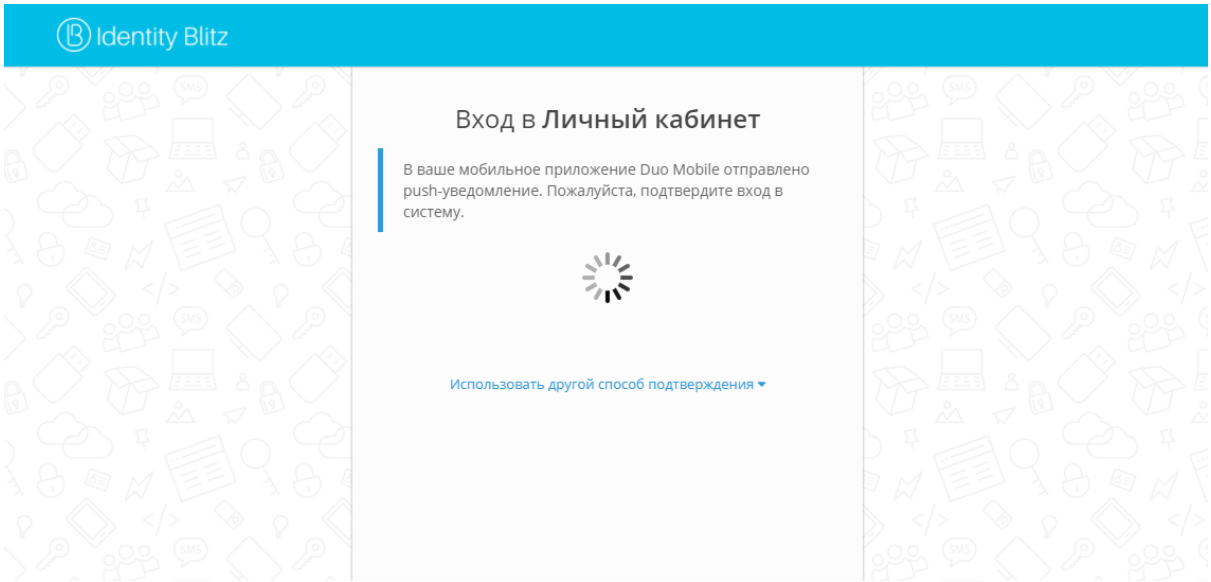
1. Выбрать способ подтверждения входа – Подтверждение с помощью мобильного приложения Duo Mobile.

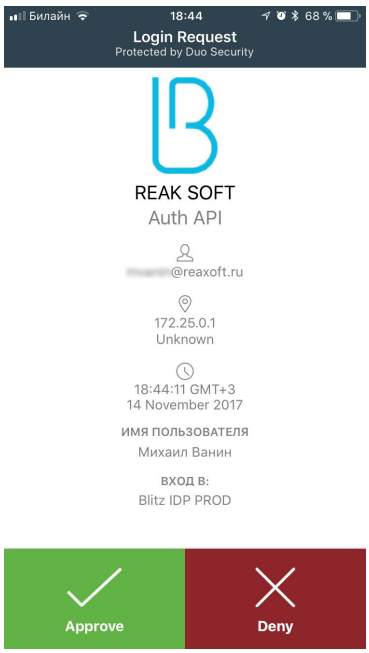
2. Установить на смартфон приложение Duo Mobile и отсканировать QR-код, а также нажать Подтвердить.
3. После проверки этот метод аутентификации будет добавлен пользователю.

В консоли управления администратор должен:

1. Найти необходимого пользователя.
2. Перейти к блоку Приложение Duo Mobile (QR-код) и нажать на кнопку Привязать Duo Mobile.
3. Попросить пользователя отсканировать QR-код с помощью мобильного приложения Duo Mobile.

На рисунках приведен пример внешнего вида страницы входа при подтверждении входа с помощью push-уведомления в приложении Duo Mobile.

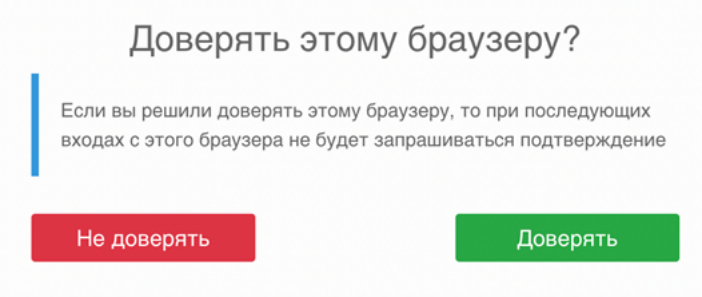




Повторное подтверждение при входе с известного устройства

Blitz Identity Provider запоминает устройства, на которых пользователь в процессе входа подтверждал вход с помощью одного из поддерживаемых Blitz Identity Provider методов подтверждения входа.

Можно настроить в процедуре входа, чтобы после успешного подтверждения входа пользователю показывался экран с вопросом, доверяет ли пользователь браузеру, чтобы при повторных входах с этого устройства и браузера у него не запрашивалось подтверждение входа.



В случае повторного входа с доверенного браузера у пользователя не будет запрашиваться подтверждение входа, если в меню Аутентификация в блоке Второй фактор включен метод аутентификации Вход с известного устройства.

Подтверждение ответом на контрольный вопрос

Blitz Identity Provider позволяет для подтверждения входа запросить пользователя ввести ответ на контрольный вопрос. Это может быть полезно в сценариях подтверждения при восстановлении забытого пароля. Для использования данного метода аутентификации выполните описанную ниже последовательность действий.

Шаг 1. Добавление метода в blitz.conf

Для того, чтобы в методах аутентификации на вкладке Второй фактор появился метод аутентификации Подтверждение ответом на контрольный вопрос, выполните следующие действия:

1. Откройте файл `/usr/share/identityblitz/blitz-config/blitz.conf`.

```
sudo vim /usr/share/identityblitz/blitz-config/blitz.conf
```

2. В блоке настроек `blitz.prod.local.idp.login.factors` во втором списке добавьте блок настроек с методом `secQsn`:

```
"login" : {
    "factors" : [
        [
            …
        ],
        [
            {
                "enabled" : false,
                "method" : "secQsn"
            },
            …
        ]
    ],
    …
}
```

3. Перезапустите сервисы.

```
sudo systemctl restart blitz-idp blitz-console blitz-recovery
```

Шаг 2. Создание справочника контрольных вопросов

Для того чтобы создать справочник контрольных вопросов, выполните следующие действия:

1. Создайте на сервере директорию `/etc/blitz-config/custom_messages/dics`.
2. Создайте файл `/etc/blitz-config/custom_messages/dics/securityQuestions` с содержимым справочника. Пример файла `securityQuestions` со справочником контрольных вопросов:

```
01=Какая девичья фамилия у вашей матери
02=Какая девичья фамилия у вашей бабушки
03=Какой фильм вы впервые посмотрели в кинотеатре
04=Какое ваше любимое литературное произведение
05=Как звали вашего учителя в третьем классе
06=Первое блюдо, которое вы научились готовить
07=Как звали вашего первого питомца
08=Кем вы хотели стать в детстве
09=Как называлась первая школа, в которую вы ходили
10=Как называлась первая улица, где вы жили в детстве
```

> **Внимание:** Число в справочнике используется для сортировки при отображении списка контрольных вопросов пользователю.

3. Проверьте владельца директории `dics` и файлов справочников в ней. Владелец должен быть `blitz:blitz`.

```
chown -R blitz:blitz /etc/blitz-config/custom_messages/dics
```

4. В конфигурационном файле `/usr/share/identityblitz/blitz-config/blitz.conf` в блок `blitz.prod.local.idp.messages` добавьте блок `dics`. В настройке `names` укажите имя справочника `securityQuestions`. Например:

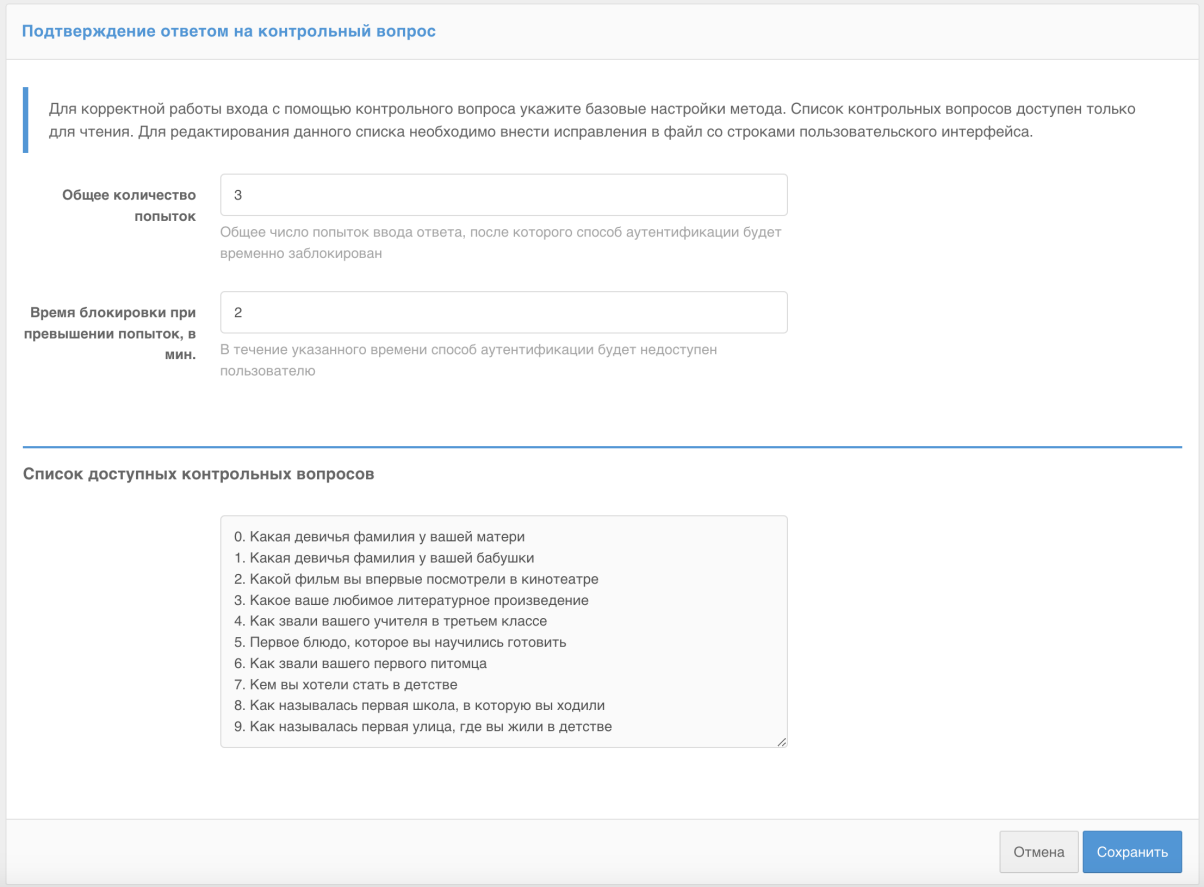
```
"dics" : {
    "dir" : "custom_messages/dics",
    "names" : [
        "securityQuestions"
    ]
}
```

Шаг 3. Настройка метода в консоли

В консоли управления необходимо задать следующие настройки:

- `Общее количество попыток` – число попыток ввода ответа на контрольный вопрос, после которых данный способ подтверждения будет заблокирован.
- `Время блокировки при превышении попыток (в минутах).`

Также в консоли управления отображается список *настроенных* (страница 136) контрольных вопросов.



Подтверждение по входящему звонку

Blitz Identity Provider позволяет передавать одноразовые коды для реализации второго фактора аутентификации в номере входящего звонка (метод Flash Call). В этом случае после успешной первичной аутентификации на номер пользователя будет выполнен звонок с заранее неизвестного телефонного номера, последние цифры которого потребуется ввести для подтверждения входа. Звонок выполняется с разрешения пользователя.

Для настройки метода Flash Call выполните описанную ниже последовательность действий.

Шаг 1. Добавление метода в blitz.conf

Для того, чтобы в методах аутентификации на вкладке Второй фактор появился метод аутентификации Подтверждение по входящему звонку, выполните следующие действия:

1. Откройте файл `/usr/share/identityblitz/blitz-config/blitz.conf`.

```
sudo vim /usr/share/identityblitz/blitz-config/blitz.conf
```

2. В блоке настроек `blitz.prod.local.idp.login.factors` во втором списке добавьте блок настроек с методом `flashCall`:

```
"login" : {
    "factors" : [
        [
            ...
        ],
        [
```

(continues on next page)

(продолжение с предыдущей страницы)

```
            {
                "enabled" : false,
                "method" : "flashCall"
            },
            …
        ]
    ],
    …
}
```

3. Перезапустите сервисы.

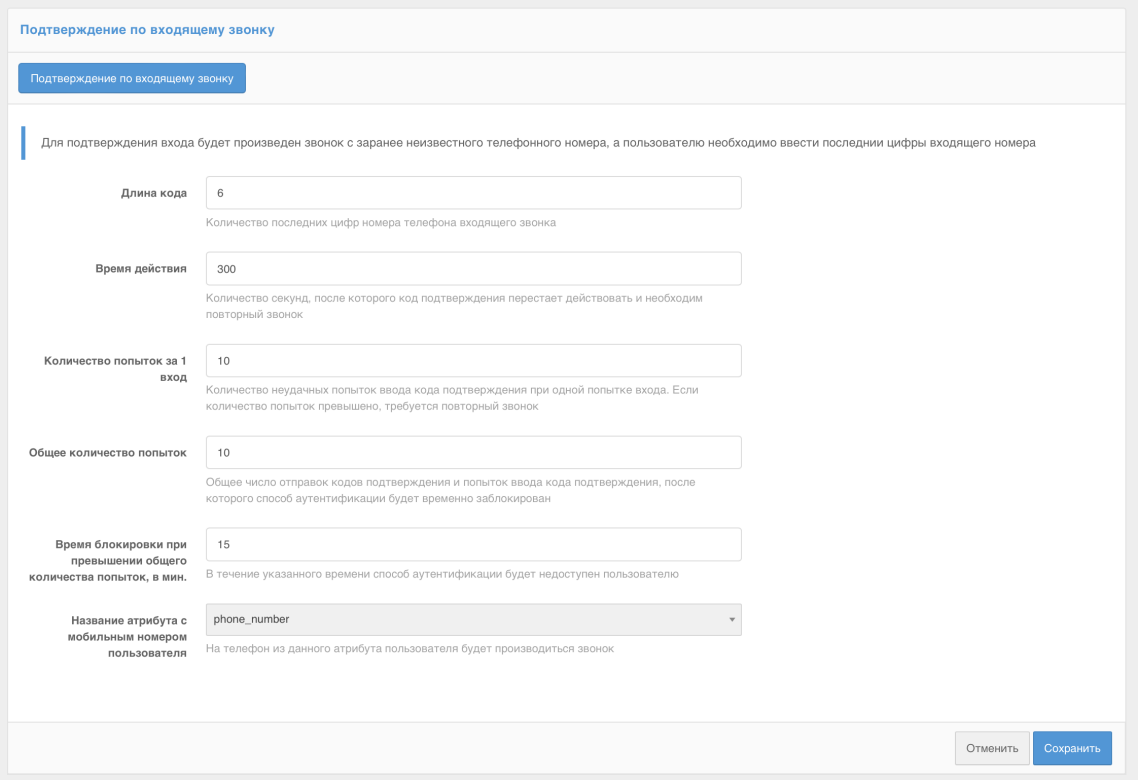
```
sudo systemctl restart blitz-idp blitz-console blitz-recovery
```

Шаг 2. Настройка метода в консоли

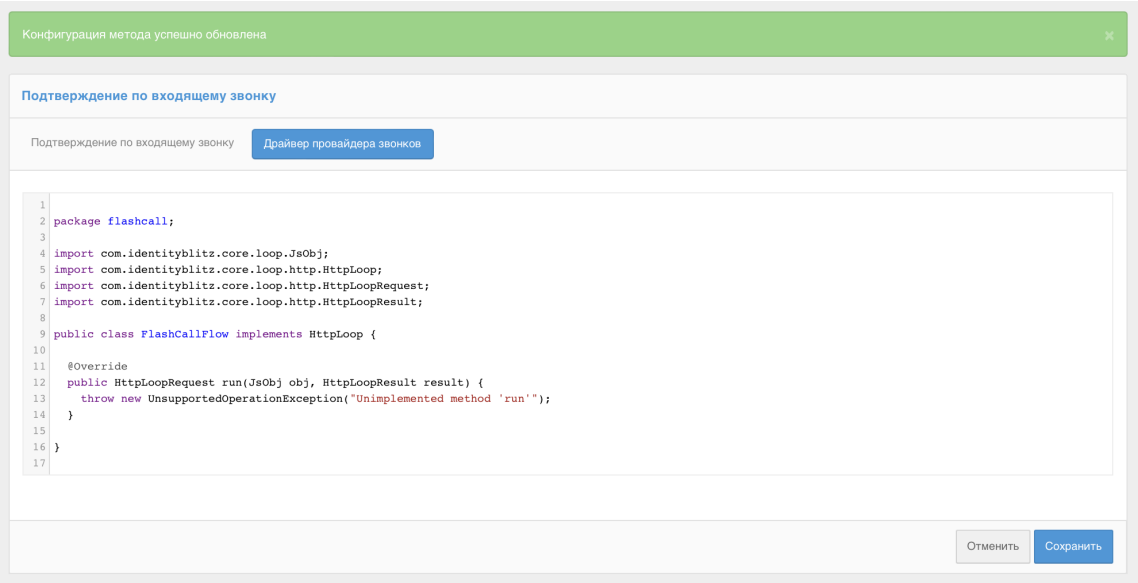
В консоли управления выполните следующие действия:

1. На вкладке Подтверждение по телефонному звонку задайте следующие настройки:

   - `Длина кода`: количество последних цифр номера входящего звонка, которые будут использоваться в качестве кода на втором факторе аутентификации.
   - `Время действия`: количество секунд, после которого код подтверждения перестает действовать и необходим повторный звонок.
   - `Количество попыток за один вход`: количество неудачных попыток ввода кода подтверждения при одной попытке входа. Если количество попыток превышено, требуется повторный звонок.
   - `Общее количество попыток`: общее число отправок кодов подтверждения и попыток ввода кода подтверждения, после которого данный способ аутентификации будет временно заблокирован.
   - `Время блокировки при превышении общего количества попыток, в минутах`: в течение указанного времени данный способ аутентификации будет недоступен пользователю.
   - `Название атрибута с мобильном номером пользователя`: выберите из списка атрибут, в котором хранится номер телефона пользователя для совершения звонка.



Нажмите Сохранить. В результате конфигурация метода будет обновлена и отобразится вкладка Драйвер провайдера звонков.



2. На вкладке Драйвер провайдера звонков задайте процедуру на Java для интеграции с REST-сервисом провайдера, предоставляющего услугу дозвона, по аналогии с примером ниже. Для написания процедуры используйте документацию провайдера и полученные при регистрации в сервисе провайдера настройки.

Список 5: Пример процедуры для интеграции с провайдером Flash Call

```
package flashcall;

import org.slf4j.LoggerFactory;
```

(continues on next page)

(продолжение с предыдущей страницы)

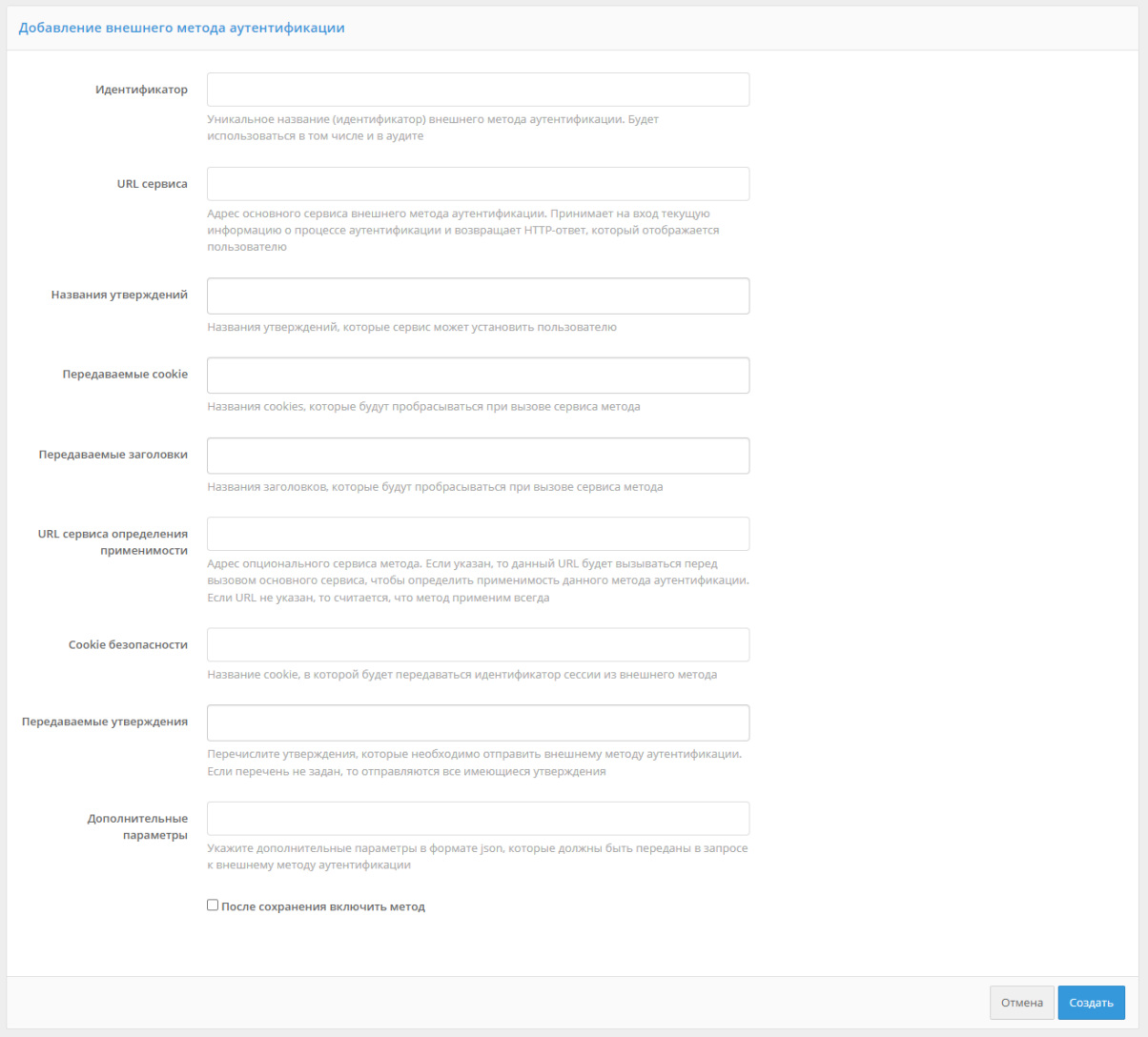
```
import org.slf4j.Logger;
import com.identityblitz.core.loop.http.HttpLoop;
import com.identityblitz.core.loop.http.HttpLoopRequest;
import com.identityblitz.core.loop.http.HttpLoopResult;
import com.identityblitz.core.loop.*;
import com.identityblitz.core.loop.http.*;
import com.identityblitz.json.JObj;
import java.util.Collections;

public class FlashCallFlow implements HttpLoop {
                private final org.slf4j.Logger logger = LoggerFactory.
↪getLogger("com.identityblitz.idp.flow.dynamic");

                @Override
        public HttpLoopRequest run(final JsObj obj, final HttpLoopResult␣
↪result) {
                if (result == null) {
                        final String number = obj.asString("phone_number");
                                logger.trace("### flash call to = {}", number);
                                return HttpLoop.callBuilder("POST", "http://
↪test.flashcall.ru/api/v1")
                                                .withHeader("X-Token", "1234567890")
                                                .withBody(JsObj.empty.addString("id",
↪"test_project").addString("dst_number", number.substring(number.length() -␣
↪10)))
                                                .withTimeout(20000)
                                                .build(JsObj.empty);
                } else if (result.status() == 200) {
                        final JsObj body = result.body();
                        String callerInfo = body.asString("CallerID").
↪substring(0, body.asString("CallerID").length() - 4) + "****";
                        return HttpLoop.Ok(JsObj.empty.addString("code", body.
↪asString("CallerID")).addString("caller_info", callerInfo));
                } else if (result.status() == 502) {
                        return HttpLoop.error("bad_gateway",
                                                Collections.<String, String>
↪singletonMap("status", "" + result.status()));
                        } else {
                        return HttpLoop.error("wrong_http_status",
                                                Collections.<String, String>
↪singletonMap("status", "" + result.status()));
                }
        }

}
```

Совет: См. *подробнее* (страница 283) об имплементации кастомных ошибок.

3. Включите метод Подтверждение по входящему звонку в списке методов на вкладке Аутентификация -> Второй фактор.

Настройка внешнего метода аутентификации

Blitz Identity Provider позволяет разработчикам при внедрении добавить поддержку своего собственного метода аутентификации. Для этого нужно разработать приложение, реализующее логику аутентификации, и подключить этого приложение к Blitz Identity Provider. В Blitz Identity Provider для этого конфигурируется метод аутентификации *«Внешний метод аутентификации»*. Можно реализовать внешний метод аутентификации для работы как в качестве первого, так и в качестве второго фактора аутентификации.



Для настройки использования Blitz Identity Provider с внешним методом аутентификации необходимо:

1. Сконфигурировать новый «внешний» метод первого или второго фактора аутентификации, нажав на ссылку *«Добавить внешний метод аутентификации»*. Указать параметры этого метода аутентификации:
    - идентификатор метода – карточка с названием метода будет отображаться среди методов аутентификации, к методу с данным идентификатором можно будет обращаться из процедуры входа;
    - URL внешнего сервиса;
    - названия утверждений – перечень утверждений, которые внешний метод может установить пользователю;
    - передаваемые `cookie` – перечень названий `cookies`, которые будут пробрасываться при вызове внешнего метода;

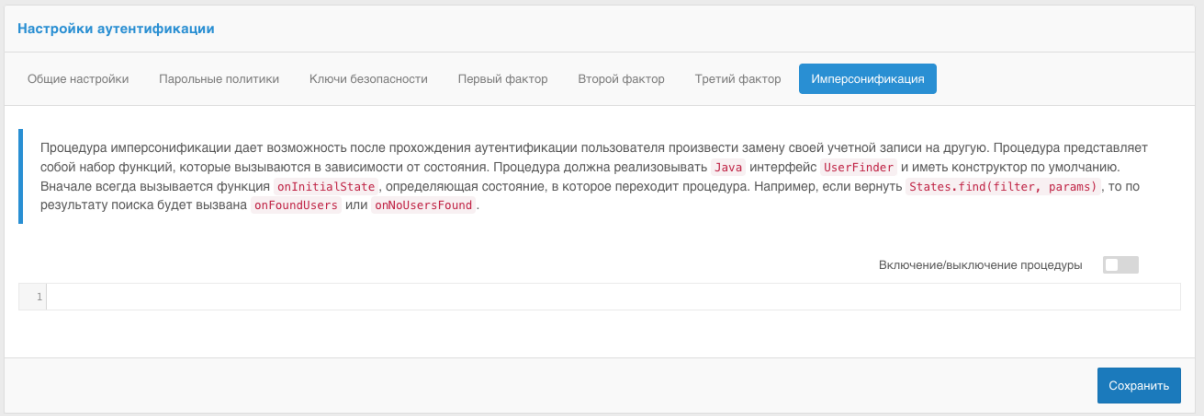
- передаваемые заголовки – перечень заголовков, которые будут пробрасываться при вызове внешнего метода;
- URL сервиса определения применимости – адрес опционального сервиса метода. Если указан, то данный URL будет вызываться перед вызовом основного сервиса, чтобы определить применимость данного метода аутентификации. Если URL не указан, то считается, что метод применим всегда;
- `cookie` безопасности – название `cookie`, в которой будет передаваться идентификатор сессии из внешнего метода.
- передаваемые утверждения – перечень утверждений, которые должны быть переданы внешнему методу (если параметр не задан, то внешнему методу будут переданы все доступные в сеансе входа утверждения);
- дополнительные параметры – задаются в формате JSON. Указанные параметры будут переданы внешнему методу. Это может быть полезно, чтобы иметь возможность конфигурировать настройки внешнего метода аутентификации через консоль управления Blitz Identity Provider.
- после сохранения включить метод – чекбокс, указывающий на то, что необходимо сразу включить метод аутентификации после сохранения настроек.

2. На стороне внешнего метода необходимо предусмотреть обработку запросов на аутентификацию и определение применимости согласно документу *Руководство по интеграции* (страница 383).

Настройка процедуры имперсонификации

Blitz Identity Provider позволяет так настроить процесс входа, что после прохождения идентификации и аутентификации основной учетной записью пользователю можно предложить выбрать для входа одну из его вспомогательных учетных записей.

Процесс выбора вспомогательных учетных записей настраивается на вкладке *«Имперсонификация»*. Для этого разрабатывается на Java процедура имперсонификации. Можно сохранить текст процедуры имперсонификации, и после успешной ее компиляции можно включить процедуру с помощью переключателя *«Включение/выключение процедуры»*.



2.2.3 Внешние поставщики идентификации

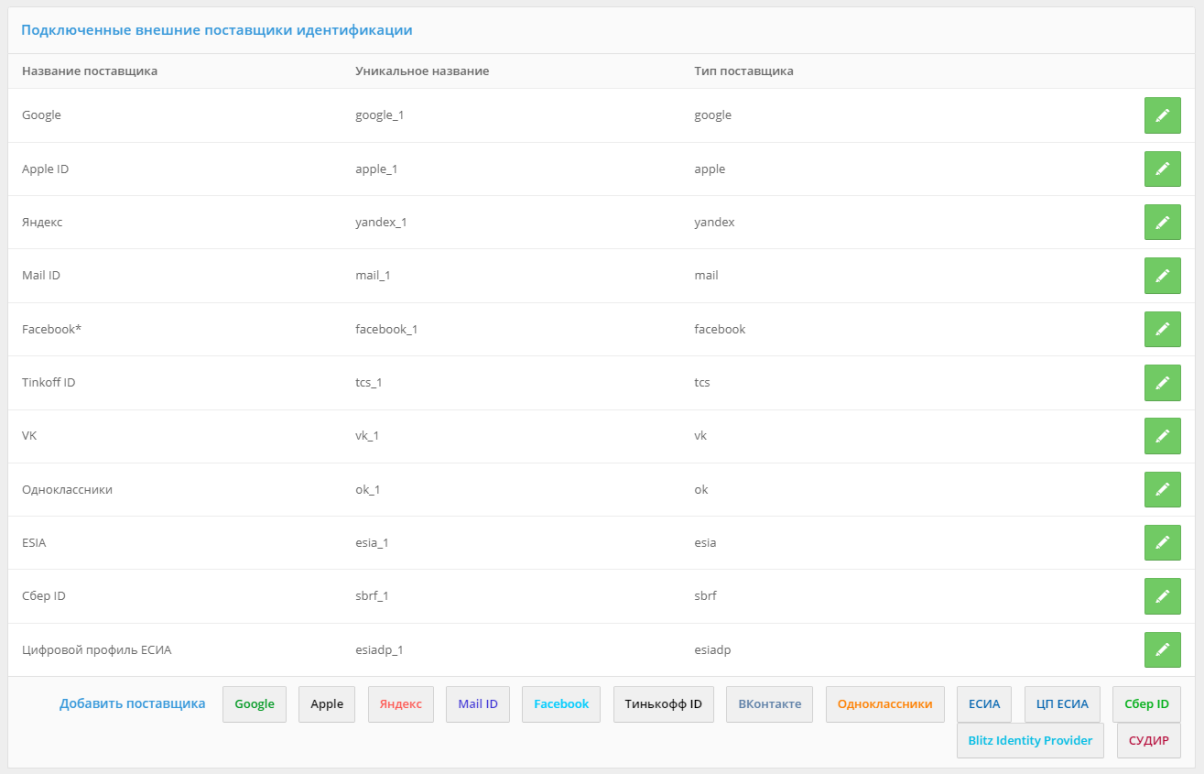
Данный раздел посвящен настройке входа через внешние поставщики идентификации.

Как настроить вход через внешние поставщики идентификации

Настройка входа через внешние поставщики идентификации включает в себя следующие шаги:

1. Выполните настройки в разделе Поставщики идентификации в консоли управления Blitz Identity Provider (см. секции в этом разделе).
2. Выполните настройки на стороне поставщика идентификации.
3. *Включите* (страница 105) возможность входа через данный поставщик идентификации в разделе Аутентификация.

Начальный экран раздела Поставщики идентификации показывает настроенные поставщики и позволяет выбрать для настройки требуемый тип поставщика идентификации.



Настройка поставщика идентификации состоит из следующих шагов:

1. Задание идентификатора поставщика и имени поставщика.
2. Задание настроек подключения к поставщику (описаны отдельно по каждому из поставщиков идентификации).
3. *Задание настроек связывания учетной записи* (страница 185) внешнего поставщика идентификации и учетной записи Blitz Identity Provider. Эти настройки не имеют специфики, зависящей от типа поставщика.

Отечественные поставщики

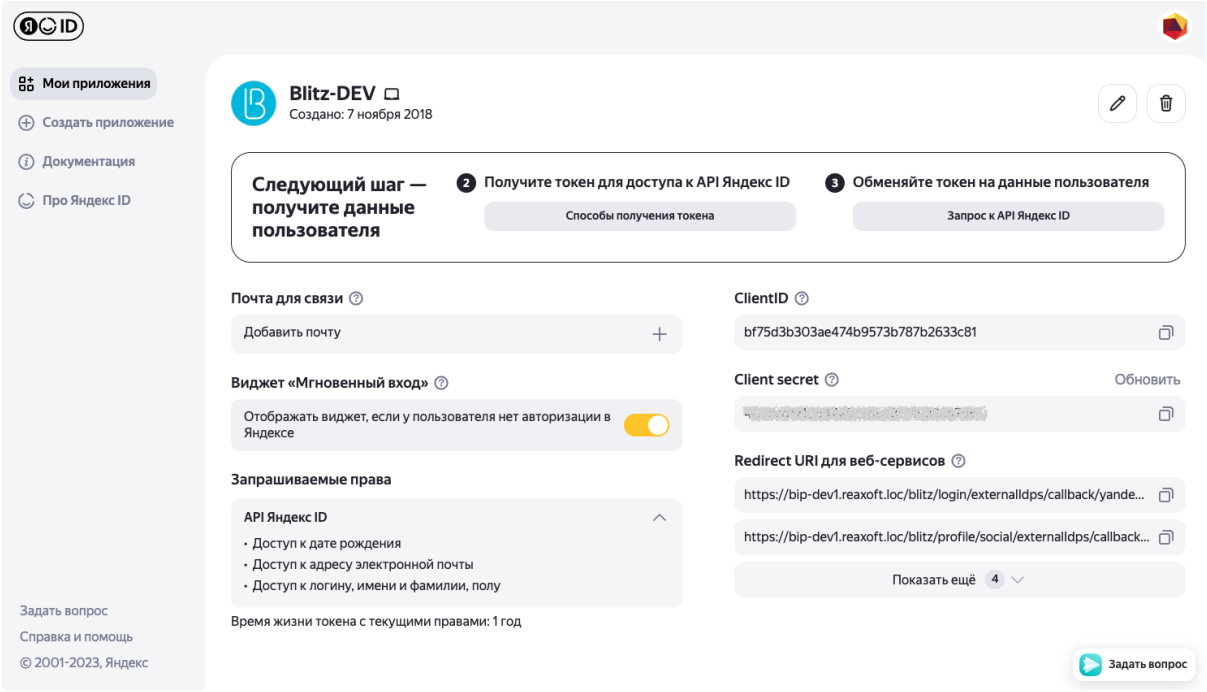
Яндекс

Для конфигурирования входа через Яндекс необходимо выполнить следующие шаги:

1. Перейти в приложение Яндекс.OAuth[34], в котором выполнить следующие операции:
   - нажать на кнопку Зарегистрировать новое приложение;
   - ввести данные приложения, в том числе в настройках Платформы отметить `Веб-сервисы` и ввести в поле `Callback URI` перечень URL, образцы которых Blitz Identity Provider показывает в настройках подключения Яндекс, например:

```
https://login.company.com/blitz/login/externalIdps/callback/yandex/yandex_1/
↪false
https://login.company.com/blitz/profile/social/externalIdps/callbackPopup/
↪yandex/yandex_1
```

   - в перечне доступов раскрыть API Яндекс.Паспорта и отметить `Доступ к адресу электронной почты`, `Доступ к дате рождения` и `Доступ к логину, имени и фамилии, полу`.
   - по результатам регистрации будет сгенерирован `ClientID` приложения и его `Client secret`, они потребуются для последующего ввода в Blitz Identity Provider.



2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип `Яндекс`.
3. Заполнить настройки поставщика идентификации:
   - Идентификатор поставщика;
   - Название поставщика;
   - Идентификатор клиента (`Client ID`), полученный в приложении Яндекс.OAuth;
   - Секрет клиента (`Client secret`), полученный в приложении Яндекс.OAuth;
   - URL для авторизации;

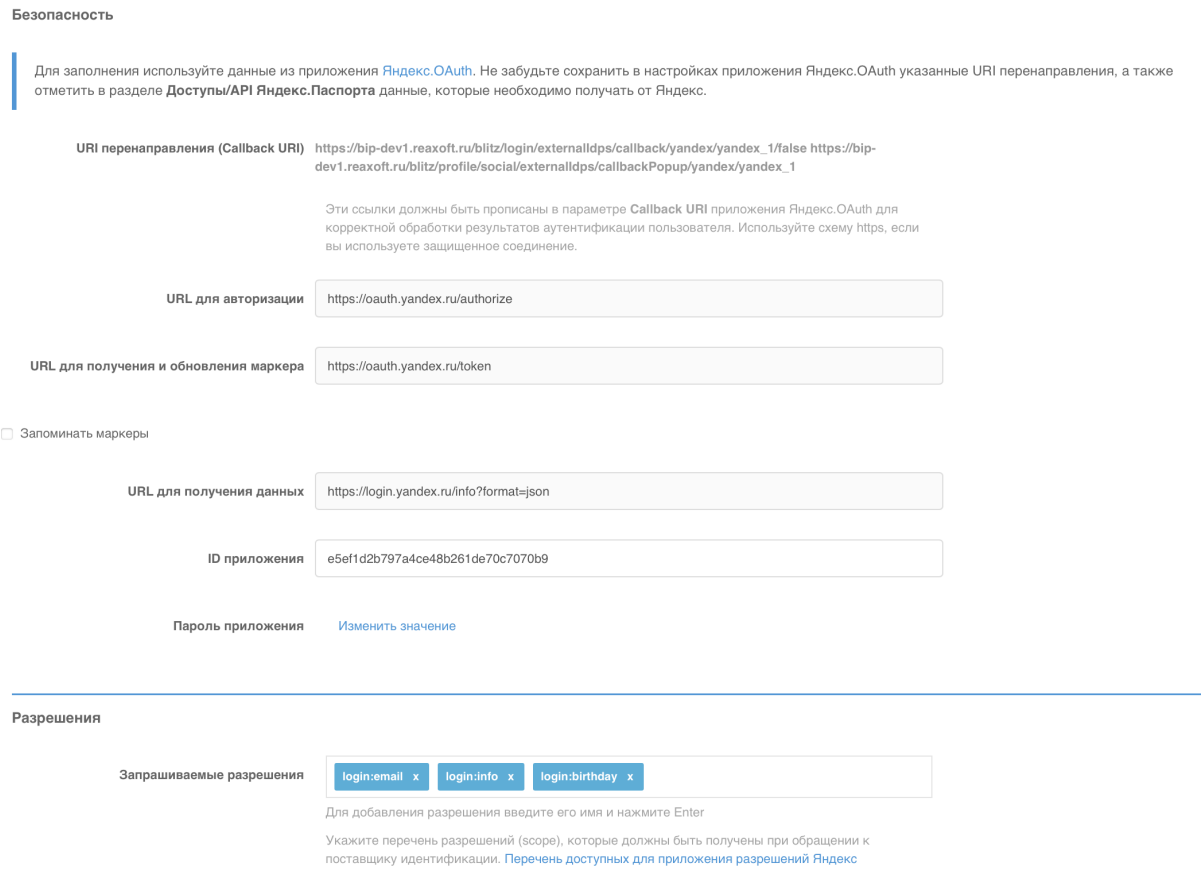
[34] https://oauth.yandex.ru/

- URL для получения и обновления маркера;

> **Примечание:** Если маркеры доступа пользователей необходимо сохранять в базу данных, установите флажок `Запоминать маркеры`. В результате маркеры будут сохраняться в следующих случаях:
>
> – при входе пользователя;
>
> – при привязке внешнего поставщика в Личном кабинете;
>
> – при привязке внешнего поставщика через REST API v2;
>
> – при регистрации пользователя через внешнего поставщика.

- URL для получения данных;
- Запрашиваемые разрешения (`scope`), предусмотренные в Яндекс.OAuth – для указанных ранее доступов следует указать `login:email`, `login:info` и `login:birthday`.

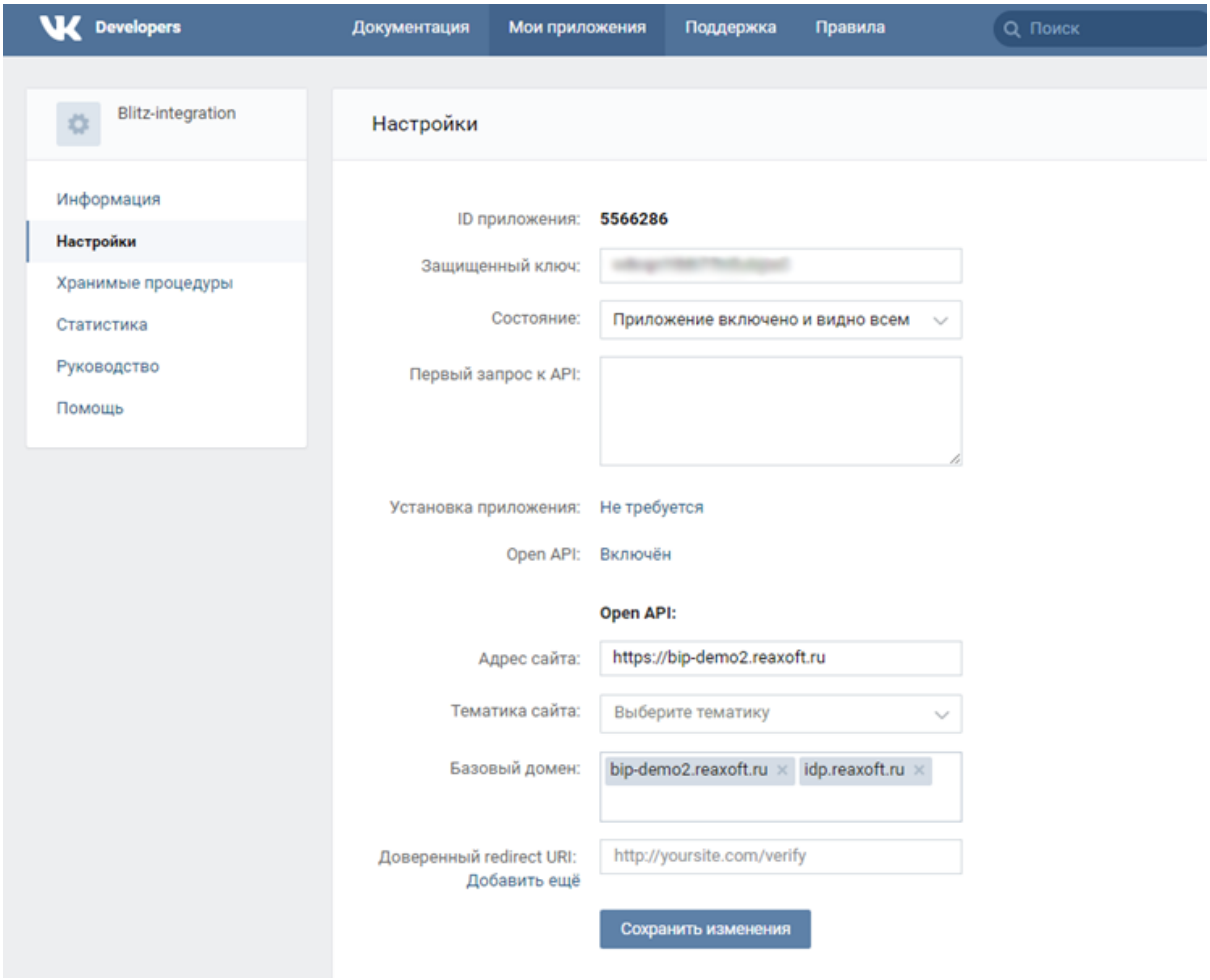
4. Настроить правила связывания.
5. В разделе Аутентификация консоли управления включить использование метода аутентификации с использованием поставщика идентификации Яндекс.



ВКонтакте

Для конфигурирования входа через ВКонтакте необходимо выполнить следующие шаги:

1. Перейти в панель VK для разработчиков[35], в котором выполнить следующие операции:
   - перейти в раздел Мои приложения;
   - выбрать пункт Создать приложение;
   - выбрать тип создаваемого приложения – `Веб-сайт`, указать его название, адрес, и домен;
   - в появившемся окне настроек приложения прописать базовый домен приложения (должен совпадать с доменом, на котором установлен Blitz Identity Provider).



2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип `ВКонтакте`.
3. Заполнить настройки поставщика идентификации:
   - Идентификатор поставщика;
   - Название поставщика;
   - Версия – указать используемую версию API ВКонтакте (например, `5.53`);
   - ID приложения, полученный в панели VK для разработчиков;
   - Защищенный ключ, полученный в панели VK для разработчиков;
   - URL для авторизации;

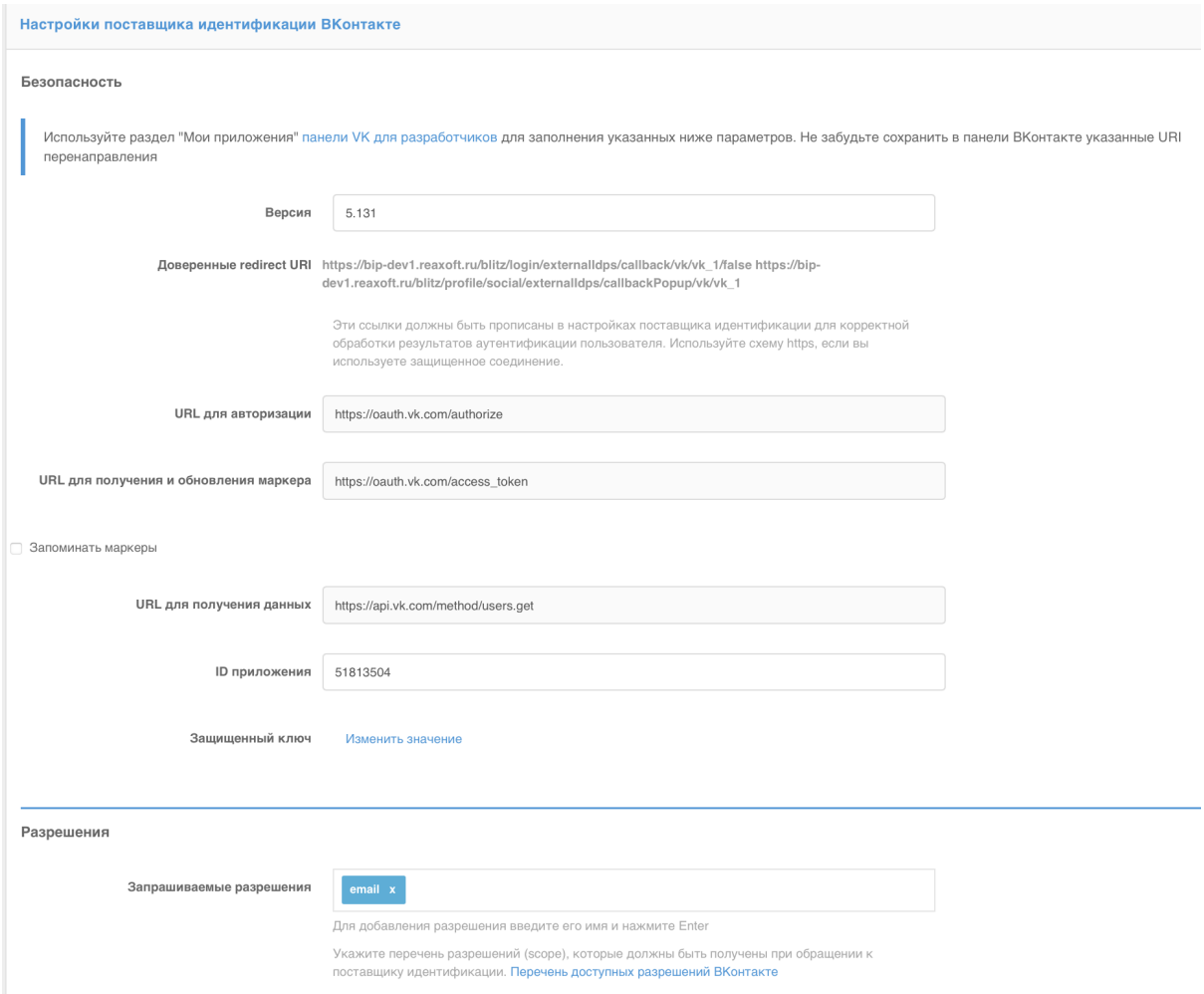
[35] https://new.vk.com/dev

- URL для получения и обновления маркера;

> **Примечание:** Если маркеры доступа пользователей необходимо сохранять в базу данных, установите флажок `Запоминать маркеры`. В результате маркеры будут сохраняться в следующих случаях:
>
> – при входе пользователя;
>
> – при привязке внешнего поставщика в Личном кабинете;
>
> – при привязке внешнего поставщика через REST API v2;
>
> – при регистрации пользователя через внешнего поставщика.

- URL для получения данных;
- Запрашиваемые разрешения, предусмотренные в ВКонтакте[36].

4. Настроить правила связывания.
5. В разделе Аутентификация консоли управления включить использование метода аутентификации с использованием поставщика идентификации ВКонтакте.



[36] https://new.vk.com/dev/permissions

Одноклассники

Для конфигурирования входа через Одноклассники необходимо выполнить следующие шаги:

1. Перейти на страницу OAuth авторизация[37], где выполнить следующие операции:
   - зарегистрироваться в сети Одноклассники и привязать к своему аккаунту email – на этот email будут приходить письма, содержащие регистрационные данные приложений;
   - получить права разработчика по ссылке `https://ok.ru/devaccess`;
   - зарегистрировать свое приложение и получить `Application ID`, публичный ключ приложения и секретный ключ приложения;
   - запросить следующие права для приложения: `VALUABLE_ACCESS`, `LONG_ACCESS_TOKEN`, `GET_EMAIL`;
   - прописать перечень разрешённых `redirect_uri`, образцы которых Blitz Identity Provider показывает в настройках подключения сети Одноклассники, например:

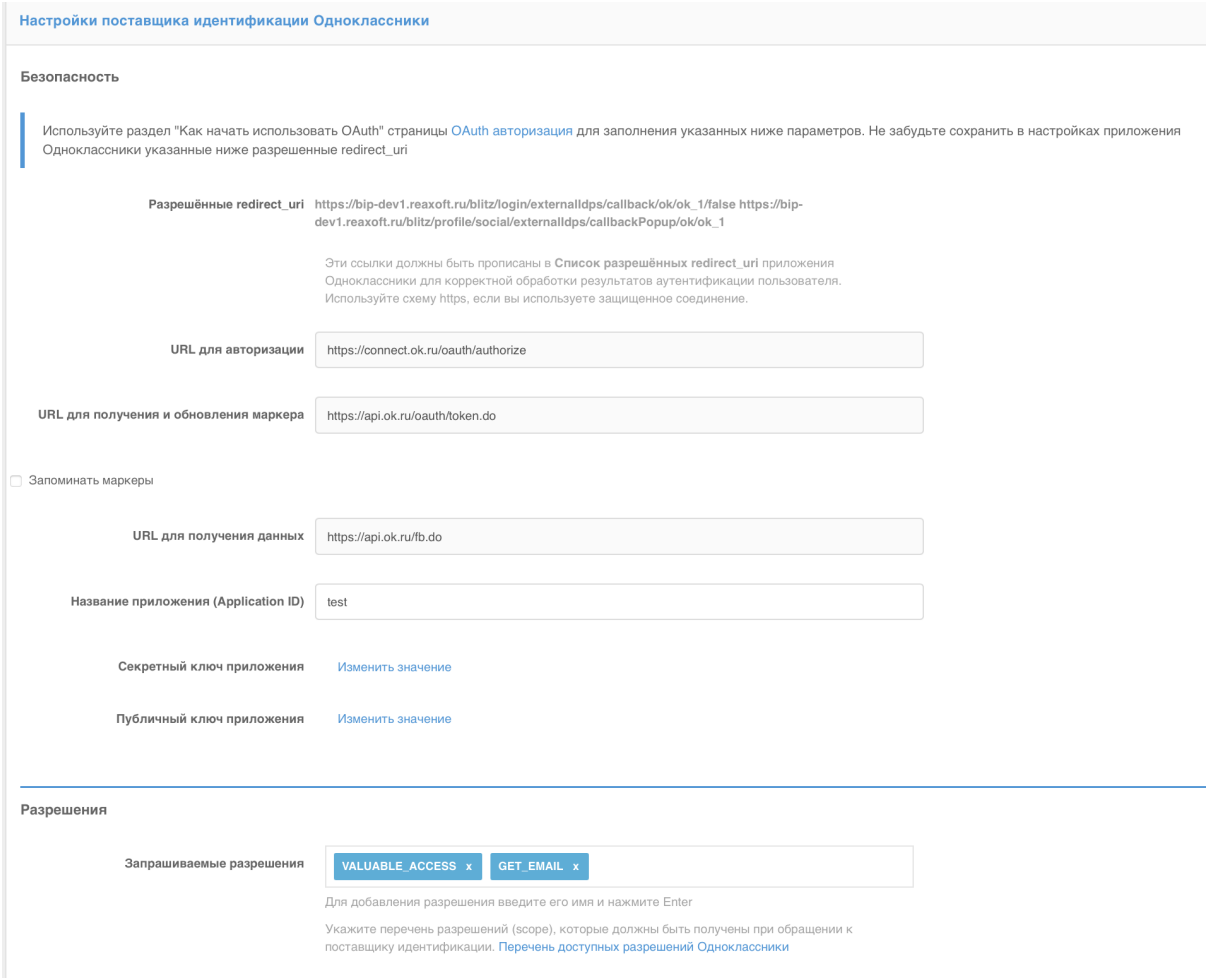
```
https://login.company.com/blitz/login/externalIdps/callback/ok/ok_1/false
https://login.company.com/blitz/profile/social/externalIdps/callbackPopup/ok/
↪ok_1
```

2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип `Одноклассники`.
3. Заполнить настройки поставщика идентификации:
   - Идентификатор поставщика;
   - Название поставщика;
   - Название приложения (`Application ID`);
   - Секретный ключ приложения;
   - Публичный ключ приложения;
   - URL для авторизации;
   - URL для получения и обновления маркера;

     Примечание: Если маркеры доступа пользователей необходимо сохранять в базу данных, установите флажок `Запоминать маркеры`. В результате маркеры будут сохраняться в следующих случаях:
     - при входе пользователя;
     - при привязке внешнего поставщика в Личном кабинете;
     - при привязке внешнего поставщика через REST API v2;
     - при регистрации пользователя через внешнего поставщика.

   - URL для получения данных;
   - Запрашиваемые разрешения.
4. Настроить правила связывания.
5. В разделе Аутентификация консоли управления включить использование метода аутентификации с использованием поставщика идентификации Одноклассники.

[37] https://apiok.ru/ext/oauth/



Mail ID

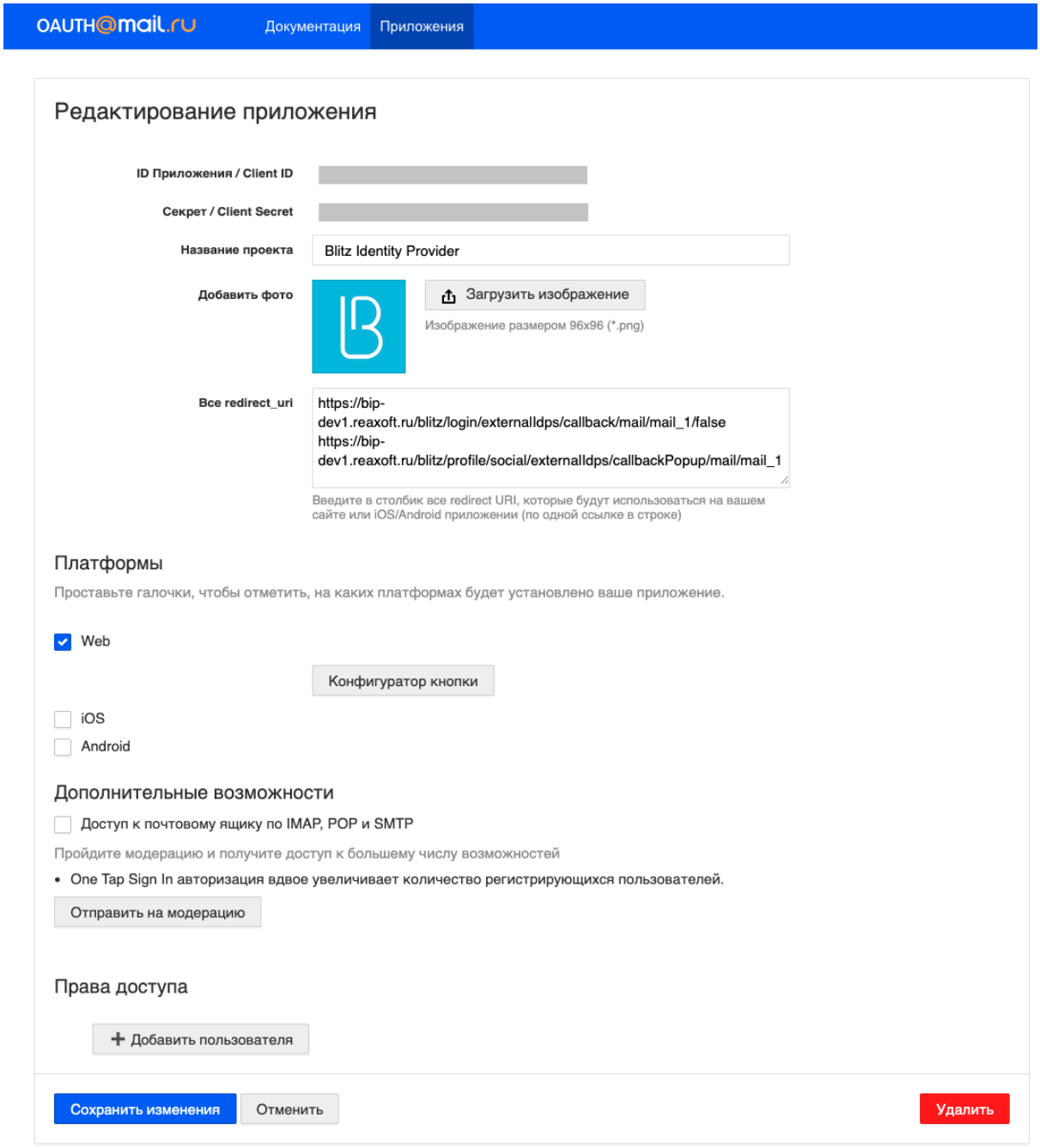
Для конфигурирования входа через Mail ID необходимо выполнить следующие шаги:

1. Перейти на страницу Создание приложения[38] в OAuth@Mail.ru, где выполнить следующие операции:
   - нажать на кнопку Создать приложение;
   - аутентифицировать под учетной записью Mail.ru;
   - ввести данные приложения, название приложения;
   - в поле `Все redirect_uri` указать перечень URI перенаправления, образцы которых Blitz Identity Provider показывает в настройках подключения Mail ID, например:

```
https://login.company.com/blitz/login/externalIdps/callback/mail/mail_1/false
https://login.company.com/blitz/profile/social/externalIdps/callbackPopup/mail/
↪mail_1
```

   - в блоке Платформы поставить галочку на Web;
   - по результатам регистрации будет сгенерирован `ClientID` приложения и его `Client secret`, они потребуются для последующего ввода в Blitz Identity Provider.

[38] https://help.mail.ru/developers/oauth/app



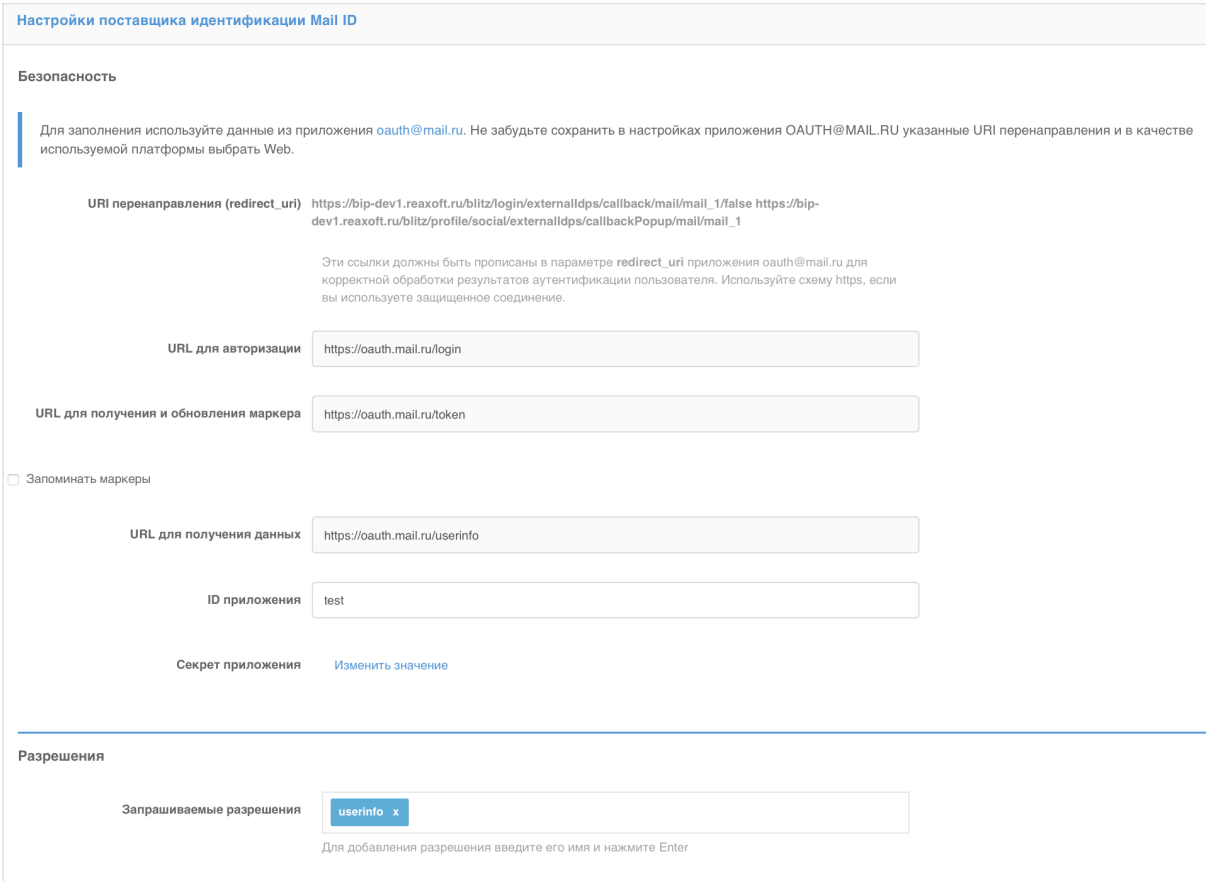
2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип `Mail ID`.
3. Заполнить настройки поставщика идентификации:
    - Идентификатор поставщика;
    - Название поставщика;
    - Идентификатор клиента (ID приложения), полученный ранее;
    - Секрет приложения, полученный ранее;
    - URL для авторизации;
    - URL для получения и обновления маркера;

Примечание: Если маркеры доступа пользователей необходимо сохранять в базу данных, установите флажок `Запоминать маркеры`. В результате маркеры будут сохраняться в следующих случаях:

- при входе пользователя;
- при привязке внешнего поставщика в Личном кабинете;
- при привязке внешнего поставщика через REST API v2;
- при регистрации пользователя через внешнего поставщика.

- URL для получения данных;
- Запрашиваемые разрешения (`scope`), например, `userinfo`.

4. Настроить правила связывания.
5. В разделе Аутентификация консоли управления включить использование метода аутентификации с использованием поставщика идентификации Mail ID.



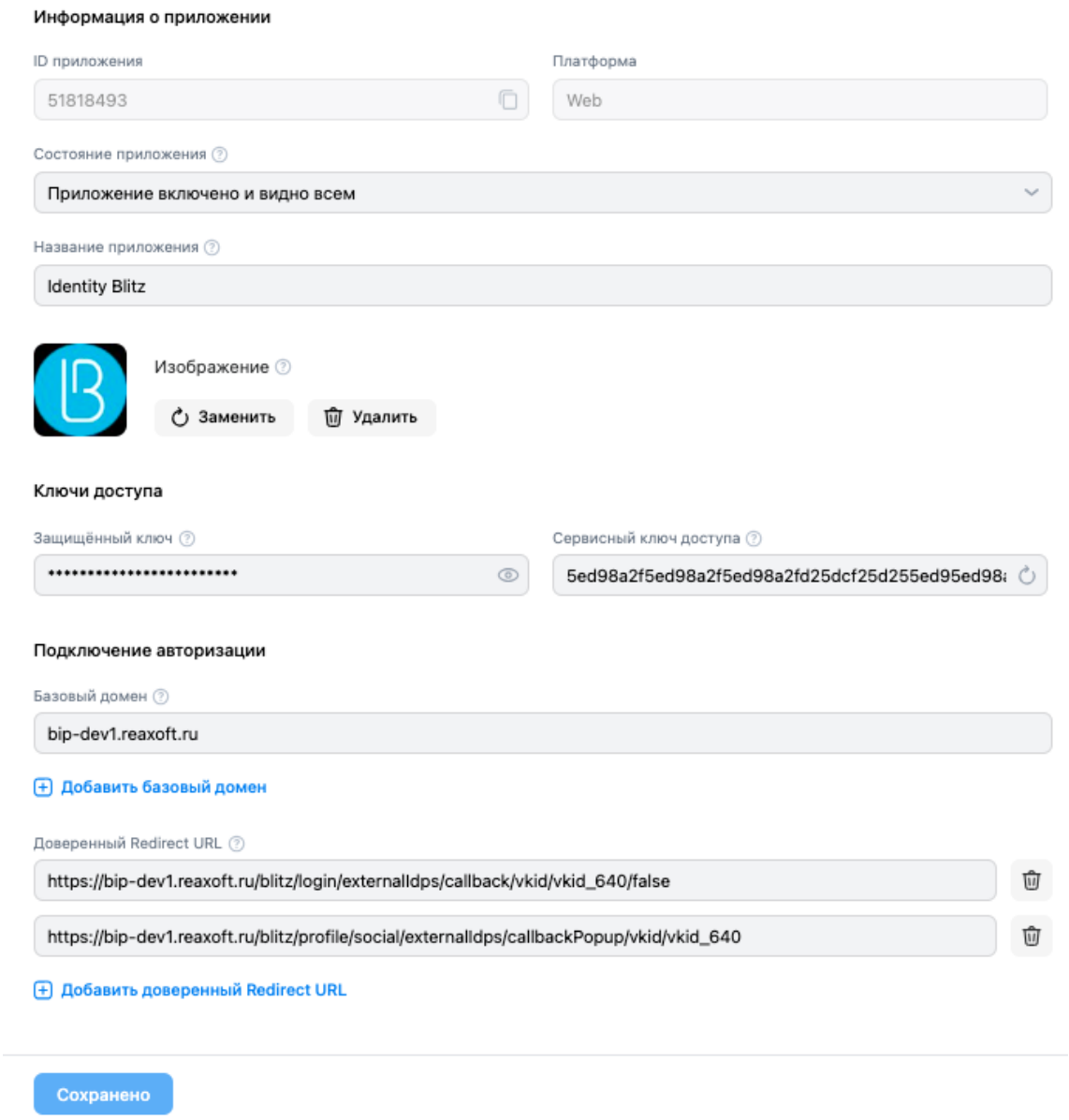
VK ID

Для конфигурирования входа через учетную запись VK ID выполните следующие настройки:

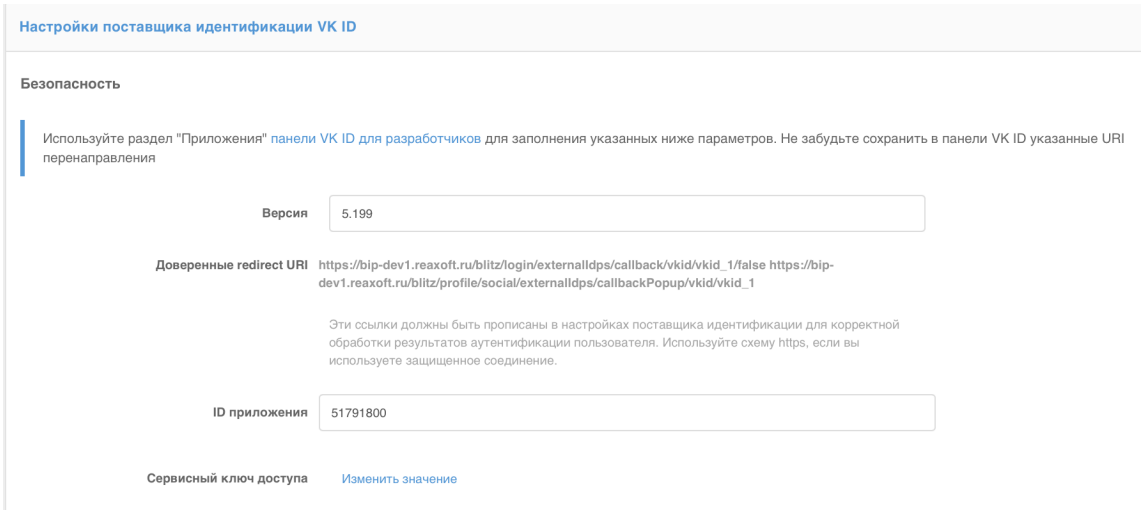
1. Перейдите в консоли управления Blitz Identity Provider на вкладку Поставщики идентификации и добавьте поставщика, имеющего тип VK ID.
2. Задайте базовые настройки: идентификатор и название поставщика. Нажмите Далее. Отобразится вкладка Настройки поставщика идентификации VK ID, на которой потребуется ввести данные регистрации приложения Blitz Identity Provider в онлайн-сервисе VK ID для разработчиков.
3. Перейдите в онлайн-сервис VK ID для разработчиков. Если у вас нет аккаунта, создайте его. В разделе Приложения аккаунта зарегистрируйте приложение Blitz Identity Provider. Для этого выполните следующие действия:
    1. Нажмите Добавить приложение.
    2. Шаг 1: введите название приложения, выберите платформу Web, задайте логотип.
    3. Шаг 2: укажите базовый домен Blitz Identity Provider в своей системе и один за другим все доверенные `Redirect URL`, образцы которых Blitz Identity Provider показывает в настройках подключения VK ID, например:

    ```
    https://login.company.ru/blitz/login/externalIdps/callback/vkid/vkid_640/
    ↪false
    https://login.company.ru/blitz/profile/social/externalIdps/callbackPopup/
    ↪vkid/vkid_640
    ```

    4. Нажмите Готово. В результате регистрации будут сгенерированы значения параметров `ID приложения` и `Сервисный ключ доступа`.



4. Вернитесь в Blitz Identity Provider на вкладку Настройки поставщика идентификации VK ID и введите `ID приложения` и `Сервисный ключ доступа`, полученные при регистрации приложения. Нажмите Сохранить.



5. *Настройте* (страница 185) правила связывания.
6. В разделе Аутентификация консоли управления *включите* (страница 105) использование метода аутентификации с использованием поставщика идентификации VK ID.

Единая система идентификации и аутентификации (ЕСИА)

Для конфигурирования входа через ЕСИА выполните следующие шаги:

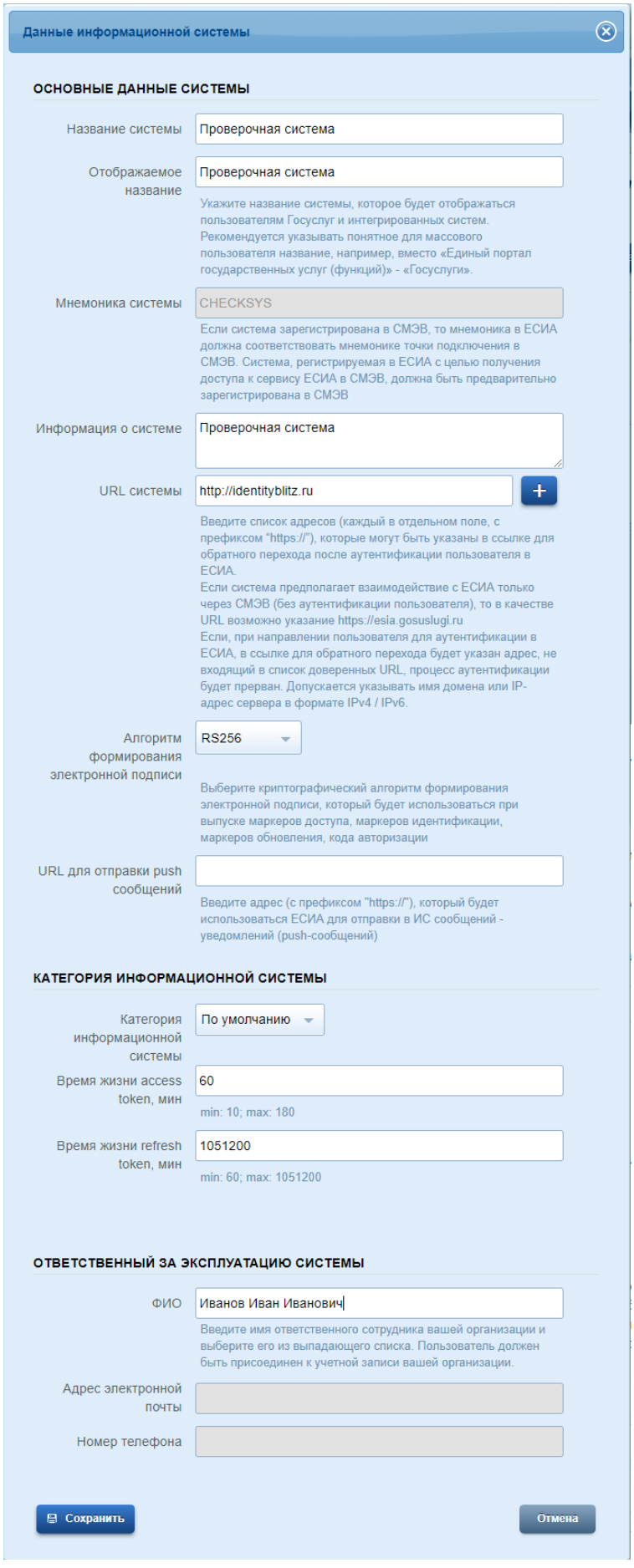
1. Получите в удостоверяющем центре ключ электронной подписи для взаимодействия с ЕСИА и выгрузите сертификат открытого ключа. Произведите конвертацию ключа в формат, *совместимый* (страница 334) с Blitz Identity Provider.

   Примечание: Сертификат ключа понадобится зарегистрировать на Технологическом портале ЕСИА (см. следующий пункт).

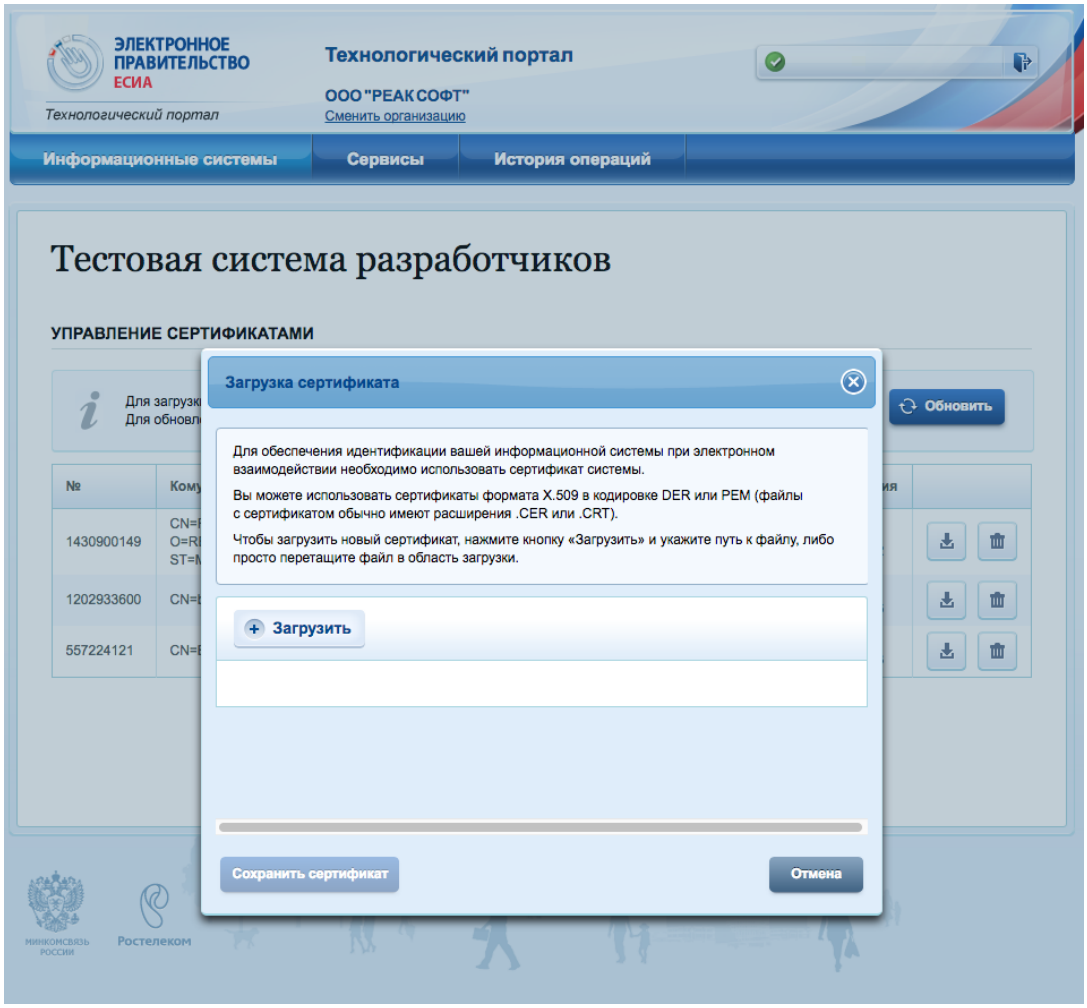
   Внимание: До регистрации ИС в ЕСИА необходимо зарегистрировать учетную запись организации в ЕСИА и дать одному из сотрудников доступ к Технологическому порталу ЕСИА.

2. Зарегистрируйте систему организации на Технологическом портале ЕСИА[39]:
   1. Нажмите на кнопку Добавить систему.
   2. Укажите название системы, отображаемое название, мнемонику системы, список URL системы (URL развернутой системы Blitz Identity Provider с указанием `https`), алгоритм формирования электронной подписи (RS256), а также выберите ответственного сотрудника.

[39] https://esia.gosuslugi.ru/console/tech/


Данные информационной системы
ОСНОВНЫЕ ДАННЫЕ СИСТЕМЫ
Название системы
Проверочная система
Отображаемое название
Проверочная система
Укажите название системы, которое будет отображаться пользователям Госуслуг и интегрированных систем. Рекомендуется указывать понятное для массового пользователя название, например, вместо «Единый портал государственных услуг (функций)» - «Госуслуги».
Мнемоника системы
CHECKSYS
Если система зарегистрирована в СМЭВ, то мнемоника в ЕСИА должна соответствовать мнемонике точки подключения в СМЭВ. Система, регистрируемая в ЕСИА с целью получения доступа к сервису ЕСИА в СМЭВ, должна быть предварительно зарегистрирована в СМЭВ
Информация о системе
Проверочная система
URL системы
http://identityblitz.ru
+
Введите список адресов (каждый в отдельном поле, с префиксом "https://"), которые могут быть указаны в ссылке для обратного перехода после аутентификации пользователя в ЕСИА.
Если система предполагает взаимодействие с ЕСИА только через СМЭВ (без аутентификации пользователя), то в качестве URL возможно указание https://esia.gosuslugi.ru
Если, при направлении пользователя для аутентификации в ЕСИА, в ссылке для обратного перехода будет указан адрес, не входящий в список доверенных URL, процесс аутентификации будет прерван. Допускается указывать имя домена или IP-адрес сервера в формате IPv4 / IPv6.
Алгоритм формирования электронной подписи
RS256
Выберите криптографический алгоритм формирования электронной подписи, который будет использоваться при выпуске маркеров доступа, маркеров идентификации, маркеров обновления, кода авторизации
URL для отправки push сообщений
Введите адрес (с префиксом "https://"), который будет использоваться ЕСИА для отправки в ИС сообщений - уведомлений (push-сообщений)
КАТЕГОРИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ
Категория информационной системы
По умолчанию
Время жизни access token, мин
60
min: 10; max: 180
Время жизни refresh token, мин
1051200
min: 60; max: 1051200
ОТВЕТСТВЕННЫЙ ЗА ЭКСПЛУАТАЦИЮ СИСТЕМЫ
ФИО
Иванов Иван Иванович
Введите имя ответственного сотрудника вашей организации и выберите его из выпадающего списка. Пользователь должен быть присоединен к учетной записи вашей организации.
Адрес электронной почты
Номер телефона
Сохранить
Отмена

3. Сохраните данные и перейдите к настройке сертификатов информационной системы.
4. Загрузите сертификат системы на Технологическом портале.



3. Перейдите в консоль управления Blitz Identity Provider и добавьте поставщика, имеющего тип `ЕСИА`.
4. Задайте базовые настройки ЕСИА:
   - `Идентификатор поставщика`;
   - `Название поставщика`.
5. Заполните настройки поставщика идентификации ЕСИА:
   - `Версия протокола`: выберите требуемую версию протокола для взаимодействия с ЕСИА. В результате поля `URL для авторизации`, `URL для получения и обновления маркера` и `URL для получения данных` будут автоматически заполнены актуальными значениями для промышленной среды ЕСИА. При необходимости измените их на значения для тестовой среды.

> **Примечание:**
>
> – `URL для авторизации`: адрес обработчика ЕСИА, вызываемого из браузера.
>
> Версия протокола 1:
>
> `https://esia-portal1.test.gosuslugi.ru/aas/oauth2/ac` (ТЕСТ ЕСИА)
>
> или
>
> `https://esia.gosuslugi.ru/aas/oauth2/ac` (ПРОД ЕСИА).

Версия протокола 2:

`https://esia-portal1.test.gosuslugi.ru/aas/oauth2/v2/ac` (ТЕСТ ЕСИА)

или

`https://esia.gosuslugi.ru/aas/oauth2/v2/ac` (ПРОД ЕСИА)

– `URL для получения и обновления маркера`: адрес обработчика ЕСИА, вызываемого с сервера Blitz Identity Provider для получения маркера доступа.

Версия протокола 1:

`https://esia-portal1.test.gosuslugi.ru/aas/oauth2/te` (ТЕСТ ЕСИА)

или

`https://esia.gosuslugi.ru/aas/oauth2/te` (ПРОД ЕСИА).

Версия протокола 2:

`https://esia-portal1.test.gosuslugi.ru/aas/oauth2/v3/te` (ТЕСТ ЕСИА)

или

`https://esia.gosuslugi.ru/aas/oauth2/v3/te` (ПРОД ЕСИА).

– `URL для получения данных` – адрес обработчика ЕСИА, вызываемого с сервера Blitz Identity Provider для получения данных учетной записи, например, `https://esia-portal1.test.gosuslugi.ru/rs/prns/${prn_oid}` (ТЕСТ ЕСИА) или `https://esia.gosuslugi.ru/rs/prns/${prn_oid}` (ПРОД ЕСИА).

Примечание: Если маркеры доступа пользователей необходимо сохранять в базу данных, установите флажок `Запоминать маркеры`. В результате маркеры будут сохраняться в следующих случаях:

– при входе пользователя;

– при привязке внешнего поставщика в Личном кабинете;

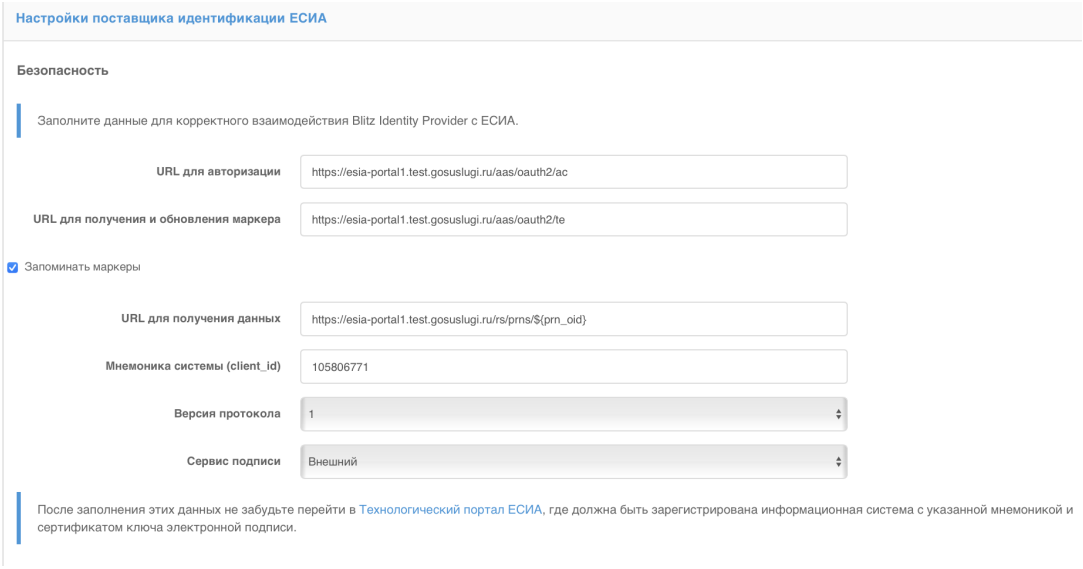
– при привязке внешнего поставщика через REST API v2;

– при регистрации пользователя через внешнего поставщика.

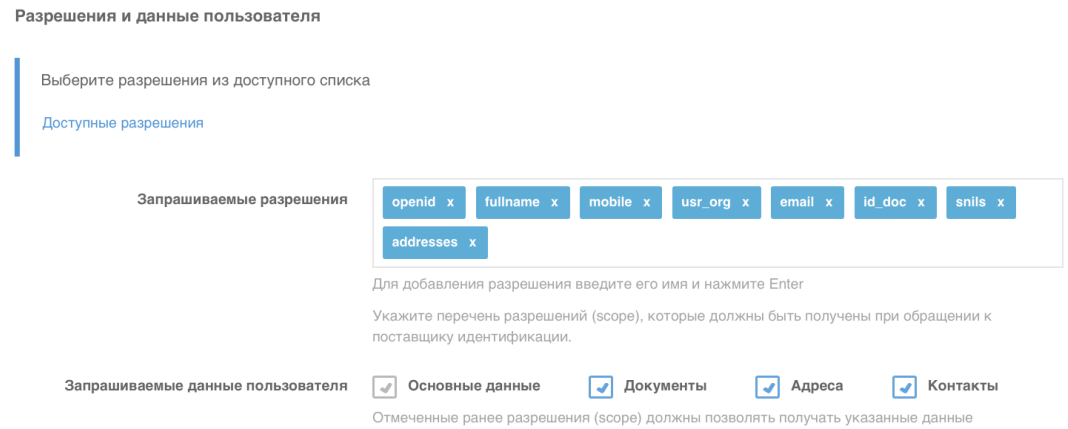
- `Мнемоника системы (client_id)`: введите значение, указанное ранее на Технологическом портале ЕСИА.
- `Сервис подписи`: выберите сервис подписи. Для версии протокола 1 доступны сервисы подписи: `По умолчанию` (КриптоПро CSP), `Внешний`. Для версии протокола 2 доступны сервисы подписи: `По умолчанию` (КриптоПро CSP), `КриптоПро` (КриптоПро JCP).
- `Идентификатор ключа электронной подписи (alias)`, `Пароль доступа к ключу электронной подписи`: идентификатор ключа электронной подписи, загруженный в хранилище Blitz Identity Provider, и пароль к данному ключу.

Примечание: Хранилище указывается в разделе `keystore` конфигурационного файла.

Важно: Сертификат соответствующего ключа электронной подписи должен быть загружен на Технологическом портале ЕСИА.



- `Запрашиваемые разрешения`: перечень запрашиваемых разрешений из ЕСИА.
- `Запрашиваемые данные пользователя`: необходимо отметить те данные, которые следует получать из ЕСИА; эти данные должны быть доступны по запрашиваемым разрешениям.



Важно: Чтобы вход через ЕСИА заработал, необходимо получить официальное разрешение на проведение идентификации и аутентификации пользователей с помощью зарегистрированной системы и получить доступ к тестовой / промышленной среде ЕСИА. Подробнее об этом можно прочитать здесь[40].

6. При необходимости настройте вход через ЕСИА в режиме выбора сотрудника организации.
7. Настройте правила связывания.
8. В разделе Аутентификация консоли управления включите использование метода аутентификации с использованием поставщика идентификации ЕСИА.

[40] https://identityblitz.ru/services/esia-integration/

Цифровой профиль ЕСИА

Для конфигурирования входа через Цифровой профиль ЕСИА выполните следующие действия:

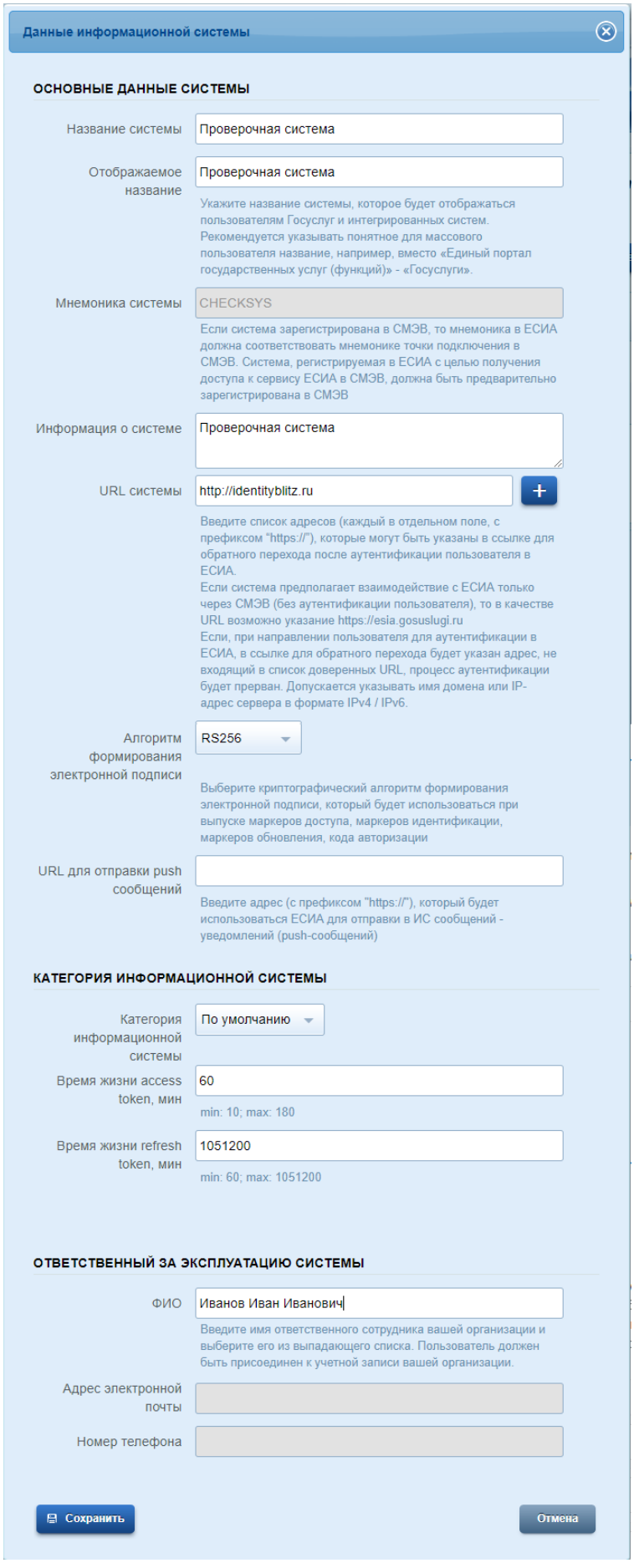
1. Получите в удостоверяющем центре ключ электронной подписи для взаимодействия с ЕСИА и выгрузите сертификат открытого ключа. Произведите конвертацию ключа в формат, *совместимый* (страница 334) с Blitz Identity Provider.

   Примечание: Сертификат ключа понадобится зарегистрировать на Технологическом портале ЕСИА (см. следующий пункт).

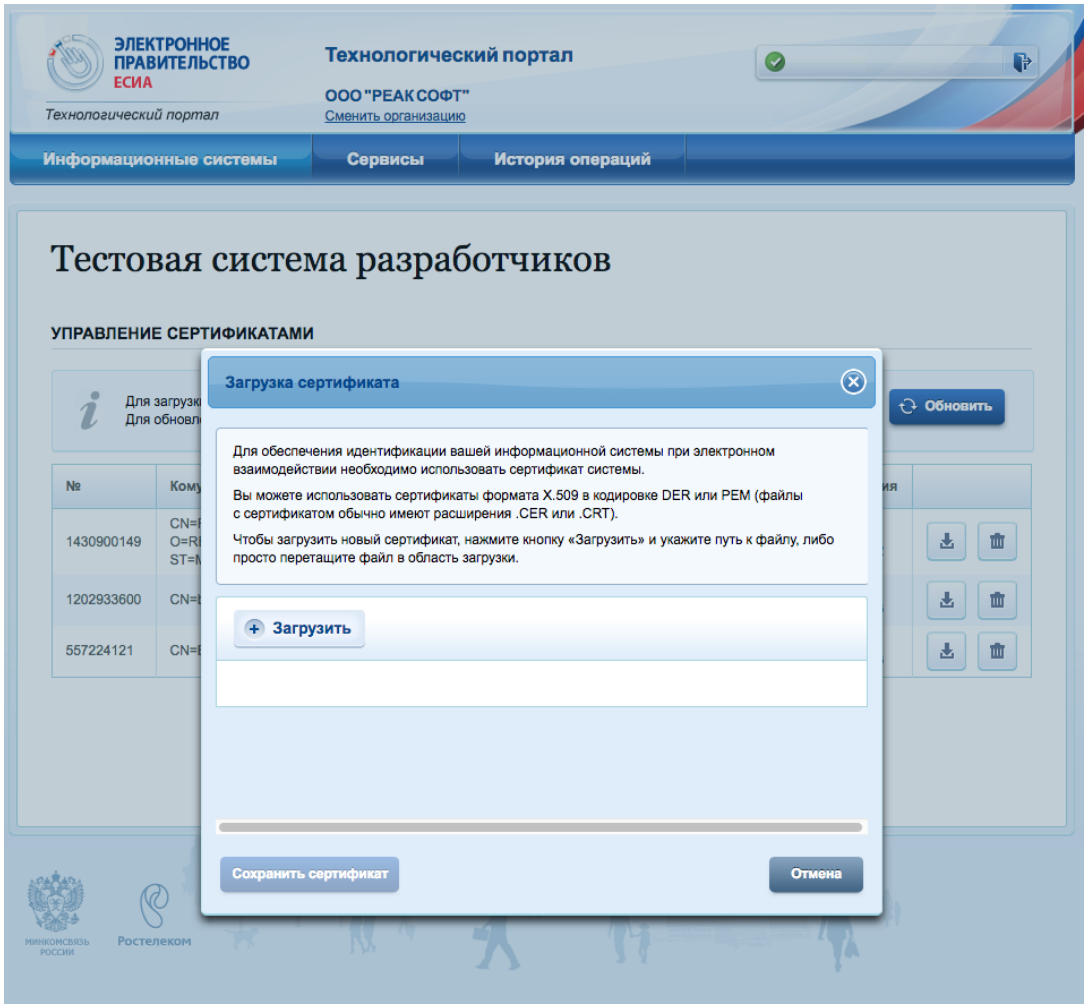
   Внимание: До регистрации ИС в ЕСИА необходимо зарегистрировать учетную запись организации в ЕСИА и дать одному из сотрудников доступ к Технологическому порталу ЕСИА.

2. Зарегистрируйте систему организации на Технологическом портале ЕСИА[41]:
   1. Нажмите на кнопку Добавить систему.
   2. Укажите название системы, отображаемое название, мнемонику системы, список URL системы (URL развернутой системы Blitz Identity Provider с указанием `https`), алгоритм формирования электронной подписи, а также выберите ответственного сотрудника.

[41] https://esia.gosuslugi.ru/console/tech/


Данные информационной системы
ОСНОВНЫЕ ДАННЫЕ СИСТЕМЫ
Название системы
Проверочная система
Отображаемое название
Проверочная система
Укажите название системы, которое будет отображаться пользователям Госуслуг и интегрированных систем. Рекомендуется указывать понятное для массового пользователя название, например, вместо «Единый портал государственных услуг (функций)» - «Госуслуги».
Мнемоника системы
CHECKSYS
Если система зарегистрирована в СМЭВ, то мнемоника в ЕСИА должна соответствовать мнемонике точки подключения в СМЭВ. Система, регистрируемая в ЕСИА с целью получения доступа к сервису ЕСИА в СМЭВ, должна быть предварительно зарегистрирована в СМЭВ
Информация о системе
Проверочная система
URL системы
http://identityblitz.ru
Введите список адресов (каждый в отдельном поле, с префиксом "https://"), которые могут быть указаны в ссылке для обратного перехода после аутентификации пользователя в ЕСИА.
Если система предполагает взаимодействие с ЕСИА только через СМЭВ (без аутентификации пользователя), то в качестве URL возможно указание https://esia.gosuslugi.ru
Если, при направлении пользователя для аутентификации в ЕСИА, в ссылке для обратного перехода будет указан адрес, не входящий в список доверенных URL, процесс аутентификации будет прерван. Допускается указывать имя домена или IP-адрес сервера в формате IPv4 / IPv6.
Алгоритм формирования электронной подписи
RS256
Выберите криптографический алгоритм формирования электронной подписи, который будет использоваться при выпуске маркеров доступа, маркеров идентификации, маркеров обновления, кода авторизации
URL для отправки push сообщений
Введите адрес (с префиксом "https://"), который будет использоваться ЕСИА для отправки в ИС сообщений - уведомлений (push-сообщений)
КАТЕГОРИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ
Категория информационной системы
По умолчанию
Время жизни access token, мин
60
min: 10; max: 180
Время жизни refresh token, мин
1051200
min: 60; max: 1051200
ОТВЕТСТВЕННЫЙ ЗА ЭКСПЛУАТАЦИЮ СИСТЕМЫ
ФИО
Иванов Иван Иванович
Введите имя ответственного сотрудника вашей организации и выберите его из выпадающего списка. Пользователь должен быть присоединен к учетной записи вашей организации.
Адрес электронной почты
Номер телефона
Сохранить
Отмена

3. Сохраните данные и перейдите к настройке сертификатов информационной системы.
4. Загрузите сертификат системы на Технологическом портале.



3. Перейдите в консоль управления Blitz Identity Provider и добавьте поставщика, имеющего тип `ЦП ЕСИА`.
4. Задайте базовые настройки ЦП ЕСИА:
   - `Идентификатор поставщика`;
   - `Название поставщика`.
5. Заполните настройки поставщика идентификации Цифровой профиль ЕСИА:
   - `Версия протокола`: выберите требуемую версию протокола для взаимодействия с ЕСИА. В результате поля `URL для авторизации`, `URL для получения и обновления маркера` и `URL для получения данных` будут автоматически заполнены актуальными значениями для промышленной среды ЕСИА. При необходимости измените их на значения для тестовой среды.

Примечание:

– `URL для авторизации`: адрес обработчика ЕСИА, вызываемого из браузера.

Версия протокола 1:

`https://esia-portal1.test.gosuslugi.ru/aas/oauth2/ac` (ТЕСТ ЕСИА)

или

`https://esia.gosuslugi.ru/aas/oauth2/ac` (ПРОД ЕСИА).

Версия протокола 2:

`https://esia-portal1.test.gosuslugi.ru/aas/oauth2/v2/ac` (ТЕСТ ЕСИА)

или

`https://esia.gosuslugi.ru/aas/oauth2/v2/ac` (ПРОД ЕСИА)

- `URL для получения и обновления маркера`: адрес обработчика ЕСИА, вызываемого с сервера Blitz Identity Provider для получения маркера доступа.

  Версия протокола 1: `https://esia-portal1.test.gosuslugi.ru/aas/oauth2/te` (ТЕСТ ЕСИА) или `https://esia.gosuslugi.ru/aas/oauth2/te` (ПРОД ЕСИА).

  Версия протокола 2: `https://esia-portal1.test.gosuslugi.ru/aas/oauth2/v3/te` (ТЕСТ ЕСИА) или `https://esia.gosuslugi.ru/aas/oauth2/v3/te` (ПРОД ЕСИА).

- `URL для получения данных` – адрес обработчика ЕСИА, вызываемого с сервера Blitz Identity Provider для получения данных учетной записи, например, `https://esia-portal1.test.gosuslugi.ru/digital/api/public/v1` (ТЕСТ ЕСИА) или `https://esia.gosuslugi.ru/digital/api/public/v1` (ПРОД ЕСИА).

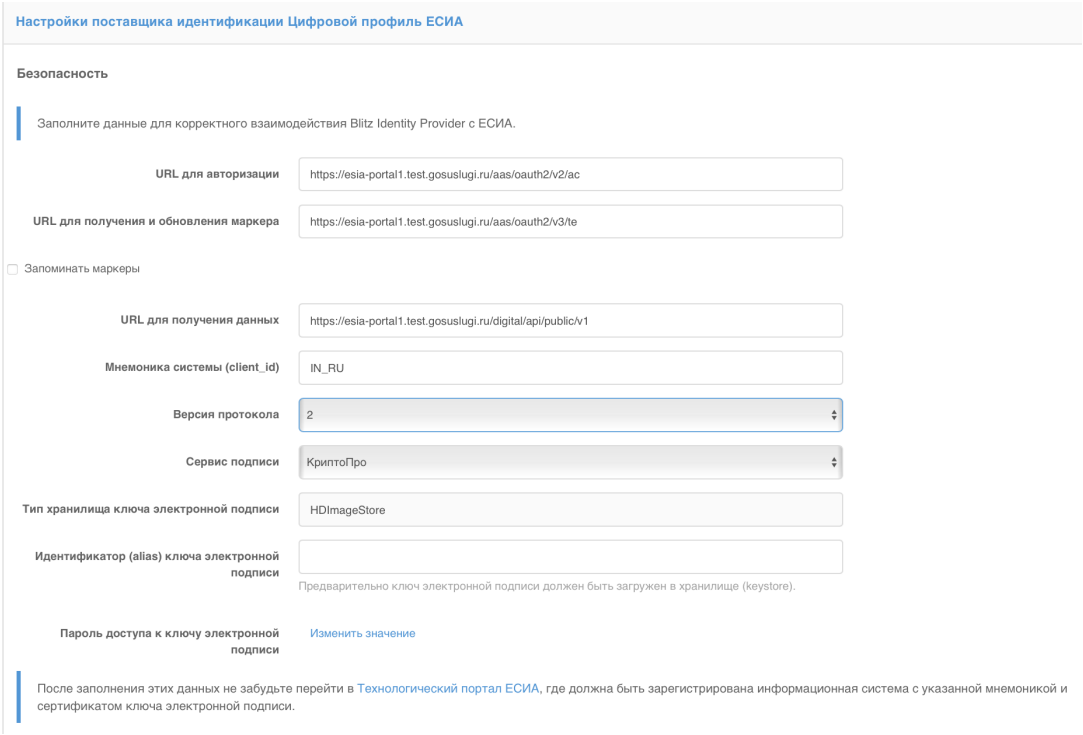
Примечание: Если маркеры доступа пользователей необходимо сохранять в базу данных, установите флажок `Запоминать маркеры`. В результате маркеры будут сохраняться в следующих случаях:

- при входе пользователя;
- при привязке внешнего поставщика в Личном кабинете;
- при привязке внешнего поставщика через REST API v2;
- при регистрации пользователя через внешнего поставщика.

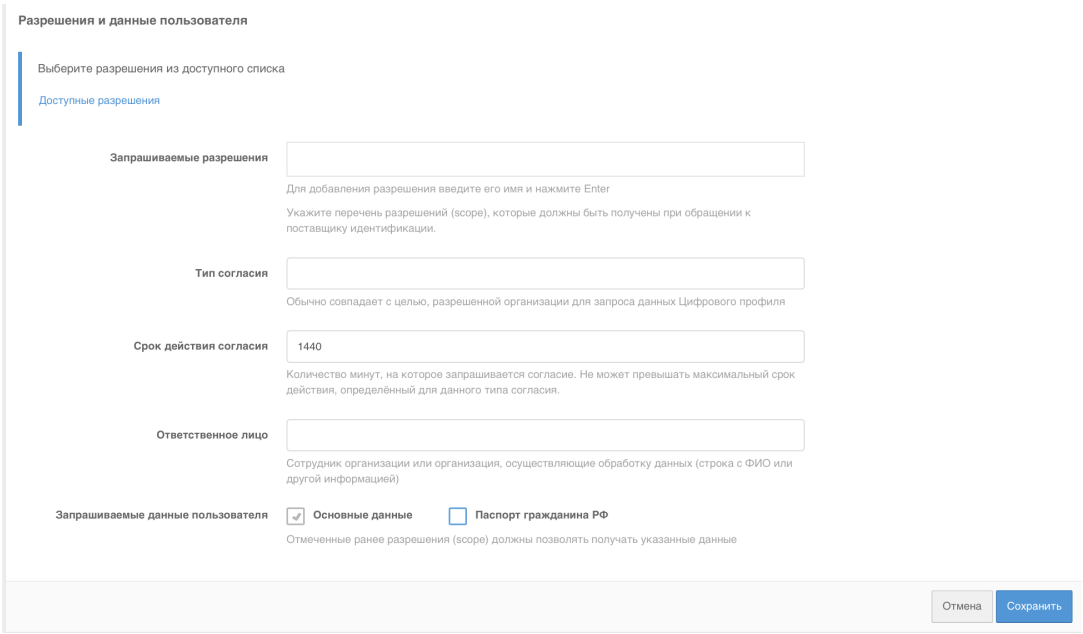
- `Мнемоника системы (client_id)`: введите значение, указанное ранее на Технологическом портале ЕСИА.
- `Сервис подписи`: выберите сервис подписи. Для версии протокола 1 доступны сервисы подписи: `По умолчанию` (КриптоПро CSP), `Внешний`. Для версии протокола 2 доступны сервисы подписи: `По умолчанию` (КриптоПро CSP), `КриптоПро` (КриптоПро JCP).
- `Идентификатор ключа электронной подписи (alias)`, `Пароль доступа к ключу электронной подписи`: идентификатор ключа электронной подписи, загруженный в хранилище Blitz Identity Provider, и пароль к данному ключу.

Примечание: Хранилище указывается в разделе `keystore` конфигурационного файла.

Важно: Сертификат соответствующего ключа электронной подписи должен быть загружен на Технологическом портале ЕСИА.



- `Запрашиваемые разрешения`: перечень запрашиваемых разрешений из ЕСИА.
- `Тип согласия` – запрашиваемый в цифровом профиле тип согласия.
- `Срок действия согласия` – количество минут, на которое запрашивается согласие.
- `Ответственное лицо` – сотрудник организации, ответственный за обработку данных, полученных из цифрового профиля.
- `Запрашиваемые данные пользователя` – необходимо отметить те данные, которые следует получать из цифрового профиля ЕСИА; эти данные должны быть доступны по запрашиваемым разрешениям.



Важно: Чтобы вход через Цифровой профиль ЕСИА заработал, необходимо получить официальное разрешение на проведение идентификации и аутентификации пользователей с помощью зареги-

стрированной системы и получить доступ к тестовой / промышленной среде ЕСИА с доступом к цифровому профилю. Подробнее об этом можно прочитать здесь[42].

6. Настройте правила связывания.
7. В разделе Аутентификация консоли управления включите использование метода аутентификации с использованием поставщика идентификации ЦП ЕСИА.

Сбер ID

Для конфигурирования входа через Сбер ID необходимо выполнить следующие шаги:

1. Зарегистрировать приложение в системе Сбер ID. Для этого воспользоваться инструкцией, размещенной на официальном сайте этого поставщика идентификации[43]. По результатам регистрации необходимо получить:
   - идентификатор клиента (`Client ID`);
   - секрет клиента (`Client Secret`);
   - сертификат системы, подключенной к Сбер ID;
   - сертификат Сбер ID.
2. Настроить защищенный канал связи между организацией и банком с использованием сертификата, полученного от ПАО «Сбербанк» в процессе регистрации.
3. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип `Сбер ID`.
4. Заполнить настройки поставщика идентификации:
   - Идентификатор поставщика;
   - Название поставщика;
   - URL для авторизации – адрес, по которому должна инициироваться аутентификация, например: `https://online.sberbank.ru/CSAFront/oidc/authorize.do`;
   - URL для получения и обновления маркера – адрес, по которому происходит получение и обновление маркера доступа. Должен быть указан внутренний адрес сети, обращение через который обеспечит работу по защищенному каналу связи между организацией и банком;

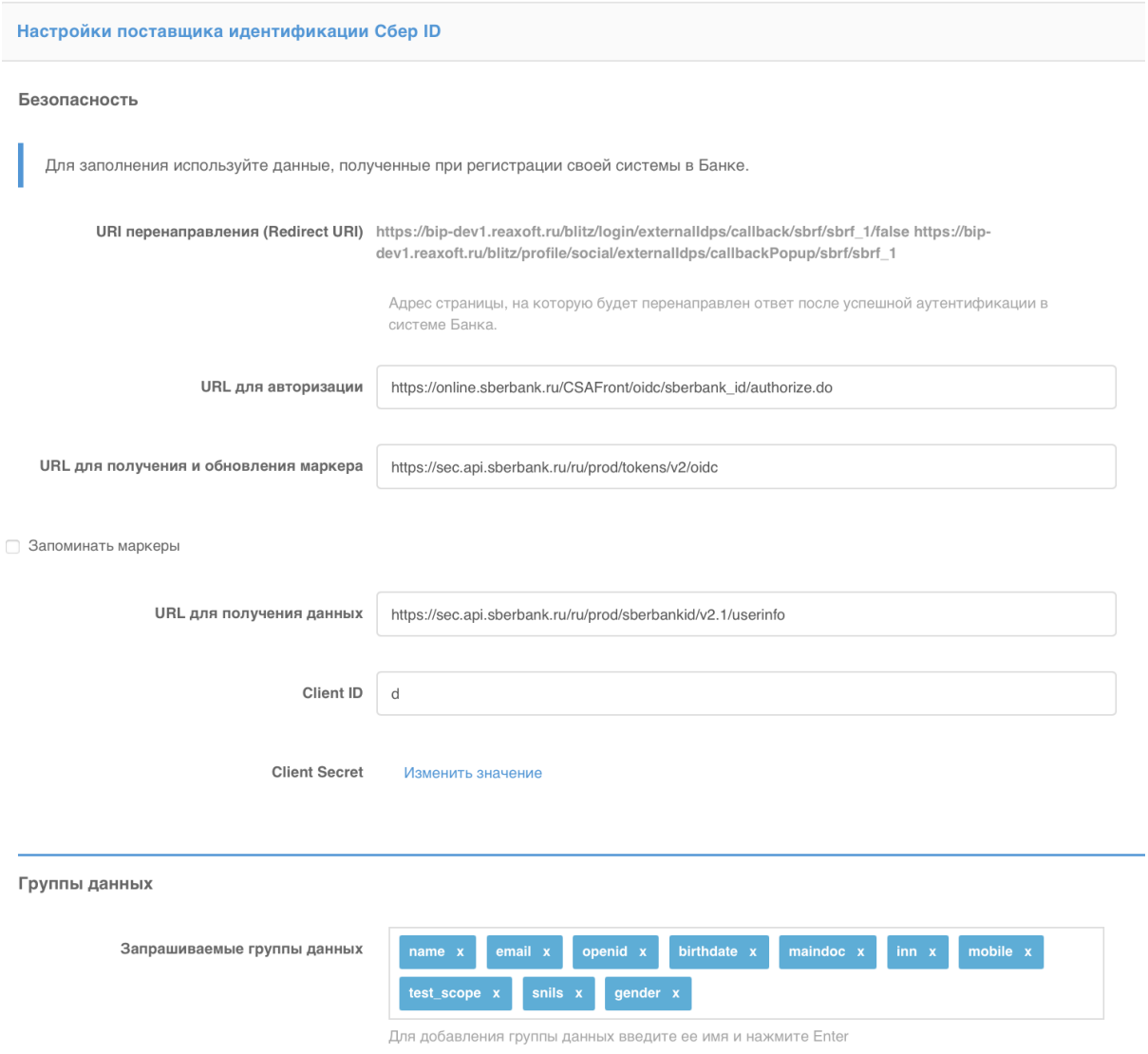
     Примечание: Если маркеры доступа пользователей необходимо сохранять в базу данных, установите флажок `Запоминать маркеры`. В результате маркеры будут сохраняться в следующих случаях:
     - при входе пользователя;
     - при привязке внешнего поставщика в Личном кабинете;
     - при привязке внешнего поставщика через REST API v2;
     - при регистрации пользователя через внешнего поставщика.

   - URL для получения данных – адрес, по которому происходит получение данных пользователя. Должен быть указан внутренний адрес сети, обращение через который обеспечит работу по защищенному каналу связи между организацией и банком;
   - Идентификатор клиента (`Client ID`);
   - Секрет клиента (`Client Secret`);

[42] https://identityblitz.ru/products/esia-bridge/digital-profile/
[43] https://developers.sber.ru/docs/ru/sberid/overview

- Запрашиваемые группы данных – какие группы данных запрашивать из Сбер ID;

5. Настроить правила связывания.
6. В разделе Аутентификация консоли управления включить использование метода аутентификации с использованием поставщика идентификации Сбер ID.



T-ID

Для конфигурирования входа через T-ID необходимо выполнить следующие шаги:

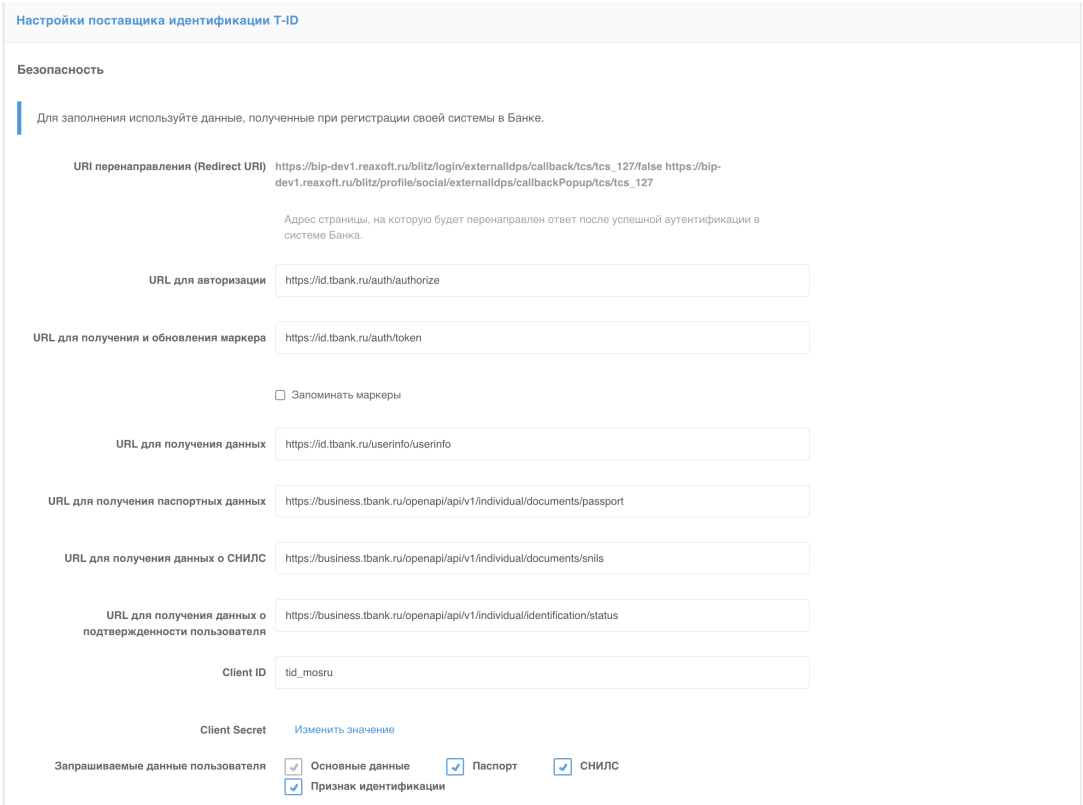
1. Зарегистрировать приложение в системе T-ID. Для этого подать заявку на официальном сайте этого поставщика идентификации[44]. По результатам регистрации необходимо получить:
   - идентификатор клиента (`Client ID`);
   - секрет клиента (`Client Secret`).
2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип `T-ID`.
3. Заполнить настройки поставщика идентификации:
   - `Идентификатор поставщика`;

[44] https://www.tinkoff.ru/corporate/business-solutions/open-api/tinkoff-id/integration/

- `Название поставщика`;
- `URL для авторизации` – адрес, по которому должна инициироваться аутентификация;
- `URL для получения и обновления маркера` – адрес, по которому происходит получение и обновление маркера доступа;

> **Примечание:** Если маркеры доступа пользователей необходимо сохранять в базу данных, установите флажок `Запоминать маркеры`. В результате маркеры будут сохраняться в следующих случаях:
>
> – при входе пользователя;
>
> – при привязке внешнего поставщика в Личном кабинете;
>
> – при привязке внешнего поставщика через REST API v2;
>
> – при регистрации пользователя через внешнего поставщика.

- `URL для получения данных` – адрес, по которому происходит получение данных пользователя;
- `URL для получения паспортных данных` – адрес, по которому происходит получение паспортных данных пользователя;
- `URL для получения данных о СНИЛС` – адрес, по которому происходит получение СНИЛС пользователя;
- `URL для получения данных о подтвержденности пользователя` – адрес, по которому происходит получение признака идентификации пользователя. Признак получают пользователи, которых идентифицировали лично по предоставленному паспорту.
- `Client ID` – идентификатор клиента;
- `Client Secret` – секрет клиента;
- `Запрашиваемые данные пользователя` – перечень запрашиваемых групп данных из T-ID.



4. Настроить правила связывания.
5. В разделе Аутентификация консоли управления включить использование метода аутентификации с использованием поставщика идентификации T-ID.

ВТБ ID

Для конфигурирования входа через ВТБ ID необходимо выполнить следующие шаги:

1. Зарегистрировать приложение в системе ВТБ ID[45]. Для этого подать заявку на официальном сайте этого поставщика идентификации. По результатам регистрации необходимо получить:
   - идентификатор клиента (`Client ID`);
   - секрет клиента (`Client Secret`).
2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип `ВТБ ID`.
3. Заполнить настройки поставщика идентификации:
   - Идентификатор поставщика;
   - Название поставщика;
   - URL для авторизации – адрес, по которому должна инициироваться аутентификация, например: `https://id.vtb.ru`;
   - URL для получения и обновления маркера – адрес, по которому происходит получение и обновление маркера доступа, например: `https://id.vtb.ru/oauth2/token`;

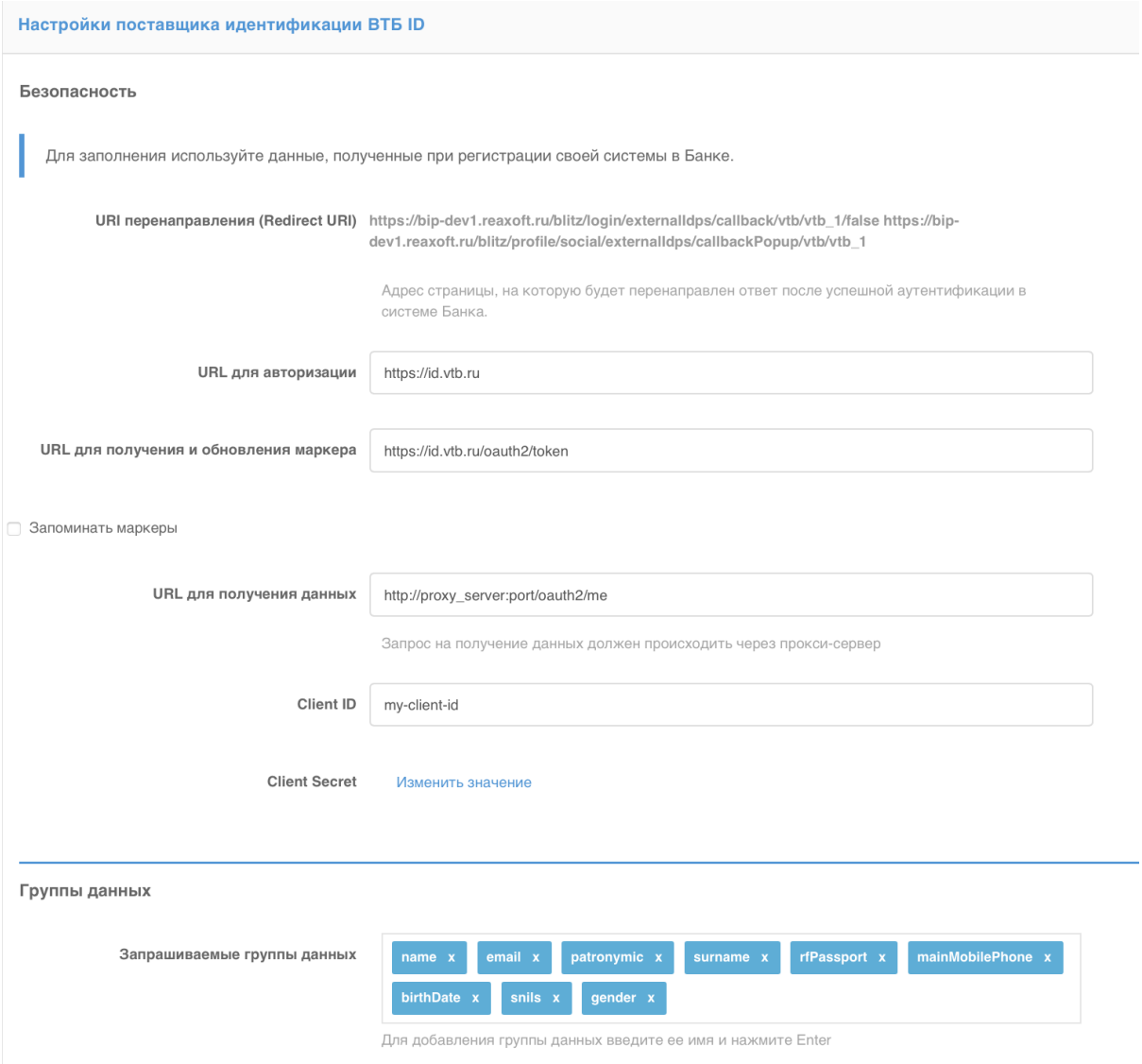
> **Примечание:** Если маркеры доступа пользователей необходимо сохранять в базу данных, установите флажок `Запоминать маркеры`. В результате маркеры будут сохраняться в следующих случаях:

[45] https://developer.vtb.ru/connection

- при входе пользователя;
- при привязке внешнего поставщика в Личном кабинете;
- при привязке внешнего поставщика через REST API v2;
- при регистрации пользователя через внешнего поставщика.

- URL для получения данных – адрес, по которому происходит получение данных пользователя, например: `http://proxy_server:port/oauth2/me`. Подключение к сервису получения данных должно осуществляться через прокси-сервер, на котором должны быть настроены TLS-сертификаты для защищенного взаимодействия с сервисом ВТБ ID;
- Идентификатор клиента (`Client ID`);
- Секрет клиента (`Client Secret`);
- Запрашиваемые группы данных – перечень запрашиваемых групп данных из ВТБ ID;

4. Настроить правила связывания.
5. В разделе Аутентификация консоли управления включить использование метода аутентификации с использованием поставщика идентификации ВТБ ID.



СберБизнес ID

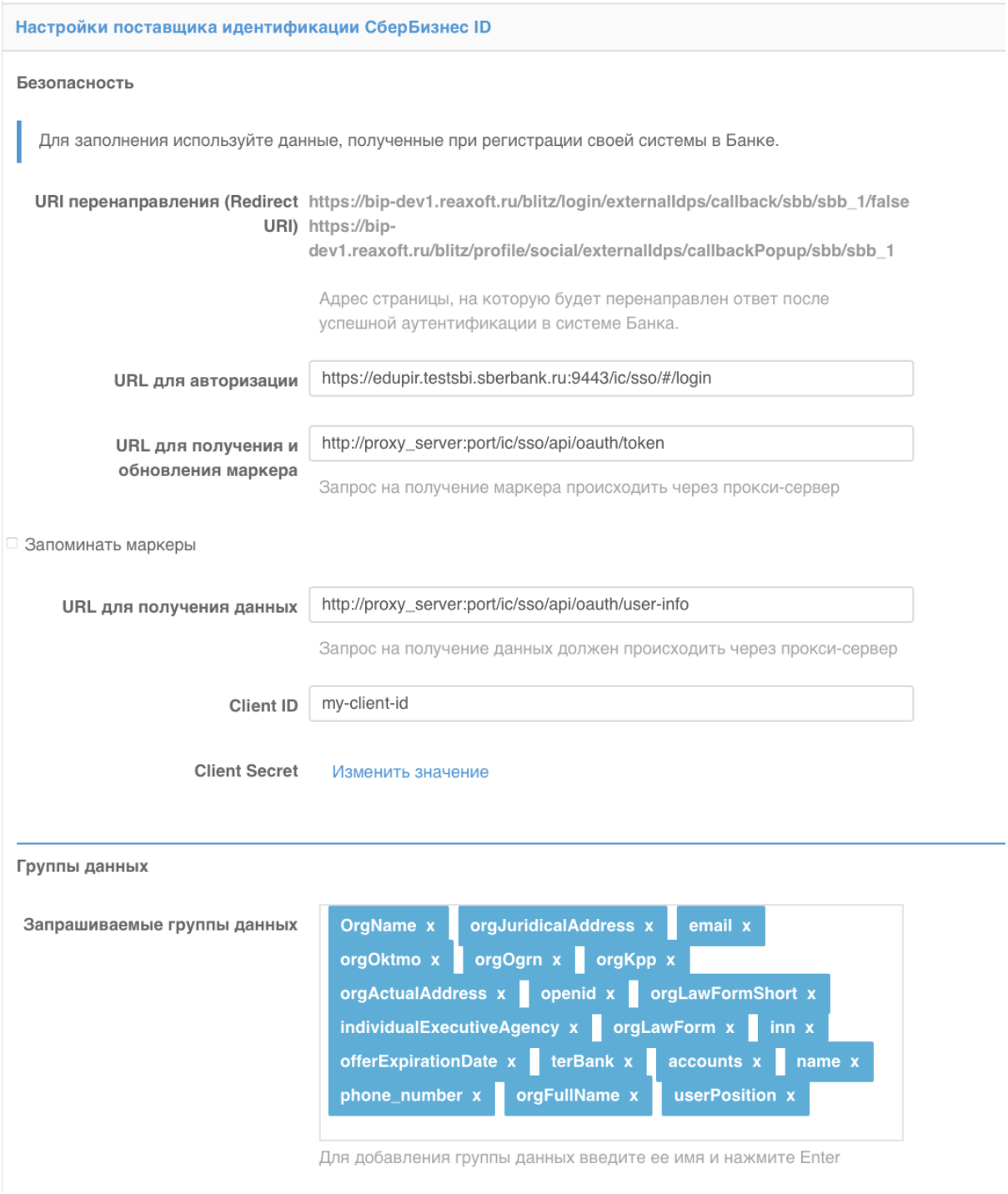
Для конфигурирования входа через СберБизнес ID необходимо выполнить следующие шаги:

1. Зарегистрировать приложение в системе СберБизнес ID[46]. Для этого подать заявку на официальном сайте этого поставщика идентификации. По результатам регистрации необходимо получить:
   - идентификатор клиента (`Client ID`);
   - секрет клиента (`Client Secret`).
2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип `СберБизнес ID`.
3. Заполнить настройки поставщика идентификации:
   - Идентификатор поставщика;
   - Название поставщика;
   - URL для авторизации – адрес, по которому должна инициироваться аутентификация, например: `https://sbi.sberbank.ru:9443/ic/sso/#/login`;
   - URL для получения и обновления маркера – адрес, по которому происходит получение и обновление маркера доступа, например: `http://proxy_server:port/ic/sso/api/oauth/token`. Подключение к сервису получения данных должно осуществляться через прокси-сервер, на котором должны быть настроены TLS-сертификаты для защищенного взаимодействия с сервисом СберБизнес ID;

     Примечание: Если маркеры доступа пользователей необходимо сохранять в базу данных, установите флажок `Запоминать маркеры`. В результате маркеры будут сохраняться в следующих случаях:
     - при входе пользователя;
     - при привязке внешнего поставщика в Личном кабинете;
     - при привязке внешнего поставщика через REST API v2;
     - при регистрации пользователя через внешнего поставщика.

   - URL для получения данных – адрес, по которому происходит получение данных пользователя, например: `http://proxy_server:port/ic/sso/api/oauth/user-info`. Подключение к сервису получения данных должно осуществляться через прокси-сервер, на котором должны быть настроены TLS-сертификаты для защищенного взаимодействия с сервисом СберБизнес ID;
   - Идентификатор клиента (`Client ID`);
   - Секрет клиента (`Client Secret`);
   - Запрашиваемые группы данных – перечень запрашиваемых групп данных из СберБизнес ID;
4. Настроить правила связывания.
5. В разделе Аутентификация консоли управления включить использование метода аутентификации с использованием поставщика идентификации СберБизнес ID.

[46] https://api.developer.sber.ru/how-to-use/sberbusiness_id



Альфа ID

Для конфигурирования входа через Альфа ID необходимо выполнить следующие настройки:

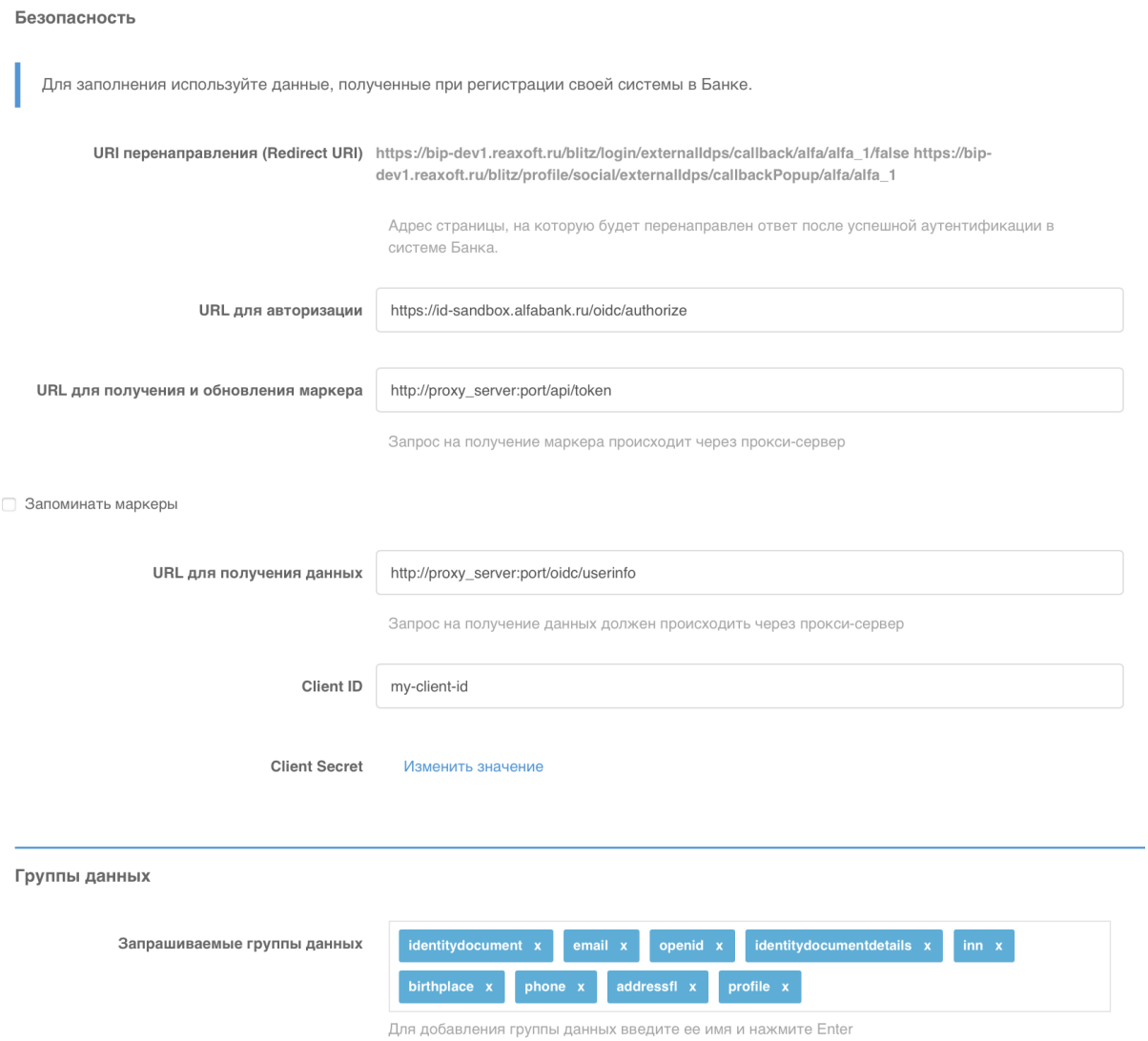
1. Зарегистрировать приложение в системе Альфа ID[47]. Для этого подать заявку на официальном сайте этого поставщика идентификации. По результатам регистрации необходимо получить:
   - идентификатор клиента (`Client ID`);
   - секрет клиента (`Client Secret`).

[47] https://alfabank.ru/sme/alfaid/

2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип `Альфа ID`.
3. Заполнить настройки поставщика идентификации:
   - Идентификатор поставщика;
   - Название поставщика;
   - URL для авторизации – адрес, по которому должна инициироваться аутентификация, например: `https://id-sandbox.alfabank.ru/oidc/authorize`;
   - URL для получения и обновления маркера – адрес, по которому происходит получение и обновление маркера доступа, например: `http://proxy_server:port/api/token`. Подключение к сервису получения данных должно осуществляться через прокси-сервер, на котором должны быть настроены TLS-сертификаты для защищенного взаимодействия с сервисом Альфа ID;

     Примечание: Если маркеры доступа пользователей необходимо сохранять в базу данных, установите флажок `Запоминать маркеры`. В результате маркеры будут сохраняться в следующих случаях:
     - при входе пользователя;
     - при привязке внешнего поставщика в Личном кабинете;
     - при привязке внешнего поставщика через REST API v2;
     - при регистрации пользователя через внешнего поставщика.

   - URL для получения данных – адрес, по которому происходит получение данных пользователя, например: `http://proxy_server:port/oidc/userinfo`. Подключение к сервису получения данных должно осуществляться через прокси-сервер, на котором должны быть настроены TLS-сертификаты для защищенного взаимодействия с сервисом Альфа ID;
   - Идентификатор клиента (`Client ID`);
   - Секрет клиента (`Client Secret`);
   - Запрашиваемые группы данных – перечень запрашиваемых групп данных из Альфа ID.
4. Настроить правила связывания.
5. В разделе Аутентификация консоли управления включить использование метода аутентификации с использованием поставщика идентификации Альфа ID.



Mos ID (СУДИР)

Для конфигурирования входа через Mos ID (СУДИР) необходимо выполнить следующие шаги:

1. Зарегистрировать приложение в системе СУДИР. Для этого подать заявку согласно инструкции, размещенной на официальном сайте этого поставщика идентификации - `https://login.mos.ru/support` (внешний СУДИР) и `https://sudir.mos.ru/support` (внутренний СУДИР). По результатам регистрации необходимо получить:
    - идентификатор (`client_id`);
    - секрет клиента (`client_secret`).
2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип `СУДИР`.
3. Заполнить настройки поставщика идентификации:
    - Идентификатор поставщика;
    - Название поставщика;
    - URL для авторизации – адрес, по которому должна инициироваться аутентификация, например: `https://login.mos.ru/sps/oauth/ae` для внешнего контура СУДИР и `https://sudir.mos.ru/blitz/oauth/ae` для внутреннего;

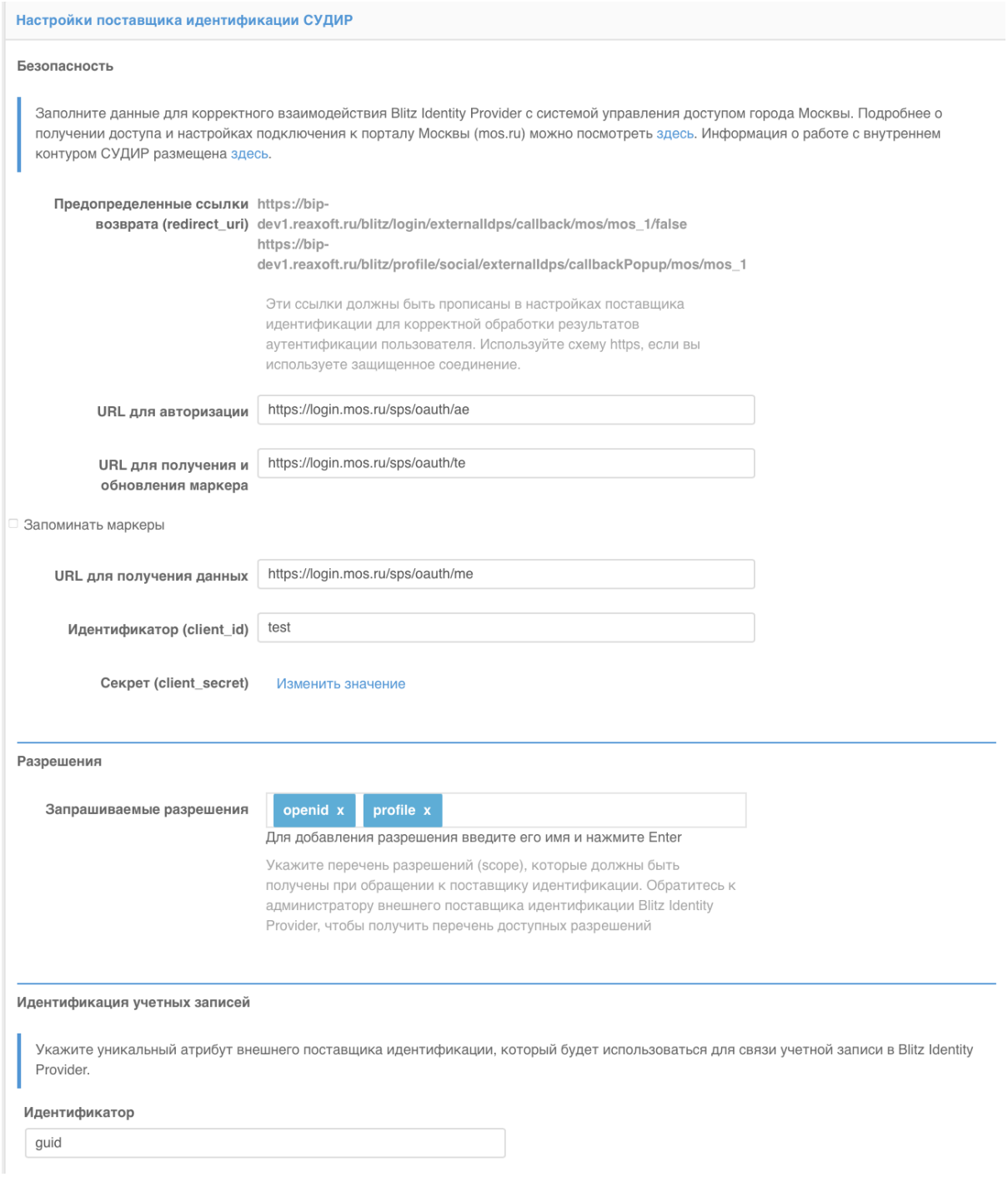
- URL для получения и обновления маркера – адрес, по которому происходит получение и обновление маркера доступа, например: `https://login.mos.ru/sps/oauth/te` для внешнего контура СУДИР и `https://sudir.mos.ru/blitz/oauth/te` для внутреннего;

  Примечание: Если маркеры доступа пользователей необходимо сохранять в базу данных, установите флажок `Запоминать маркеры`. В результате маркеры будут сохраняться в следующих случаях:

  - при входе пользователя;
  - при привязке внешнего поставщика в Личном кабинете;
  - при привязке внешнего поставщика через REST API v2;
  - при регистрации пользователя через внешнего поставщика.

- URL для получения данных – адрес, по которому происходит получение данных пользователя, например: `https://login.mos.ru/sps/oauth/me` для внешнего контура СУДИР и `https://sudir.mos.ru/blitz/oauth/me` для внутреннего;
- Идентификатор (`client_id`);
- Секрет клиента (`client_secret`).
- Запрашиваемые группы данных – перечень запрашиваемых разрешений, например, `openid` и `profile`;

4. Настроить правила связывания.
5. В разделе Аутентификация консоли управления включить использование метода аутентификации с использованием поставщика идентификации СУДИР.



Международные поставщики

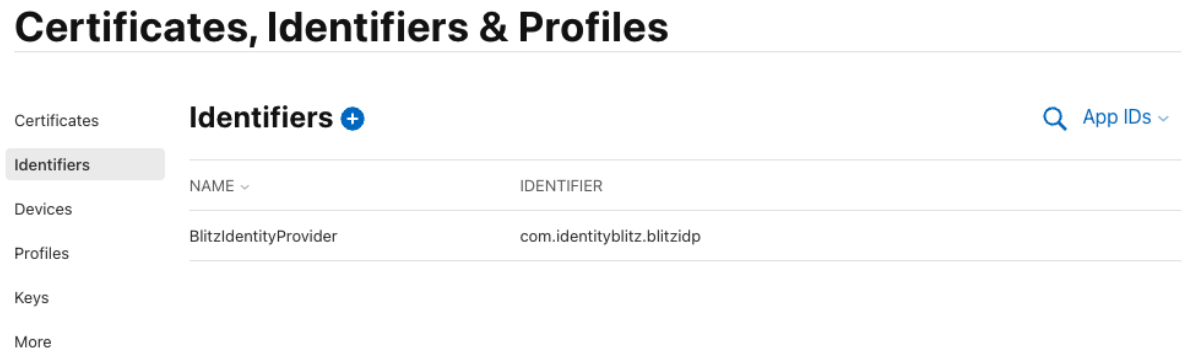
Apple ID

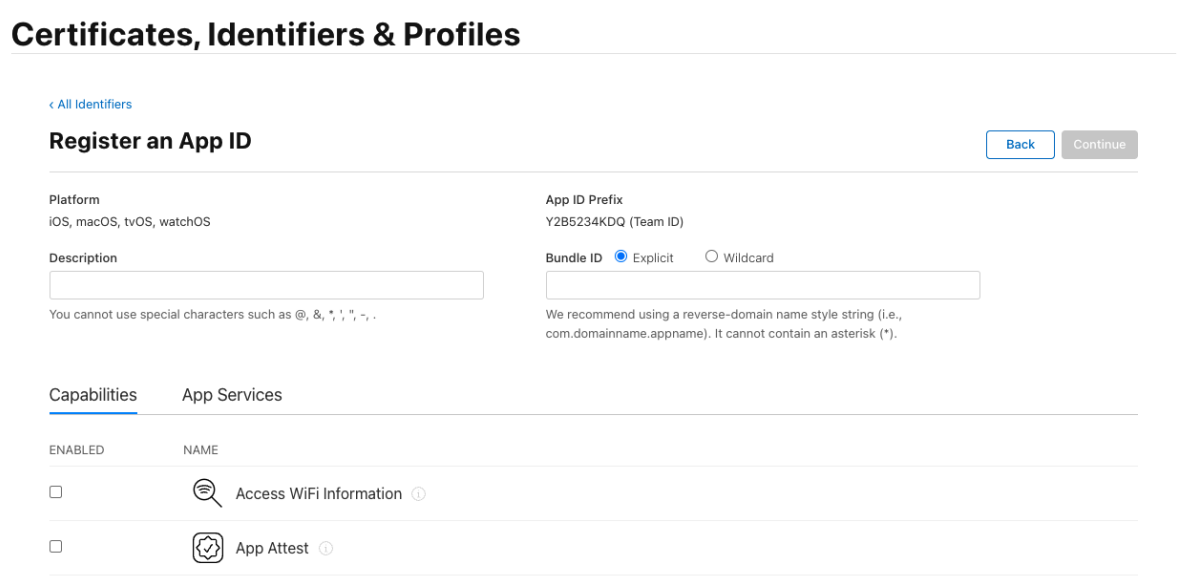
Для конфигурирования входа через Apple ID необходимо перейти в «Apple Developer Account» (обратите внимание, у компании должна быть действующая подписка Apple Developer ID) в раздел «Certificates, Identifiers & Profiles»[48], в котором выполнить следующие операции:

1. В окне *«Certificates, Identifiers & Profiles»* выбрать в правом верхнем углу фильтр *«App IDs»*. Кнопкой *«+»* рядом с *«Identifiers»* создать новый *«App ID»*:

[48] https://developer.apple.com/account/resources/identifiers/list

- выбрать тип `App`;
- задать *«Description»*. Он будет отображаться пользователю в окне подтверждения входа по Apple ID;
- в *«Bundle ID»* задать идентификатор вида `com.company.login` на основе используемого в Blitz Identity Provider домена;
- в *«Capabilities»* отметить *«Sign In with Apple»*, нажать рядом кнопку `Edit` и проверить, что выбрано *«Enable as a primary App ID»*;
- будет предложено завершить настройку – это все описано в последующих пунктах. Пока нужно нажать *«Register»*.





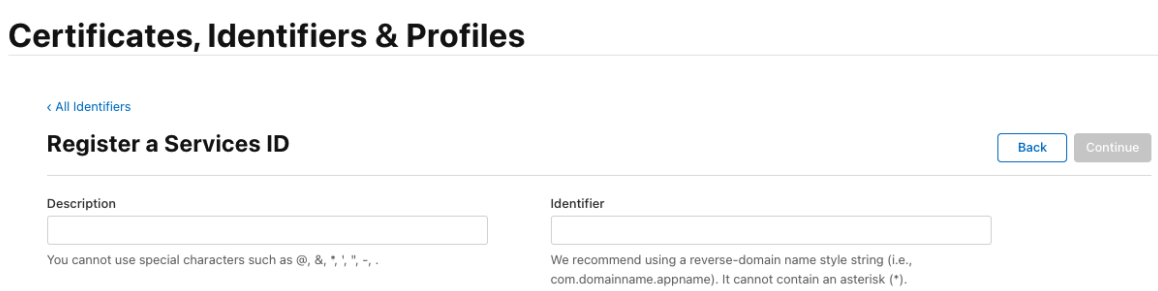
2. В окне *«Certificates, Identifiers & Profiles»* выбрать в правом верхнем углу фильтр *«Services App IDs»*. Кнопкой *«+»* рядом с *«Identifiers»* создать новый *«Services App ID»*:
   - задать *«Description»*. Он будет отображаться пользователю в окне подтверждения входа по Apple ID;
   - задать *«Identifier»*. Рекомендуется задать в виде `com.company.login` на основе используемого в Blitz Identity Provider домена. Позже созданный `Identifier` нужно будет ввести в настройках Blitz Identity Provider в качестве `client_id` в настройку *«Идентификатор клиента (Service ID)»*;
   - нажать *«Register»*;
   - выбрать созданный *«Service ID»*. В его настройках поставить чекбокс *«ENABLED»* и нажать *«Configure»*;

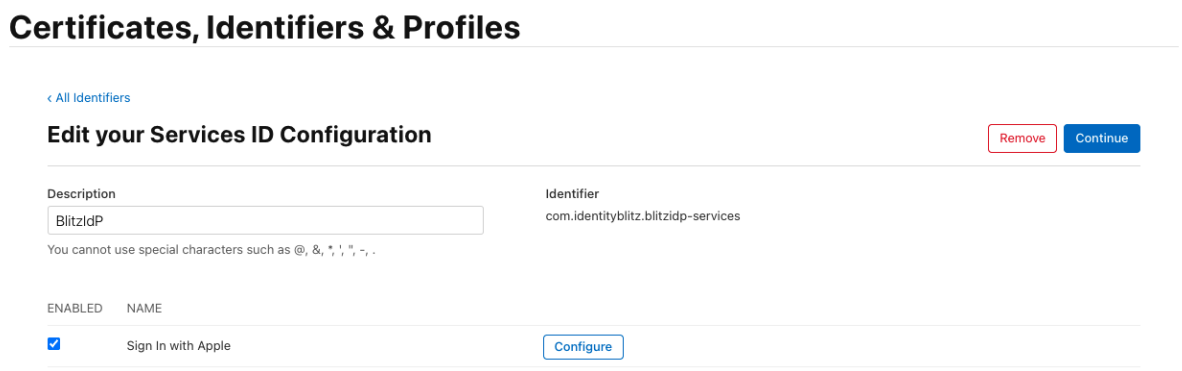
- в открывшемся окне проверить, что в *«Primary App ID»* указывается созданный ранее *«App ID»*;
- в *«Domains and Subdomains»* через запятую перечислить домены, используемые Blitz Identity Provider;
- в *«Return URLs»* перечислить `URL` возврата через запятую и с указанием `https`. Нужно указать `URL`, образцы которых Blitz Identity Provider показывает в настройках подключения Apple ID, например:

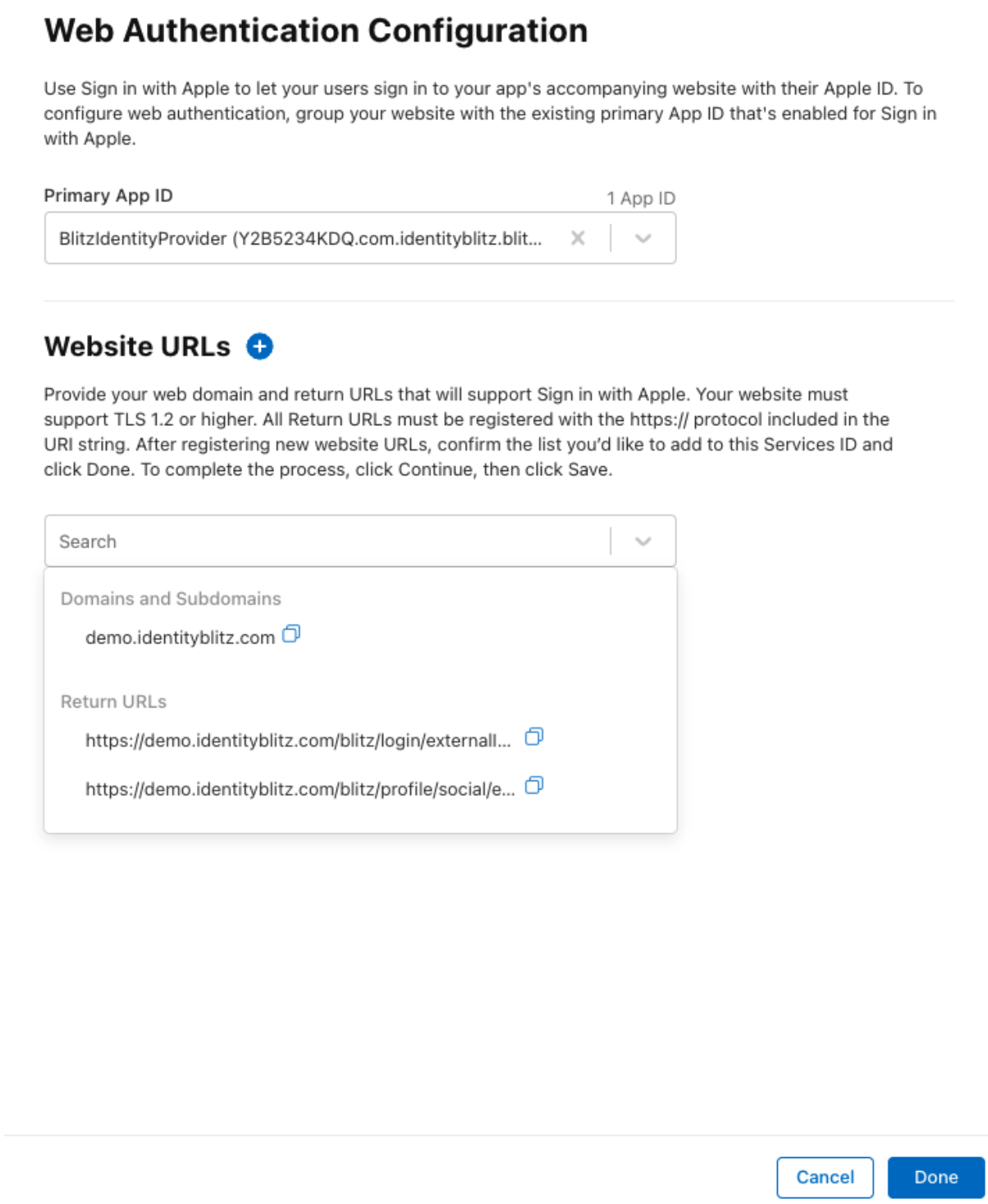
```
https://login.company.com/blitz/login/externalIdps/callback/apple/apple_1/false
https://login.company.com/blitz/profile/social/externalIdps/callbackPopup/
↪apple/apple_1
```

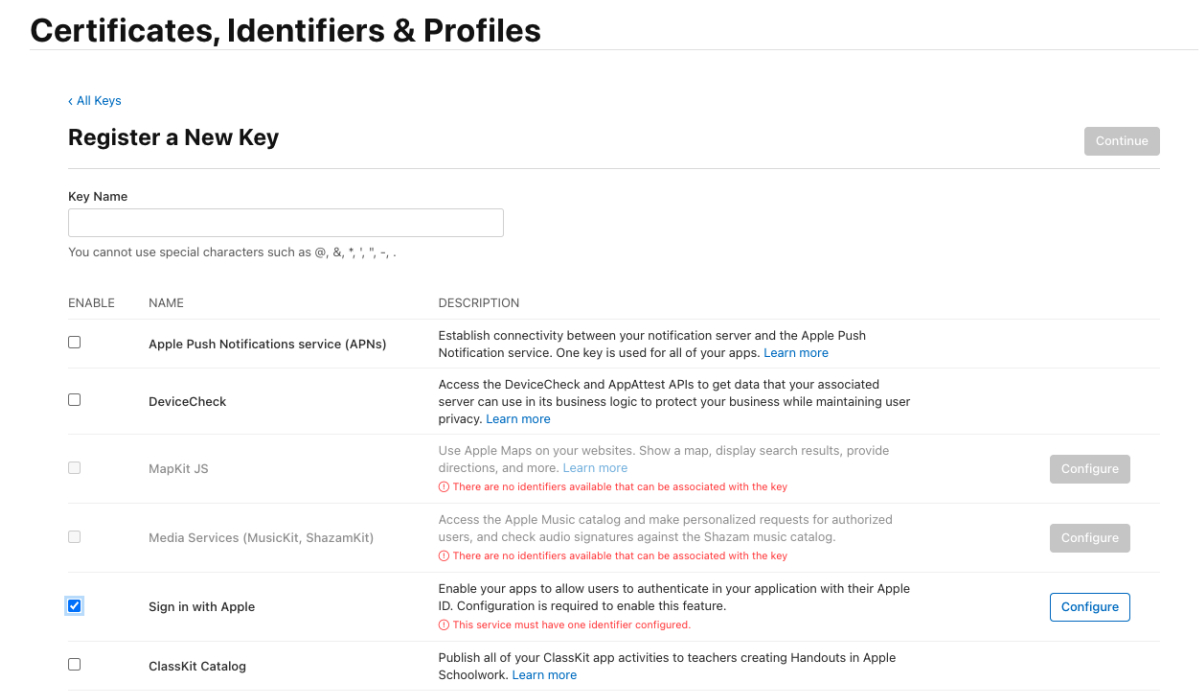
- подтвердить задание настроек, нажав последовательно *«Confirm»*, *«Done»*, *«Continue»*, *«Save»* в последующих экранах;

3. В меню *«Keys»* создать ключ для *«Sign In with Apple»*. Это можно сделать только однократно, так что рекомендуется созданный ключ где-то сохранить. В Blitz Identity Provider в настоящий момент этот ключ не используется и не понадобится, но он должен быть создан и сохранен на будущее.



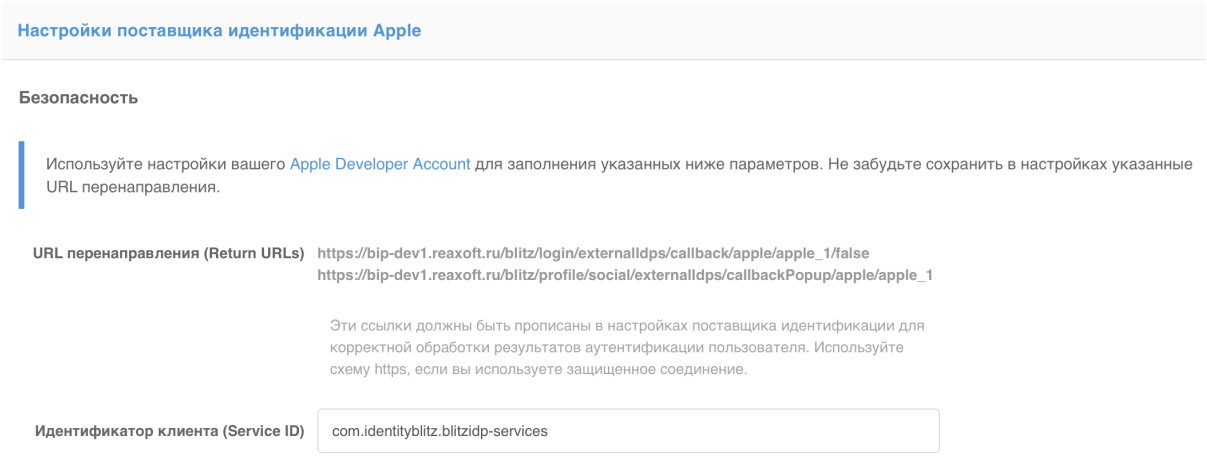



Web Authentication Configuration
Use Sign in with Apple to let your users sign in to your app's accompanying website with their Apple ID. To configure web authentication, group your website with the existing primary App ID that's enabled for Sign in with Apple.
Primary App ID
1 App ID
BlitzIdentityProvider (Y2B5234KDQ.com.identityblitz.blit...
Website URLs
Provide your web domain and return URLs that will support Sign in with Apple. Your website must support TLS 1.2 or higher. All Return URLs must be registered with the https:// protocol included in the URI string. After registering new website URLs, confirm the list you'd like to add to this Services ID and click Done. To complete the process, click Continue, then click Save.
Search
Domains and Subdomains
demo.identityblitz.com
Return URLs
https://demo.identityblitz.com/blitz/login/externall...
https://demo.identityblitz.com/blitz/profile/social/e...
Cancel
Done



После окончания настроек в «Apple Developer Account» необходимо:

1. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип `Apple`.
2. Заполнить настройки поставщика идентификации:
   - Идентификатор поставщика;
   - Название поставщика;
   - Идентификатор клиента (`Service ID`), полученный в Apple Developer Account.
3. Настроить правила связывания.
4. В разделе *«Аутентификация»* консоли управления включить использование метода аутентификации с использованием поставщика идентификации Apple.

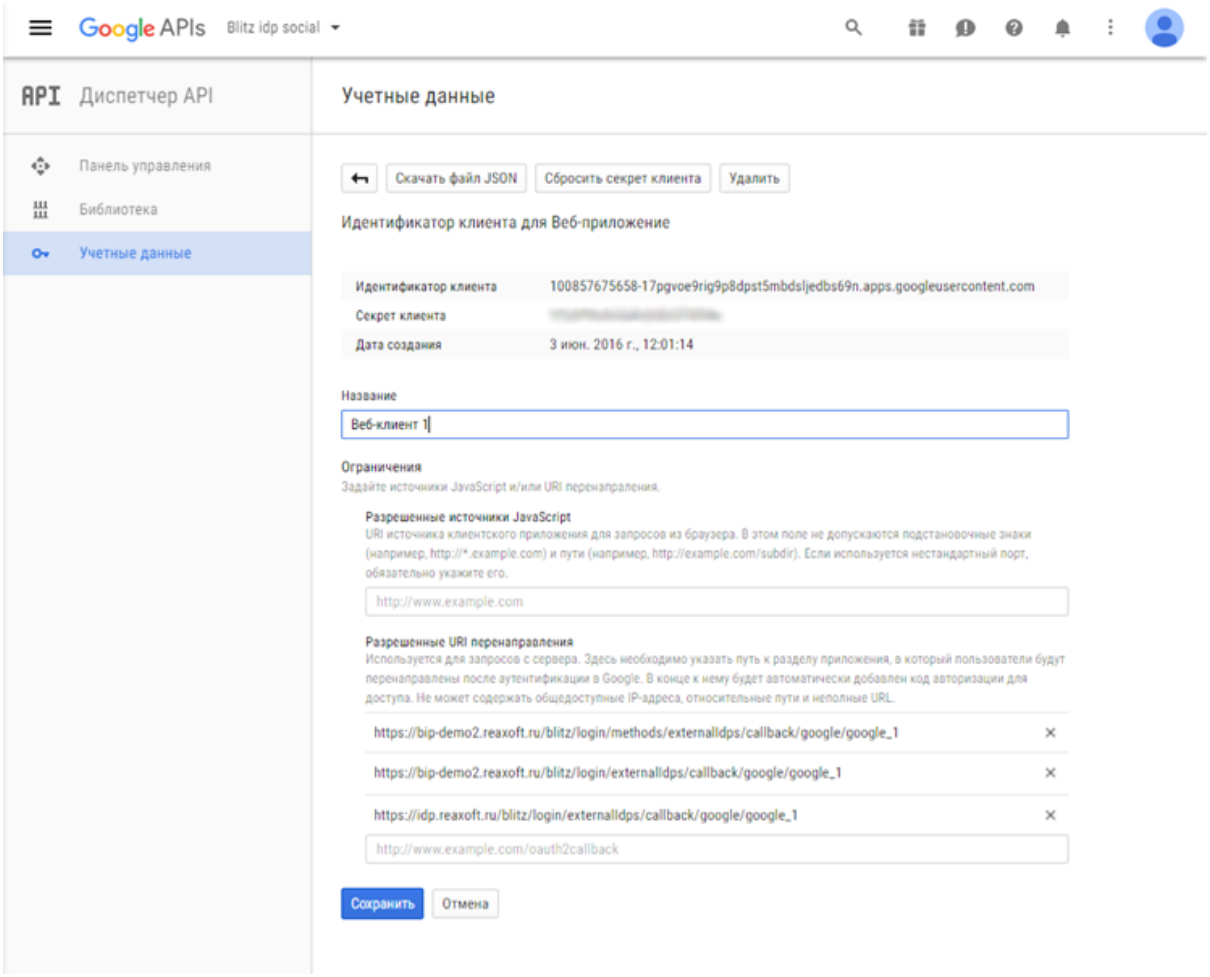


Google

Для конфигурирования входа через Google необходимо выполнить следующие шаги:

1. Перейти в Диспетчер API Google[49], в котором выполнить следующие операции:
   - перейти в раздел *«Учетные данные»*;
   - создать проект и создать новые учетные данные типа *«Идентификатор клиента OAuth»*;
   - тип нового идентификатора клиента (например, веб-приложение) и дать ему название;
   - ограничения не задавать, они будут указаны позже;
   - Google сгенерирует идентификатор и секрет клиента, они потребуются для последующего ввода в консоли управления Blitz Identity Provider.
   - в *«Разрешенные URI перенаправления»* перечислить `URL` возврата через запятую и с указанием `https`. Нужно указать `URL`, образцы которых Blitz Identity Provider показывает в настройках подключения Google, например:

```
https://login.company.com/blitz/login/externalIdps/callback/google/google_1/
↪false
https://login.company.com/blitz/profile/social/externalIdps/callbackPopup/
↪google/google_1
```



2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип `Google`.
3. Заполнить настройки поставщика идентификации:

[49] https://console.developers.google.com

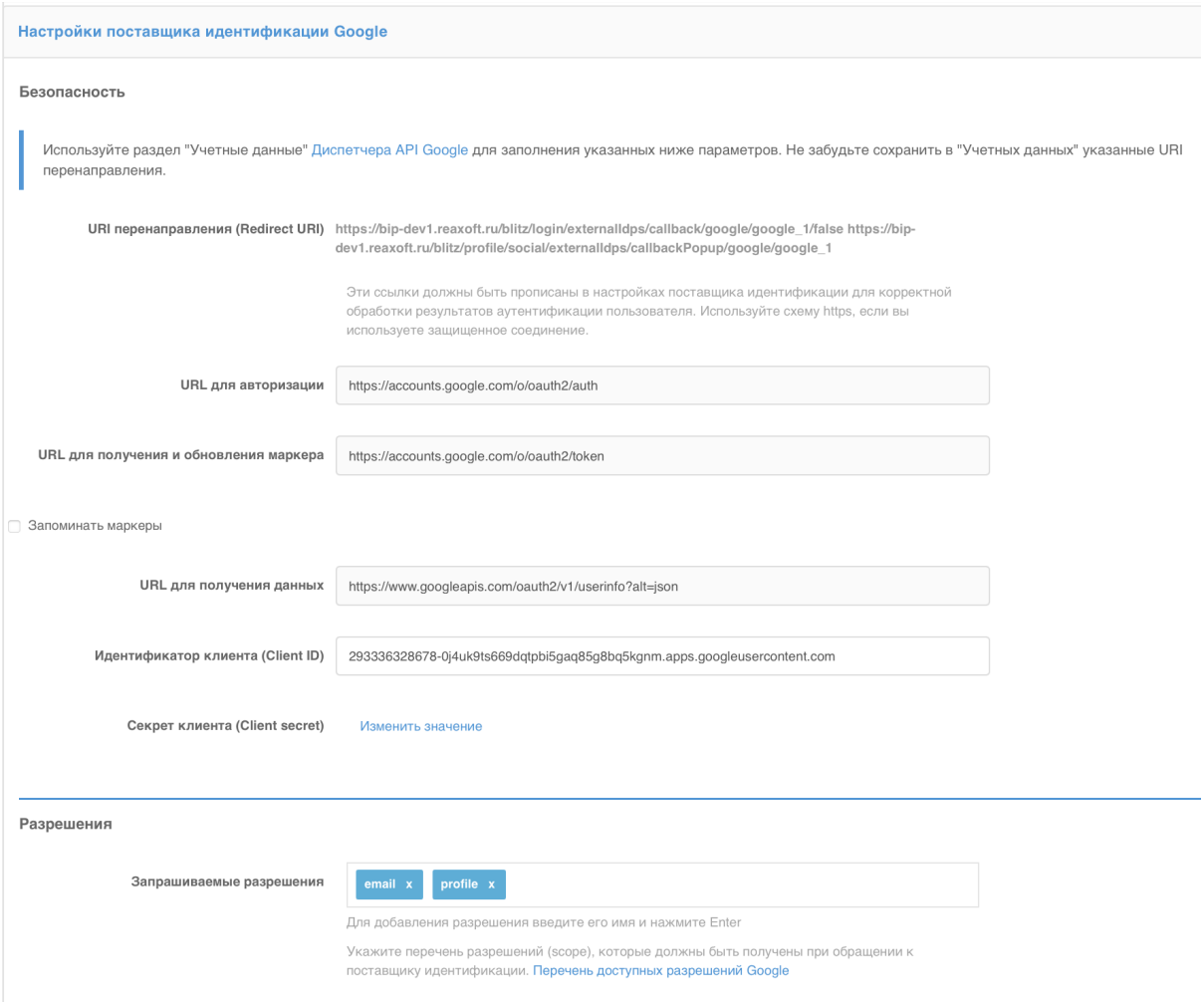
- Идентификатор поставщика;
- Название поставщика;
- Идентификатор клиента (`Client ID`), полученный в Диспетчере API Google;
- Секрет клиента (`Client secret`), полученный в Диспетчере API Google;
- URL для авторизации;
- URL для получения и обновления маркера;

> **Примечание:** Если маркеры доступа пользователей необходимо сохранять в базу данных, установите флажок `Запоминать маркеры`. В результате маркеры будут сохраняться в следующих случаях:
>
> – при входе пользователя;
>
> – при привязке внешнего поставщика в Личном кабинете;
>
> – при привязке внешнего поставщика через REST API v2;
>
> – при регистрации пользователя через внешнего поставщика.

- URL для получения данных;
- Запрашиваемые разрешения (`scope`), предусмотренные в Google[50].

4. Настроить правила связывания.
5. В разделе *«Аутентификация»* консоли управления включить использование метода аутентификации с использованием поставщика идентификации Google.

[50] https://developers.google.com/+/web/api/rest/oauth#authorization-scopes



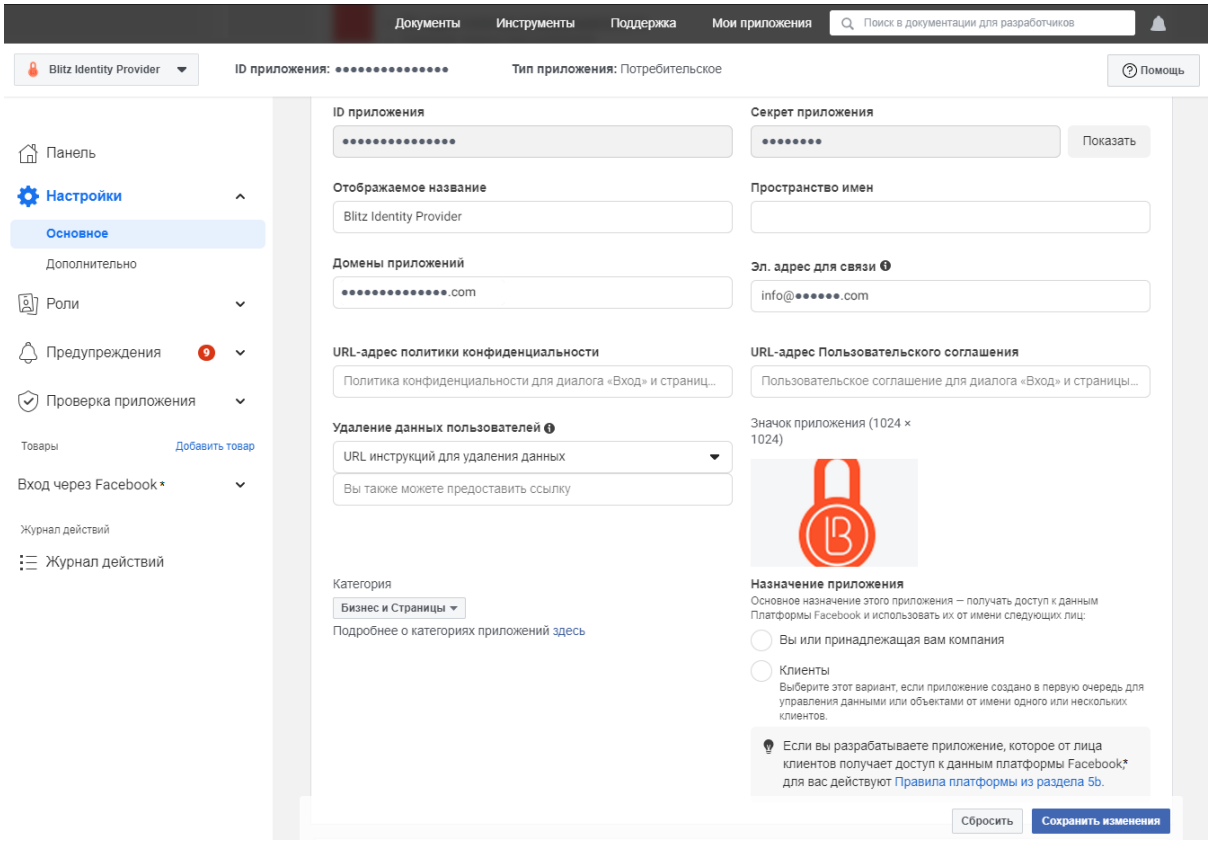
Facebook[1]

Для конфигурирования входа через Facebook необходимо выполнить следующие шаги:

1. Перейти в панель Facebook для разработчиков[51], в которой выполнить следующие настройки:
   - добавить новое приложение, указав его название, адрес электронной почты для связи и категорию приложения;
   - создать идентификатор приложения;
   - перейти в настройки приложения, раздел *«Основное»*. В этом разделе указать параметр *«Домены приложения»* (параметр должен соответствовать домену, на котором установлен Blitz Identity Provider) и добавить сайт с аналогичным URL.
   - перейти в раздел *«Проверка приложения»* и активировать пункт *«Сделать приложение «…» доступным для всех»*.

[1] Meta признана экстремистской организацией и запрещена в России, деятельность принадлежащих ей соц.сетей Facebook и Instagram также запрещена в РФ.

[51] https://developers.facebook.com/apps/



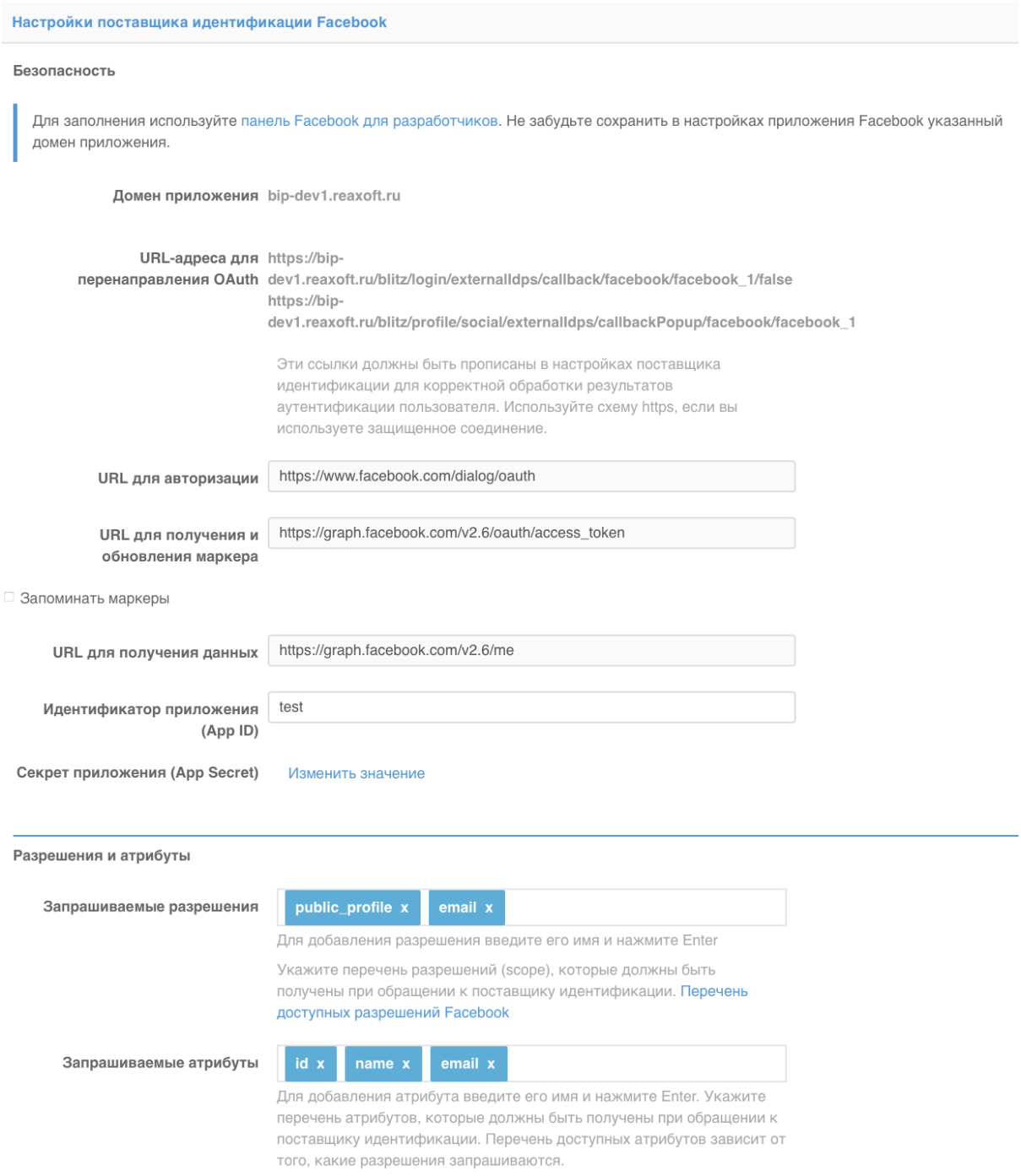
2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип `Facebook`.
3. Заполнить настройки поставщика идентификации:
   - Идентификатор поставщика;
   - Название поставщика;
   - Идентификатор приложения (`App ID`), полученный в панели Facebook для разработчиков;
   - Секрет приложения (`App Secret`), полученный в панели Facebook для разработчиков;
   - URL для авторизации;
   - URL для получения и обновления маркера;

     Примечание: Если маркеры доступа пользователей необходимо сохранять в базу данных, установите флажок `Запоминать маркеры`. В результате маркеры будут сохраняться в следующих случаях:
     - при входе пользователя;
     - при привязке внешнего поставщика в Личном кабинете;
     - при привязке внешнего поставщика через REST API v2;
     - при регистрации пользователя через внешнего поставщика.

   - URL для получения данных;
   - Запрашиваемые разрешения (`scope`), предусмотренные в Facebook[52];
   - Запрашиваемые атрибуты, предусмотренные в Facebook; допустимо указывать только те атрибуты, которые предусмотрены выбранными разрешениями.

[52] https://developers.facebook.com/docs/facebook-login/permissions/

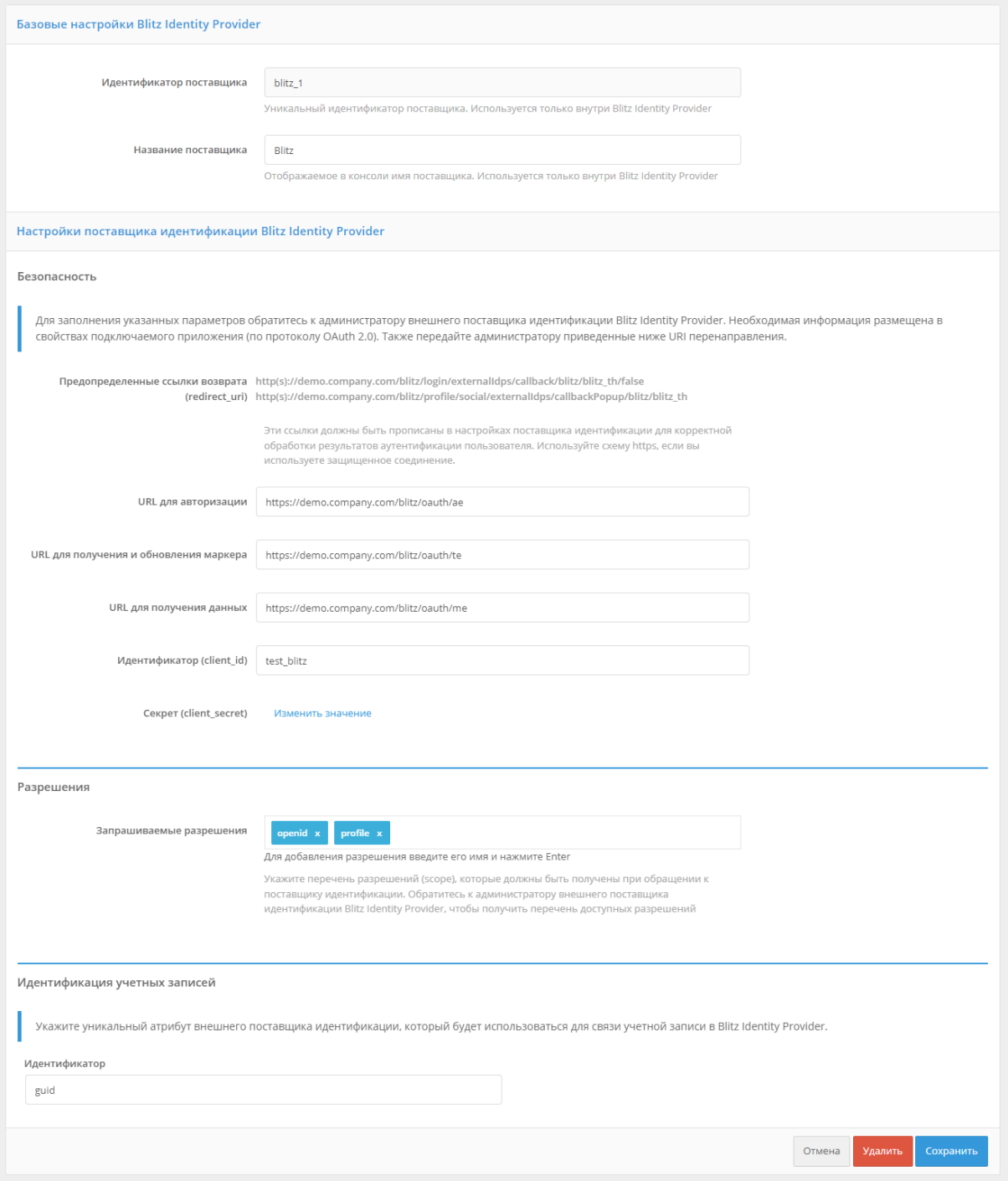
4. Настроить правила связывания.
5. В разделе *«Аутентификация»* консоли управления включить использование метода аутентификации с использованием поставщика идентификации Facebook.



Вход через другую установку Blitz Identity Provider

Для конфигурирования входа через учетную запись другого Blitz Identity Provider (например, установленного в другой организации, далее – доверенный Blitz Identity Provider) или иного поставщика идентификации, поддерживающего OIDC, необходимо выполнить следующие шаги:

1. Открыть консоль управления доверенного Blitz Identity Provider (или попросить администратора другого Blitz Identity Provider это сделать) и выполнить следующие операции:
   - перейти в раздел *«Приложения»*;
   - нажать на кнопку *«Добавить приложение»*;
   - указать идентификатор приложения, название и домен приложения;
   - сохранить приложение и перейти к его настройке;
   - выбрать протокол подключения OAuth 2.0;
   - указать секрет (`client_secret`), либо оставить предзаполненный вариант;
   - указать префикс ссылки возврата, в качестве которой указать URL основной Blitz Identity Provider, в который будет осуществляться вход;
   - произвести настройку необходимых разрешений в разделе *«OAuth 2.0»*.
2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип Blitz Identity Provider.
3. Заполнить настройки поставщика идентификации:
   - Идентификатор поставщика;
   - Название поставщика;
   - URI внешнего поставщика – домен, на котором установлен доверенный Blitz Identity Provider;
   - Идентификатор (`client_id`), указанный в настройках доверенного Blitz Identity Provider;
   - Секрет (`client_secret`), указанный в настройках доверенного Blitz Identity Provider;
   - Запрашиваемые разрешения, данные разрешения должны быть определены в разделе OAuth 2.0 доверенного Blitz Identity Provider;
   - Идентификатор – атрибут доверенного Blitz Identity Provider, который будет использоваться в качестве идентификатора пользователя (обеспечивает уникальность учетной записи даже при изменении атрибута, отвечающего за имя пользователя);
4. Настроить правила связывания.
5. В разделе *«Аутентификация»* консоли управления включить использование метода аутентификации с использованием поставщика идентификации Blitz Identity Provider.



Настройки связывания учетных записей

В настройках каждого поставщика идентификации предусмотрен раздел Связывание учетных записей. С помощью настроек данного раздела можно определить:

- правила связывания внешней учетной записи с учетной записью в Blitz Identity Provider;
- правила соответствия атрибутов внешней учетной записи и учетной записи в Blitz Identity Provider.

Предусмотрены два режима настройки: базовая и расширенная.

Связывание внешней учетной записи с учетной записью в Blitz Identity Provider происходит в следующих сценариях:

- При первом входе с использованием внешней учетной записи, если она еще не привязана ни к одной учетной записи в Blitz Identity Provider.
- При связывании в Личном кабинете.

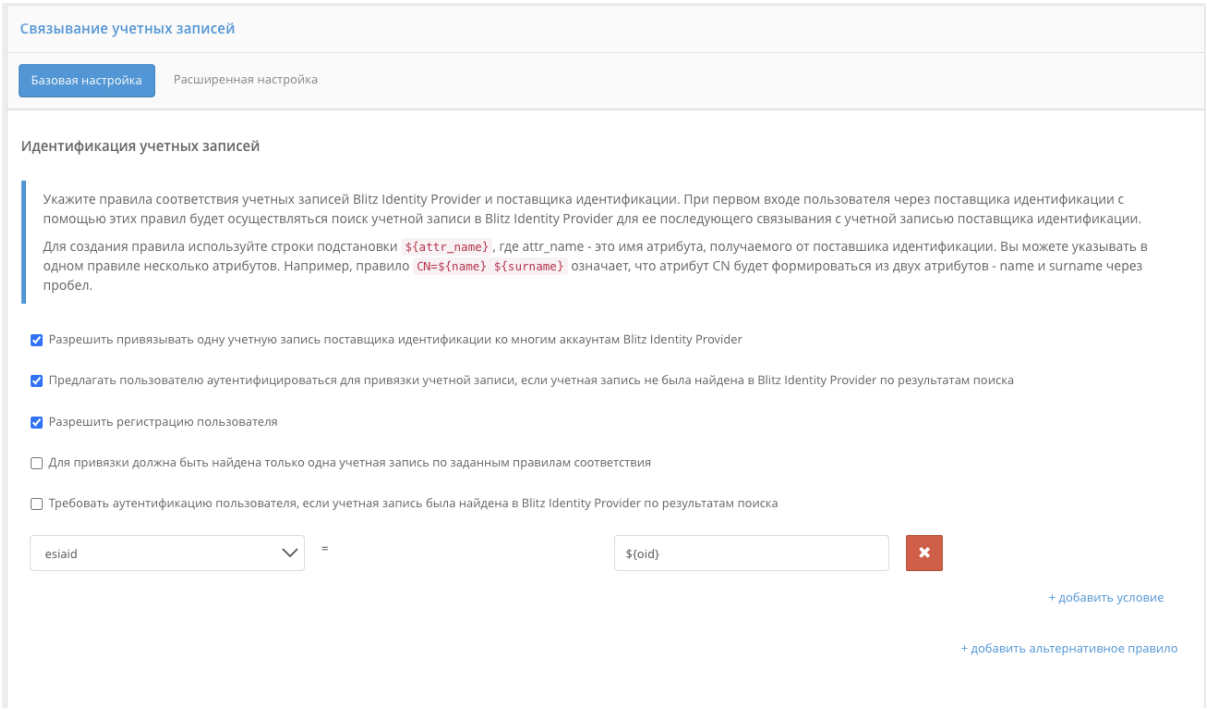
Базовая настройка

Базовая настройка выполняется с помощью конструктора правил. Данный режим подходит для типовых сценариев связывания учетных записей и сопоставления атрибутов.

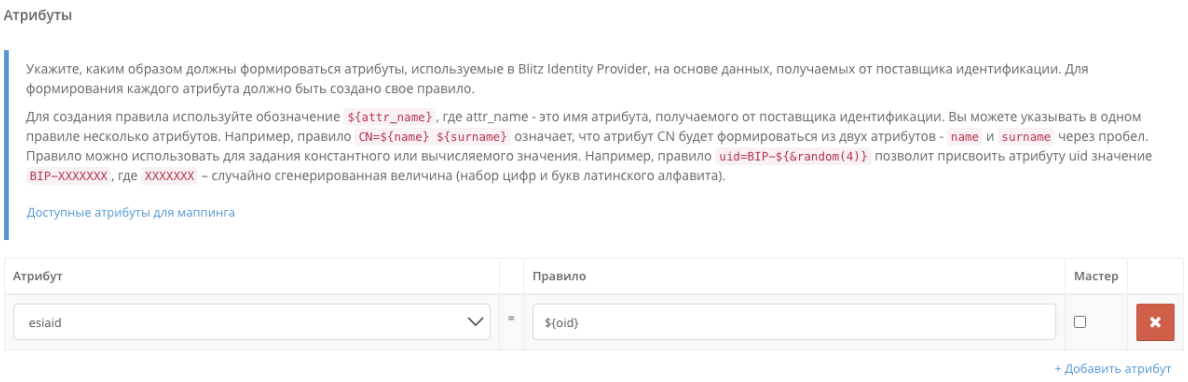
Предусмотрены следующие настройки:

- `Разрешить привязывать одну учетную запись поставщика идентификации ко многим аккаунтам`:
  - `опция выбрана` – Blitz Identity Provider разрешит привязать внешнюю учетную запись к нескольким учетным записям в Blitz Identity Provider. При входе пользователя такой внешней учетной записью в процессе входа ему будет показан выбор из нескольких привязанных учетных записей.
  - `опция не выбрана` – Blitz Identity Provider не позволит привязать внешнюю учетную запись к учетной записи Blitz Identity Provider, если такая внешняя учетная запись уже привязана к другой учетной записи Blitz Identity Provider.
- `Предлагать пользователю ввести логин и пароль для привязки, если учетная запись не была идентифицирована`:
  - `опция выбрана` – пользователю, будет предложено пройти идентификацию и аутентификацию альтернативным способом, чтобы привязать внешнюю учетную запись, если по настроенным правилам не удалось найти учетную запись в Blitz Identity Provider.
  - `опция не выбрана` – Blitz Identity Provider не разрешит вход пользователя, для которого не удалось сопоставить учетные записи. Если настроен процесс регистрация для внешних учетных записей, то будет автоматически запущен процесс регистрации.
- `Разрешить регистрацию пользователя`:
  - `опция выбрана` – в форме ввода пароля отображается ссылка, по которой можно зарегистрироваться во внешнем поставщике.
  - `опция не выбрана` – в форме ввода пароля нет возможности перейти к регистрации во внешнем поставщике.
- `Для привязки должна быть найдена только одна учетная запись по заданным правилам соответствия`:
  - `опция выбрана` – если по правилам соответствия найдено более одной учетной записи, то пользователю будет выведено сообщение об ошибке.
  - `опция не выбрана` – если по правилам соответствия найдено более одной учетной записи, то будет возможность продолжить процесс привязки.
- `Требовать ввод пароля, если учетная запись была идентифицирована`:
  - `опция выбрана` – пользователю нужно будет пройти аутентификацию для привязки его учетной записи к аккаунту внешнего поставщика.
  - `опция не выбрана` – учетная запись будет автоматически привязана к аккаунту внешнего поставщика.
- Настройка правил идентификации учетных записей – можно создать правила соответствия идентификационных атрибутов из внешней учетной записи идентификационным атрибутам в Blitz Identity Provider. Для создания правил идентификации нужно использовать строки подстановки `${attr_name}`, где `attr_name` – это имя атрибута, получаемого от внешнего поставщика идентификации. Можно указывать в одном правиле несколько атрибутов. Например, правило `email=${default_email-}` означает, что атрибут `email` в Blitz Identity Provider будет со-

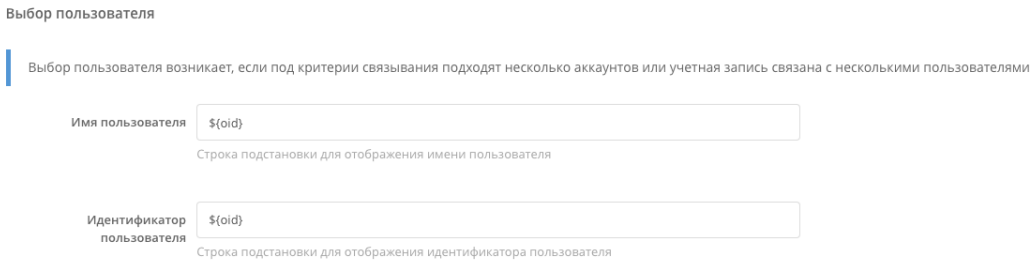
поставляться с атрибутом `default_email` внешней учетной записи при условии, что атрибут `default_email` не пустой. Можно указать несколько условий (с помощью ссылки + добавить условие, которые должны выполняться одновременно и можно добавлять альтернативные правила с помощью ссылки + добавить альтернативное правило).



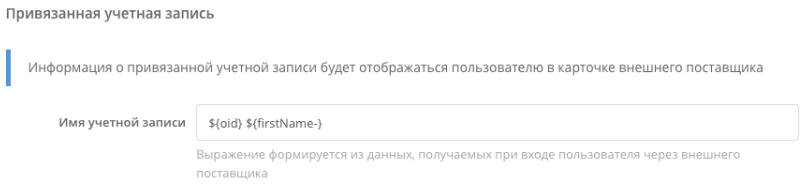
- Блок Атрибуты с правилами сохранения атрибутов. Например, правило `email=${default_email}` означает, что атрибут с именем `email` в Blitz Identity Provider будет заполняться значением из атрибута `default_email` внешней учетной записи (для пользователей, воспользовавшихся этим поставщиком идентификации). Если у атрибута отмечен чекбокс `Мастер`, то заполнение или обновление атрибута будет происходить при каждом входе через внешний поставщик идентификации. Если чекбокс `Мастер` не отмечен, то заполнение произойдет только при первом входе, в результате которого возникло связывание учетных записей.



- Блок Выбор пользователя определяет правила отображения пользователю найденной по настроенным правилам соответствия учетной записи в Blitz Identity Provider. Настройка `Имя пользователя` определяет информацию, отображаемую в верхней строке карточки пользователя (строке, предназначенной для отображения имени учетной записи). Например, `${family_name- } ${given_name-}` определяет, что пользователю в верхней строке будут показаны фамилия и имя (если они заполнены). Настройка `Идентификатор пользователя` определяет информацию, отображаемую в нижней строке карточки пользователя (строке, предназначенной для отображения идентификатора учетной записи). При настройке можно использовать маскирование значений. Например, правило `${phone_number&maskInMiddle(3,3)}` будет отображать средние числа номера телефона в виде `*`.



- Блок Привязанная учетная запись определяет правило отображения привязанной учетной записи пользователю в карточке внешнего поставщика в консоли и Личном кабинете. Выражение формируется из данных, получаемых при входе пользователя через внешнего поставщика.



Расширенная настройка

В случае расширенной настройки правила связывания учетных записей и правила соответствия атрибутов задаются с помощью процедуры связывания на языке программирования Java. Данный режим обеспечивает максимальную гибкость настройки и подходит для узкоспециализированных сценариев связывания учетных записей и сопоставления атрибутов.

См.также:

Процедуры привязки аккаунтов внешних поставщиков (страница 286)

2.2.4 Пользовательские сервисы

Blitz Identity Provider предоставляет веб-приложения, с помощью которых пользователи самостоятельно могут выполнять ряд операций:

1. Веб-приложение **Личный кабинет**. Позволяет выполнить ряд операций с учетной записью, например, посмотреть/изменить свои данные, настроить способы аутентификации, посмотреть последние события, сменить пароль. Если включен, то доступен по адресу: `https://{hostname}/blitz/profile`.
2. Веб-приложение **Регистрация пользователей**. При включении становится доступен переход со страницы входа на форму самостоятельной регистрации (ссылка Нет аккаунта? Зарегистрироваться).
3. Веб-приложение **Восстановление доступа**. Позволяет пользователю сменить пароль от своей учетной записи после прохождения проверок. Если приложение включено, то пользователи смогут перейти со страницы входа (ссылка Забыли пароль?) на форму восстановления доступа.

Настройка данных сервисов осуществляется в разделе Сервисы самообслуживания консоли управления.

Внимание: Администратор консоли управления должен самостоятельно проверять корректность помещаемых на страницу входа JS-скриптов и содержимое страниц регистрации и личного кабинета на предмет возможных уязвимостей.

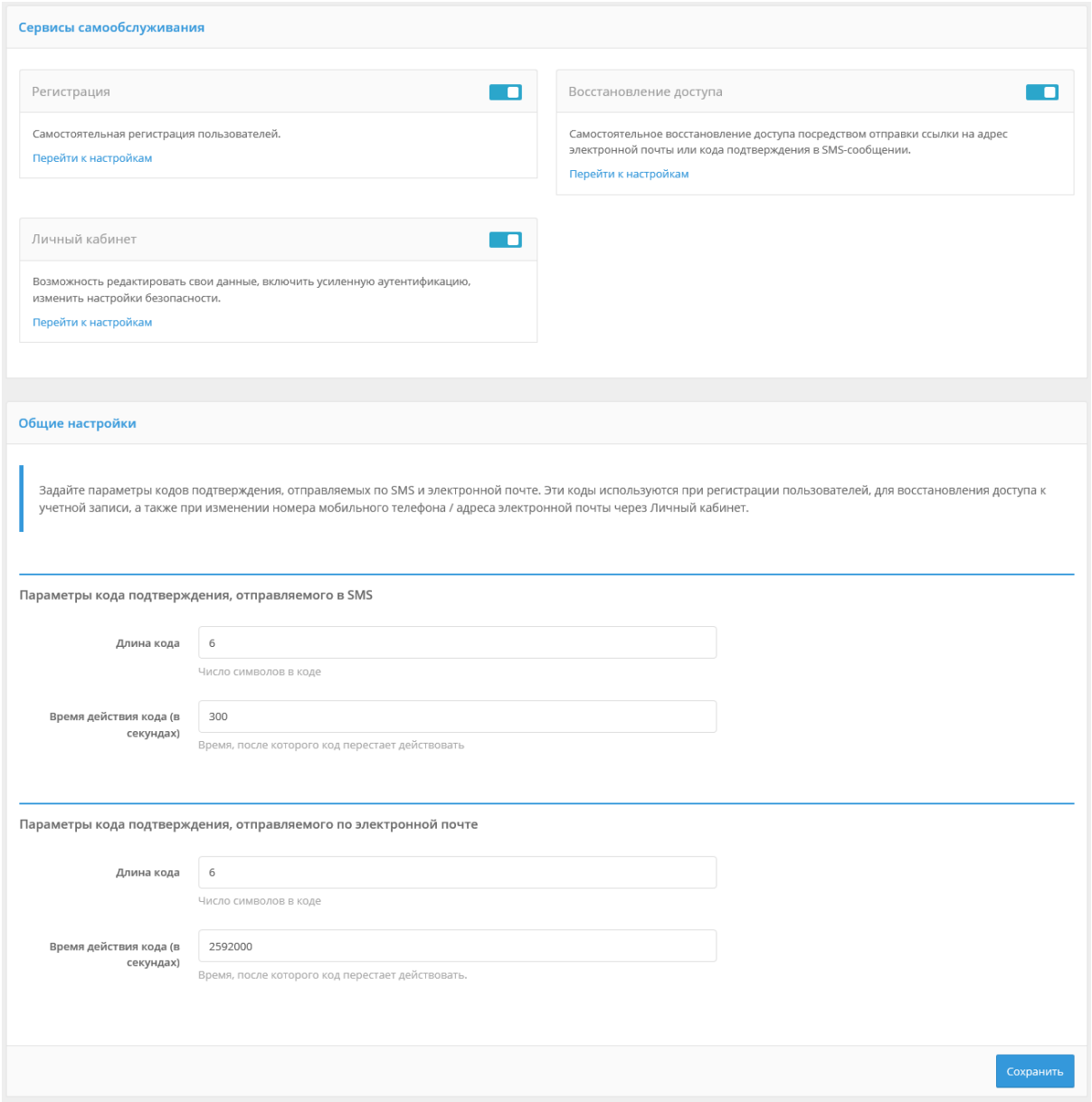
Общие настройки

На главной странице раздела Сервисы самообслуживания можно включить или выключить соответствующие приложения (сервисы), используя переключатель (). Следует при этом учесть, что переключатель лишь влияет на отображение ссылок (например, Забыли пароль?), тогда как наличие самого сервиса зависит от того, было ли соответствующее приложение установлено администратором:

- `blitz-idp` – веб-приложение **Личный кабинет**,
- `blitz-registration` – веб-приложение **Регистрация пользователей**,
- `blitz-recovery` – веб-приложение **Восстановление доступа**.

Также на главной странице можно настроить параметры, применяемые во всех сервисах самообслуживания:

- параметры кода подтверждения, отправляемого в SMS – можно изменить длину кода и время его действия, а также количество попыток;
- параметры кода подтверждения, отправляемого по электронной почте – можно изменить длину кода и время его действия.



В подразделах осуществляется настройка каждого сервиса самообслуживания в отдельности.

Регистрация пользователей

`Регистрация пользователей` – веб-приложение, позволяющее пользователю самостоятельно создать свою учетную запись. Настройка регистрации включает в себя конфигурирование формы регистрации, изменение параметров сервиса и создание процедуры регистрации (опционально).

Форма регистрации

Перечень запрашиваемых данных пользователя определяется HTML-шаблоном. Шаблон представляет собой текстовый файл, который компилируется с помощью шаблонизатора Twirl[53]. В шаблоне необходимо разместить функции, позволяющие пользователю при регистрации вводить данные о себе.

Примеры функций, доступных в шаблоне:

- `@attrInput("email", msg("reg.email"), Map("placeholder" -> "mail@example.com", "error-messages" -> msg("reg.email.wrong"), "input-type" -> "mail"))` – отображает на странице поле для ввода атрибута `email`, описанного в системе. `msg("reg.email")` - это название атрибута, которое берется из файла сообщений в соответствии с текущей локалью. При пустом поле ввода в нем отображается `"mail@example.com"` в качестве подсказки, а при некорректном вводе – сообщение `msg("reg.email.wrong")` из файла сообщений. Для элемента задается `input-type` равный `mail`;
- `@attrInput("family_name", "Фамилия", Map("placeholder" -> "Фамилия", "error-messages" -> "Ошибка"))` – отображает на странице поле для ввода фамилии пользователя в переменную `family_name`. Эту переменную далее можно использовать при исполнении процедуры регистрации.
- `@securityQuestionInput` – отображает на странице поля ввода контрольного вопроса и ответа на контрольный вопрос;
- `@passwordsInput` - отображает на странице поля ввода пароля и его подтверждение;
- `@agreement` - отображает ссылку на условия использования;
- `@attrExpr` – функция, позволяющая создать вычисляемый атрибут (или присвоить атрибуту константное значение);
- `@submitButton` - отображает кнопку Зарегистрироваться.

Пример шаблона для регистрации:

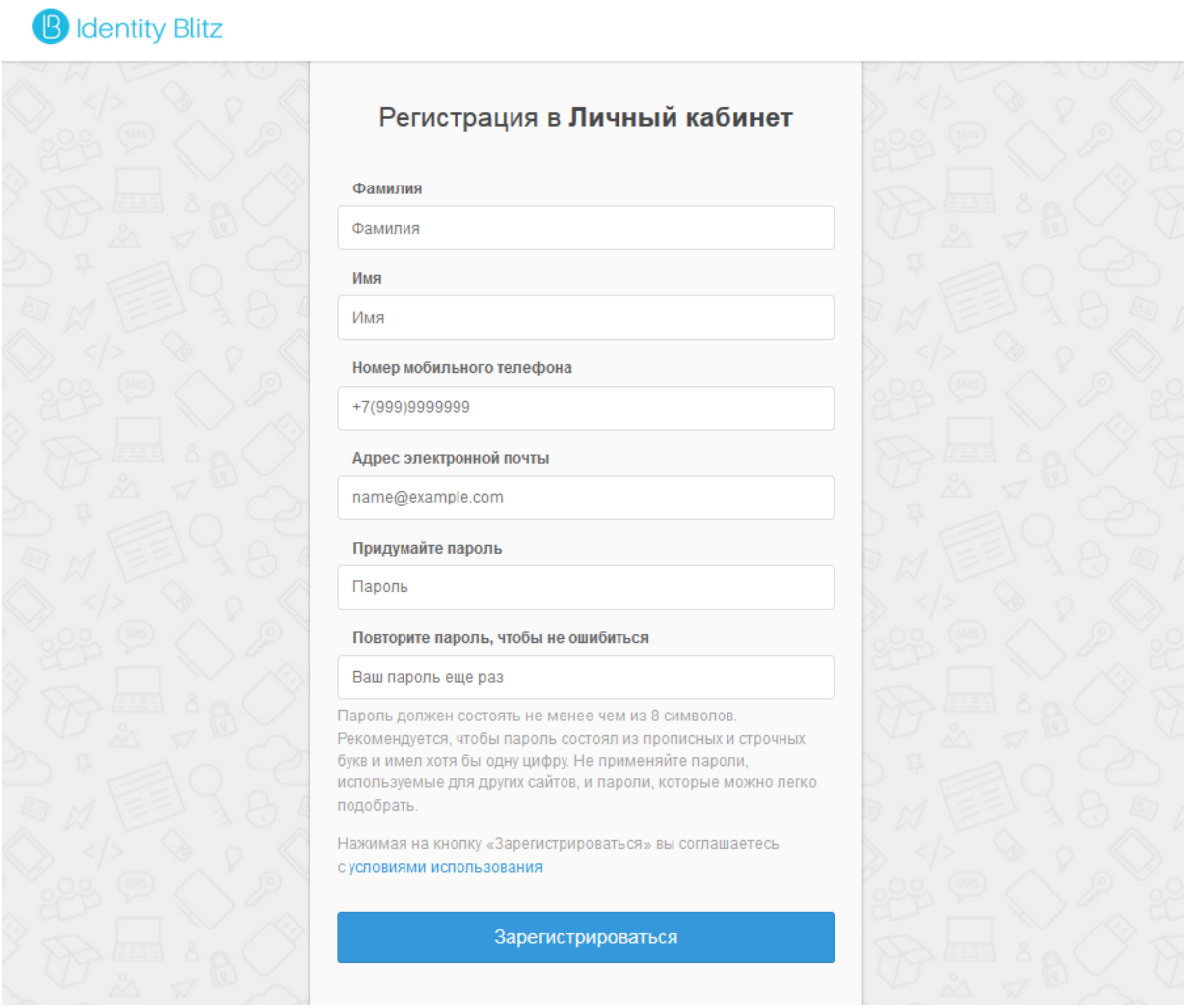
```
@attrInput("family_name", "Фамилия", Map("placeholder" -> "Фамилия", "error-
↪messages" -> "Ошибка"))
@attrInput("given_name", "Имя", Map("placeholder" -> "Имя", "error-messages" ->
↪"Ошибка"))
@attrInput("phone_number", "Номер мобильного телефона", Map("placeholder" ->
↪"+7(999)9999999", "error-messages" -> "reg.page.mobile.req.err.msg"))
@attrInput("email", "Адрес электронной почты", Map("placeholder" -> "name@example.
↪com", "error-messages" -> "reg.page.email.req.err.msg", "input-type" -> "mail"))
@passwordsInput
@agreement
@attrExpr("sub","BIP-${&random(4)}")
@submitButton
```

Совет: Для автогенерации GUID создаваемых учетных записей используйте следующую формулу `@attrExpr`:

```
@attrExpr("sub","${&rUUID()}")
```

Результат использования указанного шаблона в интерфейсе веб-приложения **Регистрация пользователя** представлен на рисунке:

[53] https://github.com/playframework/twirl



Для добавления на форму регистрации выпадающего списка для выбора значений атрибута из справочника необходимо:

1. Создать на сервере Blitz Identity Provider директорию `/etc/blitz-config/custom_messages/dics`;
2. Создать файл `/etc/blitz-config/custom_messages/dics/dic_name` с содержимым справочника (вместо `dic_name` указать имя справочника, например, `company_id`). Пример файла `company_id` для выпадающего справочника выбора компании:

```
001=Тестовая компания 1
002=Тестовая компания 2
003=Тестовая компания 3
```

Число в справочнике будет записываться в значение атрибута. Строка в справочнике будет показываться пользователю на форме регистрации.

3. Проверить владельца директории `dics` и файлов справочников в ней. Владелец должен быть `blitz:blitz`.

```
chown -R blitz:blitz /etc/blitz-config/custom_messages/dics
```

4. В конфигурационном файле `blitz.conf` в блок `blitz.prod.local.idp.messages` добавить блок `dics`. В настройке `names` перечислить все имена справочников (для каждого справочника должен быть создан свой файл со значениями справочника). Например:

```
"dics" : {
    "dir" : "custom_messages/dics",
    "names" : [
        "company_id"
    ]
}
```

5. Перезапустить приложение `blitz-registration`.
6. В консоли управления в шаблоне страницы регистрации добавить строку с заполнением атрибута из справочника:

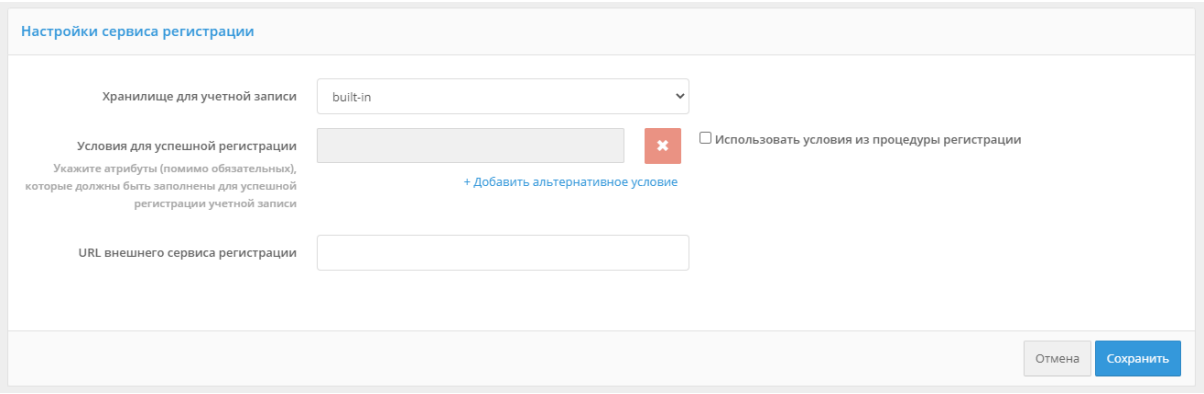
```
@attrInput("company", msg("Компания"), Map("dic" -> "company_id", "dic-default" ->
↪"0", "sort" -> "key"))
```

Настройки сервиса регистрации

В качестве настроек можно задать:

- хранилище для учетной записи – нужно выбрать одно из сконфигурированных хранилищ (раздел Источники данных) для сохранения учетной записи;
- необходимые для регистрации атрибуты пользователя – атрибуты, наличие которых необходимо для завершения процедуры регистрации. Обязательные атрибуты пользователя не нужно включать в данный список. Возможно добавление нескольких альтернативных правил. Если отмечен чекбокс `Использовать условия из процедуры регистрации`, то настроенные условия игнорируются и применяются условия, определенные функцией `isEnough` из процедуры регистрации.
- URL внешнего сервиса регистрации. Если задать в качестве параметра этот URL, то по этому адресу будет направлен пользователь при переходе к процессу регистрации (вместо приложения регистрации Blitz Identity Provider).

Скриншот фрагмента страницы настроек регистрации представлен на рисунке:



Процедура регистрации

Процедура регистрации – Java-код, реализующий необходимые проверки после того, как пользователь заполнит форму регистрации. В ходе исполнения процедуры возможно выполнение следующих действий:

- выполнение дополнительных проверок введенных данных;
- выполнение преобразования введенных данных;
- сохранение значений атрибутов в хранилище;
- вызов внешних REST-сервисов.

При необходимости преобразовать данные, введенные пользователем, и далее сохранить их в виде атрибутов, в шаблоне страницы регистрации следует использовать функцию `@attrInput` вместо `@textInput`.

Изменение текста условий использования

На странице регистрации пользователя размещена ссылка на условия использования. Условия использования размещены в архиве `assets.zip`, расположенном в директории `assets` установки Blitz Identity Provider в заархивированном каталоге `documents\user_agreement`.

Для изменения правил использования следует распаковать архив `assets.zip`, заменить файлы `user_agreement_ru.pdf` (русская версия) и `user_agreement_en.pdf` (английская версия) на требуемые и заархивировать архив с сохранением исходной структуры.

Также возможно изменить ссылку на правила использования. Для этого следует *отредактировать* (страница 301) строку `reg.page.reg.action.agreement` и `setPswd.page.agreement`. Такой способ рекомендуется применять, если правила использования размещены на внешнем ресурсе, например, в виде отдельной веб-страницы.

Личный кабинет

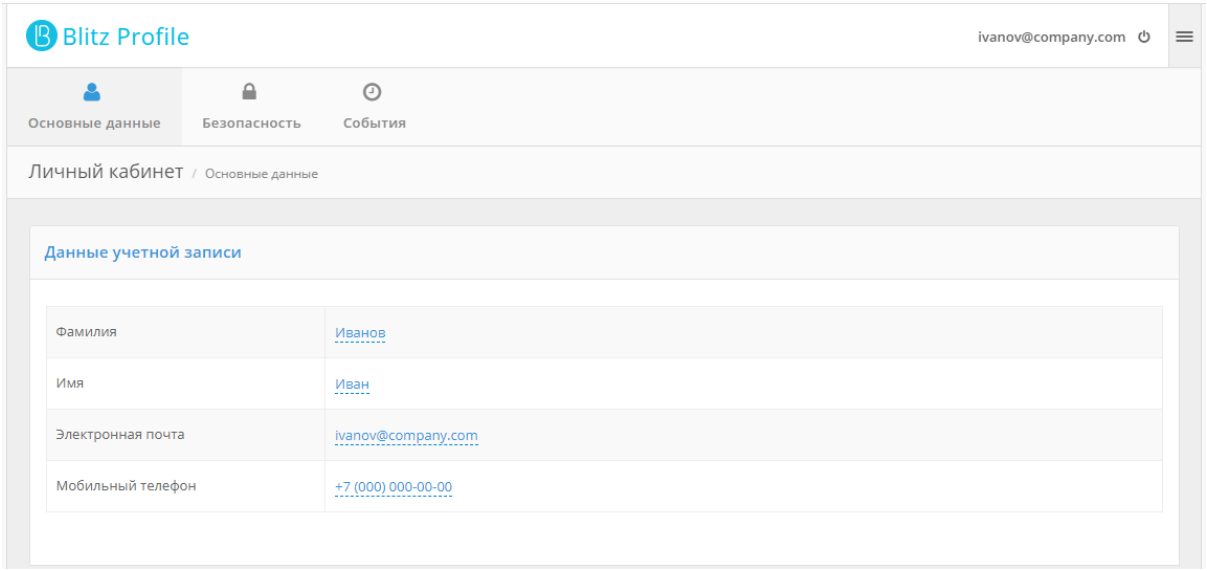
`Личный кабинет` – веб-приложение, в котором пользователь может выполнить следующие действия:

- посмотреть или изменить данные своей учетной записи;
- посмотреть последние события безопасности (например, события входа);
- сменить пароль;
- посмотреть и настроить способы подтверждения входа (двухфакторной аутентификации);
- посмотреть и настроить ключи безопасности;
- посмотреть привязанные учетные записи социальных сетей, привязать новые «внешние» учетные записи, отвязать лишние учетные записи;
- посмотреть привязанные устройства доступа, отвязать лишние устройства;
- посмотреть и отозвать выданные приложениями разрешения на доступ к данным;
- посмотреть события безопасности.

Настройка Личного кабинета включает в себя конфигурирование способа отображения атрибутов пользователя и изменение дополнительных параметров.

Отображение атрибутов пользователя

На основной странице Личного кабинета отображается блок с данными учетной записи. Пример этого блока представлен на рисунке ниже.



Отображение данных пользователя определяется HTML-шаблоном. Шаблон представляет собой текстовый файл, который компилируется с помощью шаблонизатора Twirl[54]. В шаблоне необходимо разместить функции, позволяющие пользователю в Личном кабинете вводить и редактировать данные о себе.

В шаблоне доступны следующие функции:

- `@show(attrName)` – отображает значение атрибута;
- `@showStrings(attrName, values)` – отображает значение массива;
- `@editAsText(attrName, readableName, errorMsg)` – отображает значение атрибута и позволяет его изменить (параметр `errorMsg` необязательный);
- `@editAsBoolean(attrName, readableName)` – отображает значение логического типа (`true/false`) атрибута и позволяет его изменить;
- `@editAsStrings(attrName, readableName, values)` – отображает значение (массив) атрибута и позволяет его изменить.

В этих функциях используются следующие параметры:

- `attrName` – название атрибута, определенное в разделе Источники данных;
- `readableName` – отображаемое в письме пользователю имя атрибута (можно задать как идентификатор из файла сообщений или как текст);
- `values` – значения, представляющие собой формат `ключ – описание`, где ключ – значение массива, описание – читаемое значение ключа (например, `ListMap("a" -> "значение a", "c" -> "значение c")`), может задаваться как идентификатор из файла сообщений или как текст;
- `errorMsg` – описание ошибки, которое отображается в случае ошибочного ввода значения (можно задать как идентификатор из файла сообщений или как текст). Про файлы сообщений см. *Текстовые сообщения веб-интерфейса* (страница 301). Рекомендуется использовать файлы сообщений при необходимости поддержки мультиязычности.

Примеры функций:

[54] https://github.com/playframework/twirl

Список 6: Отображение атрибута email

```
@editAsText("email", "Электронная почта")
```

Список 7: Отображение атрибута phone_number с возможностью его редактировать

```
@editAsText("phone_number", "Мобильный телефон", "Ошибка")
```

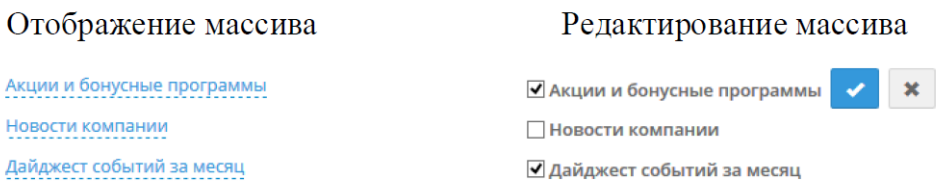
Список 8: Отображение булевого атрибута info с возможностью его редактировать

```
@editAsBoolean("info", "Подписка")
```

Список 9: Отображение массива строк massiv с возможностью его редактировать (выбор значений)

```
@editAsStrings("massiv", "Подписки", ListMap("a" -> "Акции и бонусные программы",
↪"b" -> "Новости компании", "c" -> "Дайджест событий за месяц"))
```

Пример отображения массива строк в интерфейсе веб-приложения **Личный кабинет** представлен на рисунке:

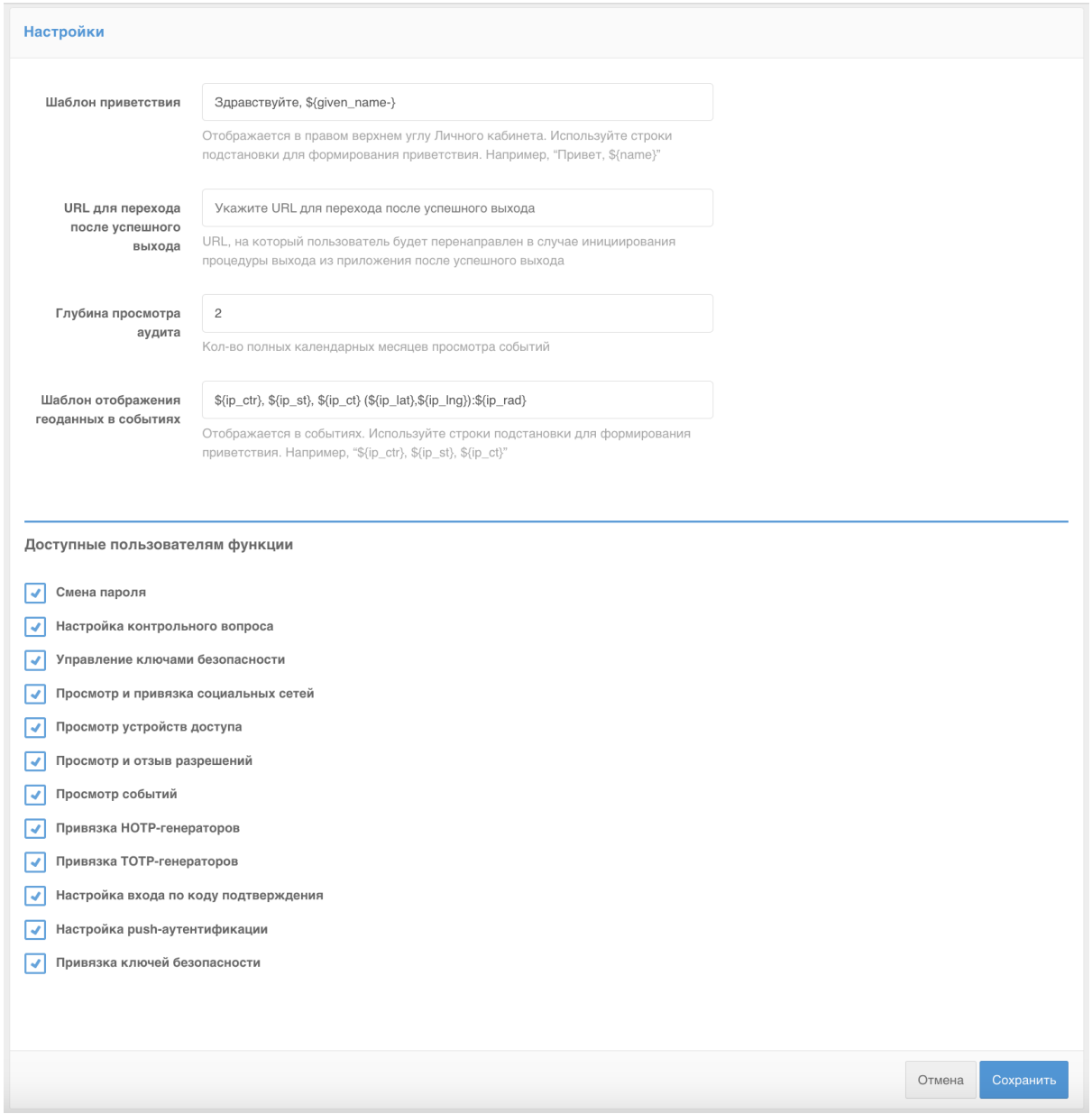


Дополнительные параметры

В качестве дополнительных параметров можно задать:

- шаблон приветствия – информацию, которая отображается в правом верхнем углу Личного кабинета. Допустимо использовать строки подстановки. Например, `${family_name} ${given_name}` позволит отобразить фамилию и имя пользователя;
- URL для перехода после успешного выхода из Личного кабинета;
- период отображаемых пользователям событий аудита (в календарных месяцах от текущей даты);
- шаблон отображения геоданных в событиях (см. *База геоданных* (страница 330)). Шаблон можно составить из следующих элементов, содержащих сведения о стране, регионе, городе и координатах: `${ip_ctr}, ${ip_st}, ${ip_ct}, ${ip_lng}, ${ip_lat}, ${ip_rad}`
- доступные пользователям функции, т.е. функции, которые могут быть задействованы пользователем из Личного кабинета. Возможно включить или выключить следующие функции:
  - смена пароля;
  - настройка контрольного вопроса;
  - управление ключами безопасности;
  - просмотр и привязка социальных сетей;
  - просмотр устройств доступа;
  - просмотр и отзыв разрешений;

- просмотр событий;
- привязка HOTP-генераторов;
- привязка TOTP-генераторов;
- настройка подтверждения входа по SMS-коду;
- настройка push-аутентификации;
- привязка ключей безопасности.



Восстановление доступа

Настройки в консоли

Настройка Возможные атрибуты для поиска сервиса восстановления доступа определяет атрибуты, по которым будет производиться поиск учетной записи.

С помощью настройки Атрибуты для сверки можно определить, значения каких атрибутов дополнительно должен ввести пользователь в процессе восстановления пароля для подтверждения владения учетной записью. Добавление такой проверки усложняет атаку на сброс пароля через множественный перебор в форме восстановления забытого пароля. На главной странице у пользователя будут запрошены атрибуты для сверки (например, фамилия) и восстановление будет выполнено только в том случае, если найденная учетная запись будет иметь идентичное значение атрибута.

Опция Проверять наличие пользователей, имеющих право менять пароль в найденной учетной записи определяет, что если у найденного пользователя имеется связанная («родительская») учетная запись, имеющая право менять пароль этому пользователю, то об этом будет выведено предупреждение при попытке восстановления пароля.

Настройка Возможные контакты восстановления доступа определяет атрибуты с контактами (адреса электронной почты и/или номера мобильного телефона), которые будут использованы для восстановления доступа. Атрибуты с контактами должны быть определены в разделе Источники данных в качестве адреса электронной почты и номера мобильного телефона.

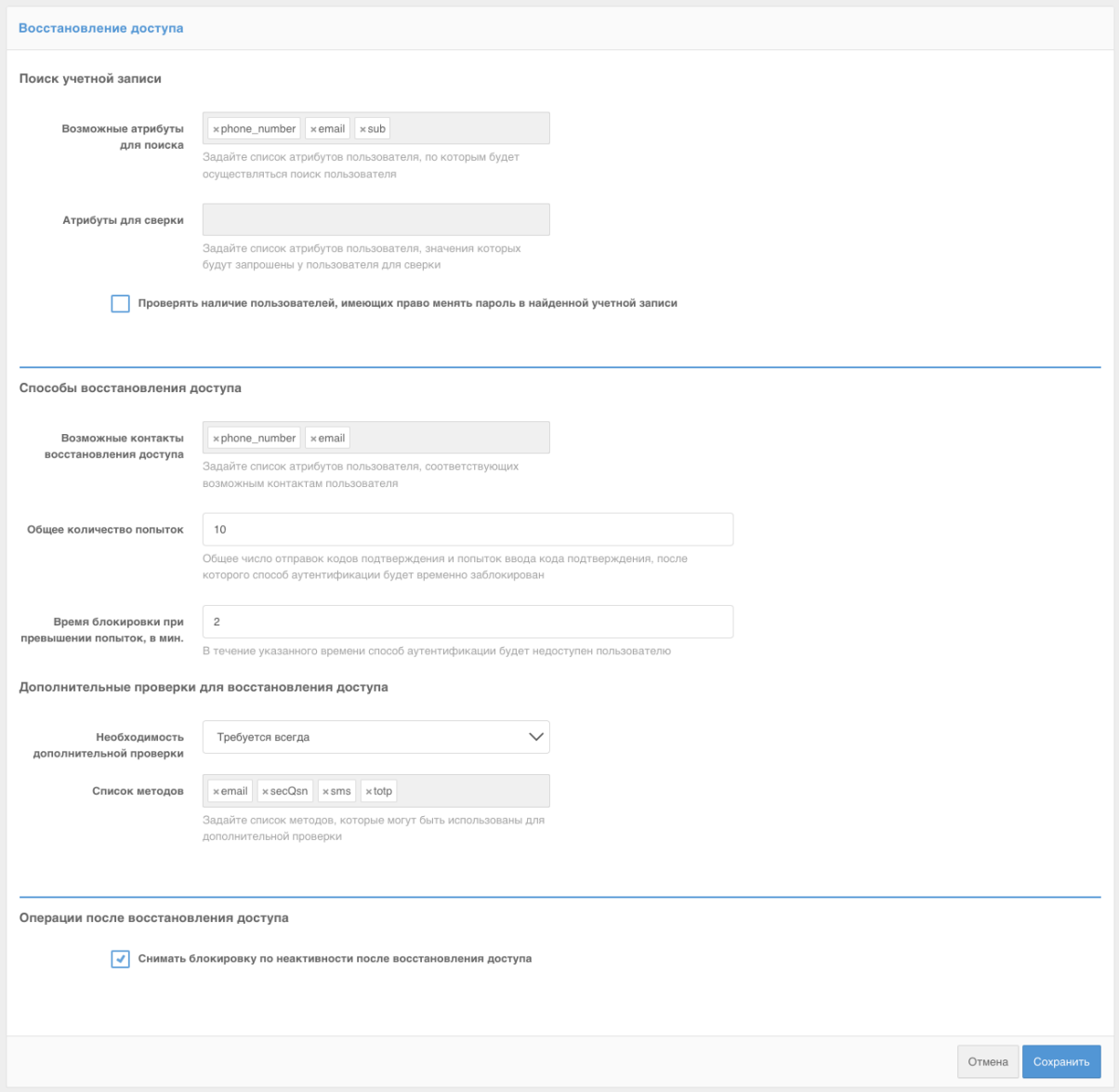
С помощью настроек Общее количество попыток и Время блокировки при превышении попыток, в мин. можно ограничить количество попыток запроса отправки и неуспешного ввода кодов подтверждения, отправленных по электронной почте и SMS для учетной записи, при превышении которых временно для учетной записи будет ограничена возможность восстановления пароля.

Настройка Необходимость дополнительной проверки определяет, в каких случаях должна выполняться дополнительная аутентификация при восстановлении доступа. Возможные значения настройки:

- Отсутствует – дополнительная аутентификация не требуется;
- Согласно настройкам пользователя в личном кабинете – дополнительная аутентификация требуется, если пользователю включил для своей учетной записи двухфакторную аутентификацию;
- Требуется всегда – дополнительная аутентификация требуется всегда;
- Требуется, если доступна – дополнительная аутентификация требуется, если для пользователя доступен хотя бы один из методов, указанных в настройке Список методов.

В случае если требуется дополнительная аутентификация, то в настройке Список методов можно выбрать доступные методы аутентификации для подтверждения восстановления доступ: подтверждение кода, полученного по электронной почте, по SMS, с помощью кода, сгенерированного TOTP-приложением, с помощью ответа на контрольный вопрос.

Настройка Снимать блокировку по неактивности после восстановления доступа определяет, что для заблокированных по причине длительной неактивности учетных записей разрешено восстановление пароля, и что после замены пароля в результате успешного восстановления блокировка по причине длительной неактивности должна быть отменена.



Тексты формы

После определения набора *атрибутов для сверки* (страница 199) необходимо задать соответствующие им тексты в форме восстановления доступа. Для этого используйте *стандартный алгоритм* (страница 301). Добавьте тексты для следующих строк:

- `recovery.page.verify.<имя_атрибута>.label`: название поля для ввода значения атрибута;
- `recovery.page.verify.<имя_атрибута>.placeholder`: текст внутри поля для ввода значения атрибута.

Список 10: Пример задания текстов для атрибутов phone_number и family_name

```
recovery.page.verify.phone_number.label=Номер мобильного телефона
recovery.page.verify.phone_number.placeholder=Введите номер, указанный при␣
↪регистрации
recovery.page.verify.family_name.placeholder=Фамилия
recovery.page.verify.family_name.placeholder=Введите фамилию
```

2.2.5 Администрирование пользователей

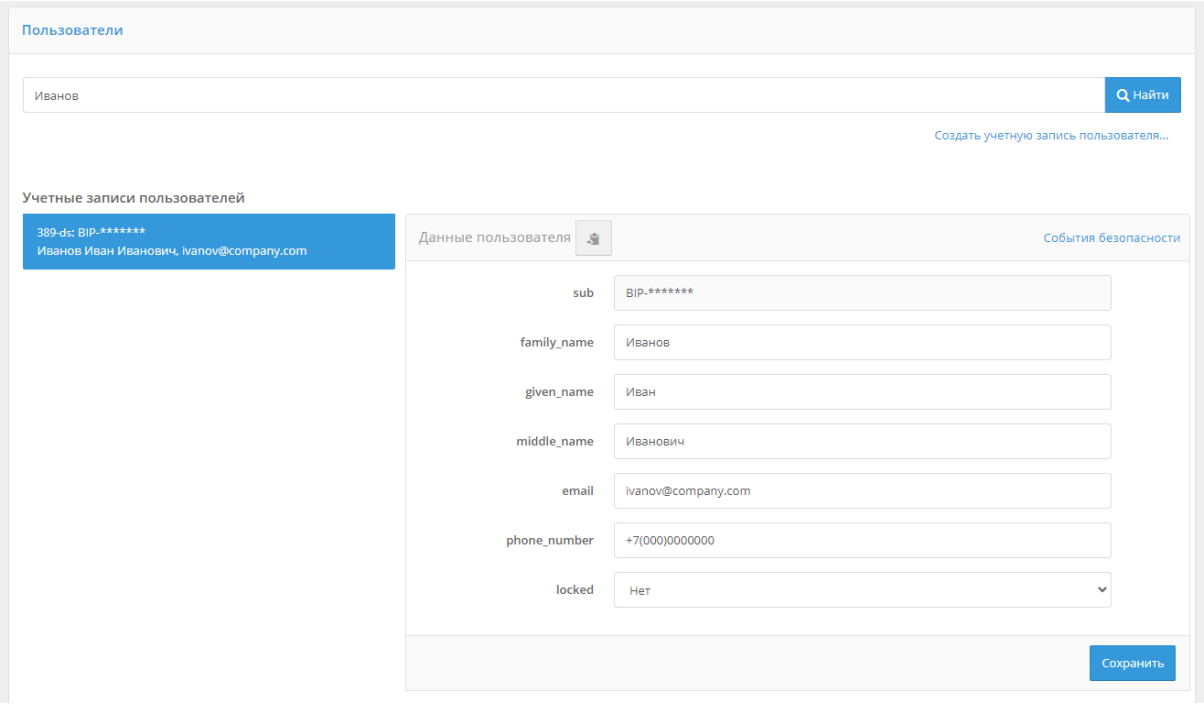
Настоящий раздел посвящен администрированию пользователей в Blitz Identity Provider.

Управление учетными записями

В разделе Пользователи консоли управления администратор Blitz Identity Provider может осуществлять следующие операции:

- *поиск учетных записей пользователей* (страница 202);
- *добавление учетной записи пользователя* (страница 202);
- *просмотр и редактирование атрибутов учетной записи пользователя* (страница 205);
- *сброс сессий пользователя* (страница 205);
- *изменение пароля учетной записи пользователя* (страница 205);
- *просмотр и отвязка учетных записей внешних поставщиков идентификации* (страница 206);
- *привязка устройств для проведения двухфакторной аутентификации* (страница 207);
- *просмотр групп, в которые включен пользователь, управление членством пользователя в группах* (страница 208);
- *просмотр, привязка, удаление ключей безопасности пользователя* (страница 212);
- *просмотр прав учетной записи пользователя, назначение и отзыв прав* (страница 209);
- *просмотр разрешений, выданных пользователем приложениям* (страница 213);
- *просмотр и удаление запомненных устройств* (страница 211);
- удаление учетной записи пользователя.

Общий вид страницы управления данными пользователей представлен на рисунке.



Поиск пользователей

Для поиска пользователей необходимо ввести идентификатор пользователя и нажать на кнопку *«Найти»*. В качестве отображаемого идентификатора используется атрибут, определенный в разделе *«Источники данных»* в качестве базового идентификатора, а также атрибуты, отмеченные как поисковые.

Перечень найденных пользователей содержит:

- значение идентификатора найденного пользователя;
- хранилище, в котором найден пользователь;
- имя пользователя, сконфигурированное в разделе *«Источники данных»*.

Нажатие на любую из найденных учетных записей открывает детальную информацию о пользователе.

Также доступны:

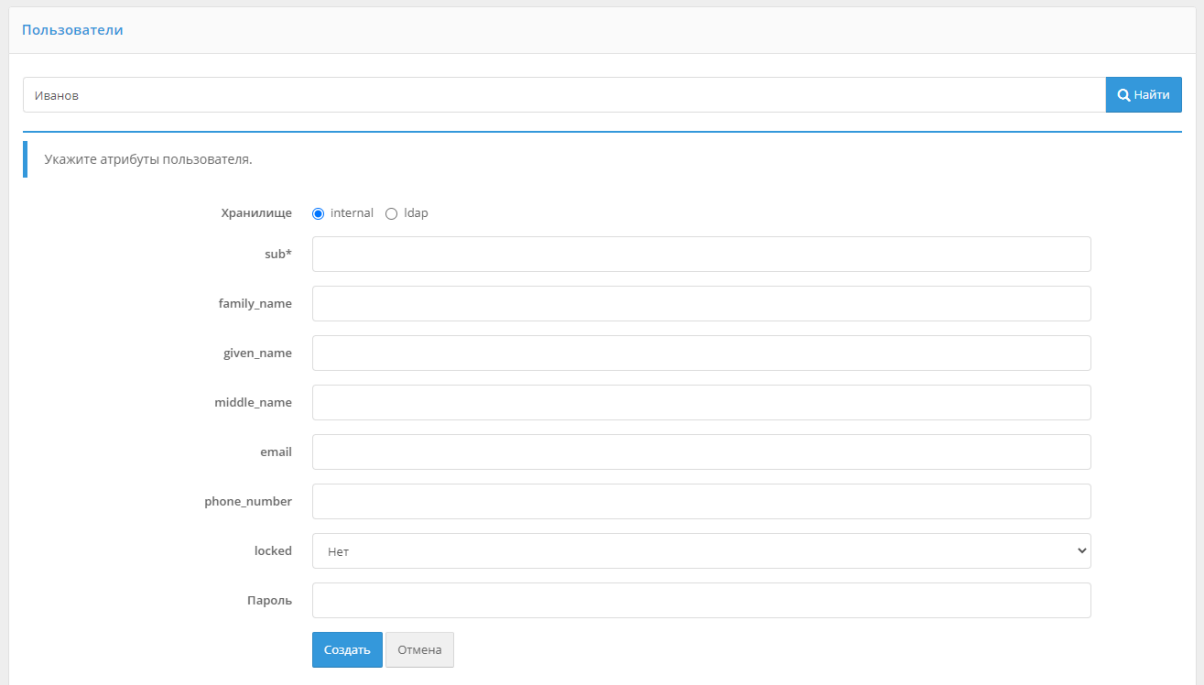
- кнопка копирования ссылки на найденного пользователя – при ее нажатии ссылка на пользователя копируется в буфер обмена;
- ссылка «События безопасности» для быстрого перехода к просмотру событий безопасности за текущий день, в которых найденный пользователь фигурирует в качестве объекта доступа.

Добавление пользователя

Для добавления новой учетной записи требуется нажать на ссылку *«Создать учетную запись пользователя…»*. В открывшемся окне:

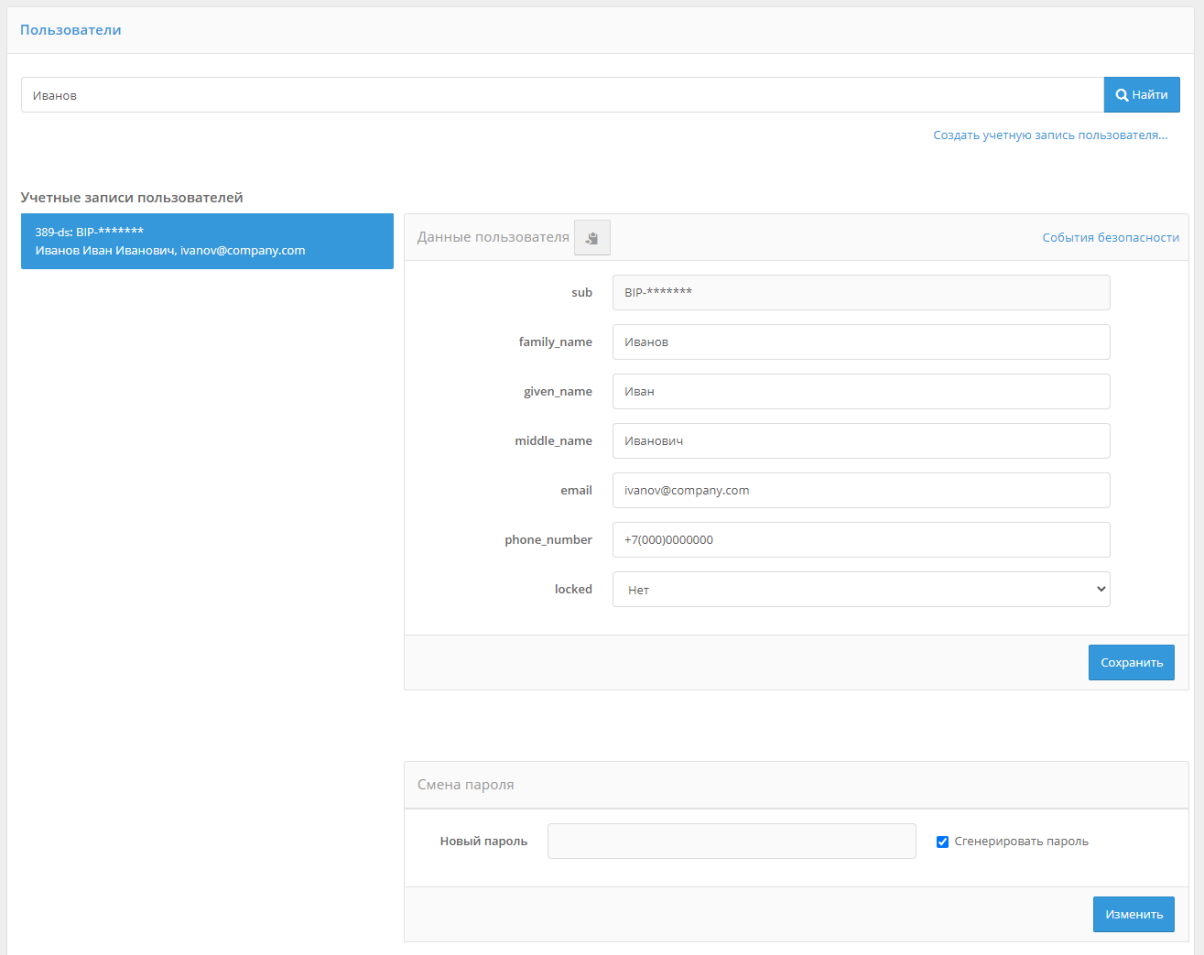
- указать хранилище, в которым следует сохранить данные пользователя;
- задать все необходимые атрибуты;
- нажать на кнопку *«Создать»*.

Важно: При создании учетной записи следует учитывать те ограничения, которые настроены для хранилища данных, в которое осуществляется запись. Например, если сохранение производится в LDAP-каталог, то должны быть заполнены все обязательные атрибуты, не нарушены ограничения на уникальность атрибутов и пр. При этом с точки зрения Blitz Identity Provider обязательным является только идентификатор и обязательные атрибуты (соответствующие атрибуты отмечены знаком «звездочка» (*)).



Просмотр и изменение атрибутов пользователя

При нажатии на идентификатор любого найденного пользователя отображается информация о нем – карточка пользователя. Она содержит значения атрибутов, которые были определены в разделе *«Источники данных»*, а также привязанные учетные записи внешних поставщиков идентификации, устройства пользователя, ключи безопасности и др.



На карточке пользователя можно совершать следующие операции:

- редактировать атрибуты пользователя;
- сбросить сессии пользователя;
- изменять пароль;
- просматривать перечень привязанных учетных записей внешних поставщиков аутентификации, отвязывать внешние учетные записи;
- изменять требуемый уровень аутентификации для пользователя;
- привязывать или удалять устройства для проведения аутентификации: генераторы разовых паролей и мобильные приложения для получения push-уведомлений;
- просматривать группы, в которые включен пользователь;
- просматривать права пользователя и права, которые имеются в отношении данного пользователя;
- просматривать и удалять запомненные устройства и браузеры пользователя;
- просматривать, добавлять и удалять ключи безопасности пользователя;
- просматривать и удалять выданные приложениям разрешения.

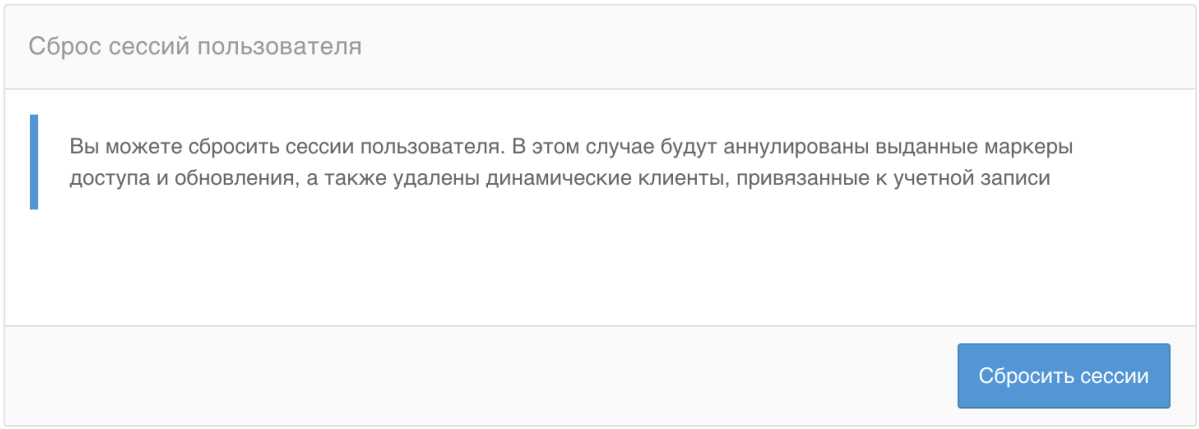
Редактирование атрибутов

При просмотре карточки выбранной учетной записи пользователя администратор может изменить любой атрибут пользователя. При редактировании учетной записи следует учитывать те ограничения, которые настроены для хранилища данных, в которое осуществляется запись.

Следует учитывать, что при изменении данных через интерфейс редактирования атрибутов не учитываются правила, используемые в процессе самостоятельной регистрации пользователя. Например, изменение адреса электронной почты или номера мобильного телефона не требует подтверждения.

Сброс сессий

Для сброса сессий пользователя используется кнопка Сбросить сессии в блоке Сброс сессий пользователя.



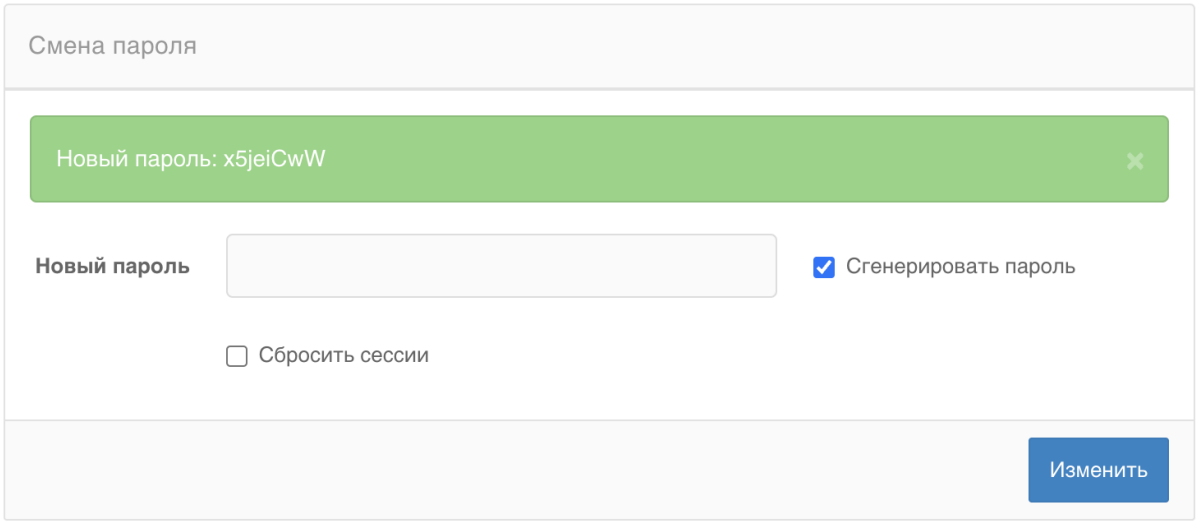
При сбросе сессий пользователя выполняются следующие действия:

- выданные на пользователя приложениям маркеры безопасности (маркеры доступа, маркеры обновления, маркеры идентификации) становятся недействительными – при вызове в Blitz Identity Provider сервиса интроспекции с такими маркерами сервис вернет, что маркер недействителен;
- в запомненных для пользователя устройствах убираются флаги доверенных устройств и запоминания на них длительных сессий;
- привязанные к учетной записи пользователя выпущенные для мобильных приложений пары динамических `client_id`/`client_secret` аннулируются;
- запомненные в браузере пользователя SSO-сессии становятся недействительными, так что при очередном запросе со стороны приложений идентификации в Blitz Identity Provider будет запрошена новая идентификация и аутентификация.

Смена пароля

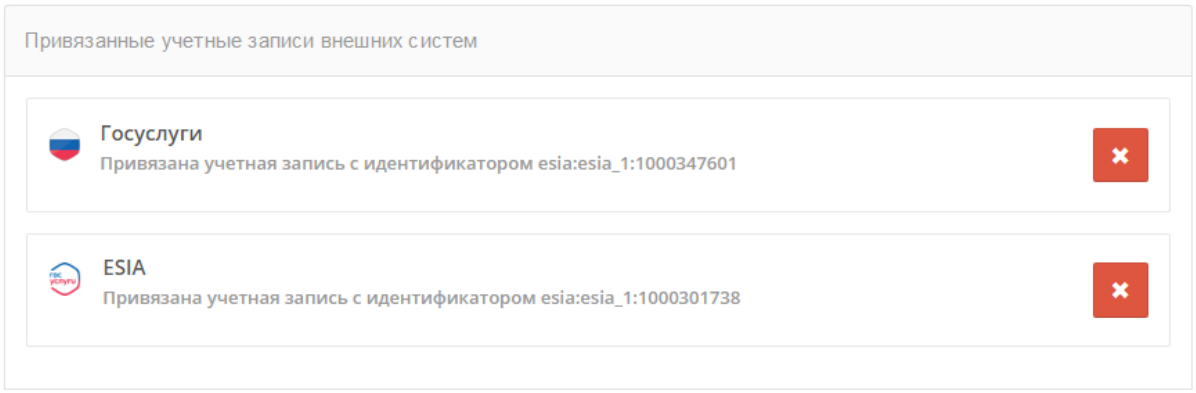
Для смены пароля используется блок Смена пароля. Новый пароль можно ввести вручную, либо сгенерировать – для этого необходимо оставить чекбокс Сгенерировать пароль. Новый пароль будет отображен в информационном блоке успешного выполнения операции. При смене пароля можно также установить чекбокс Сбросить сессии, тогда одновременно со сменой пароля будут сброшены сессии пользователя.

При задании нового пароля вручную следует учитывать ограничения парольной политики для того хранилища, куда сохраняется пароль.



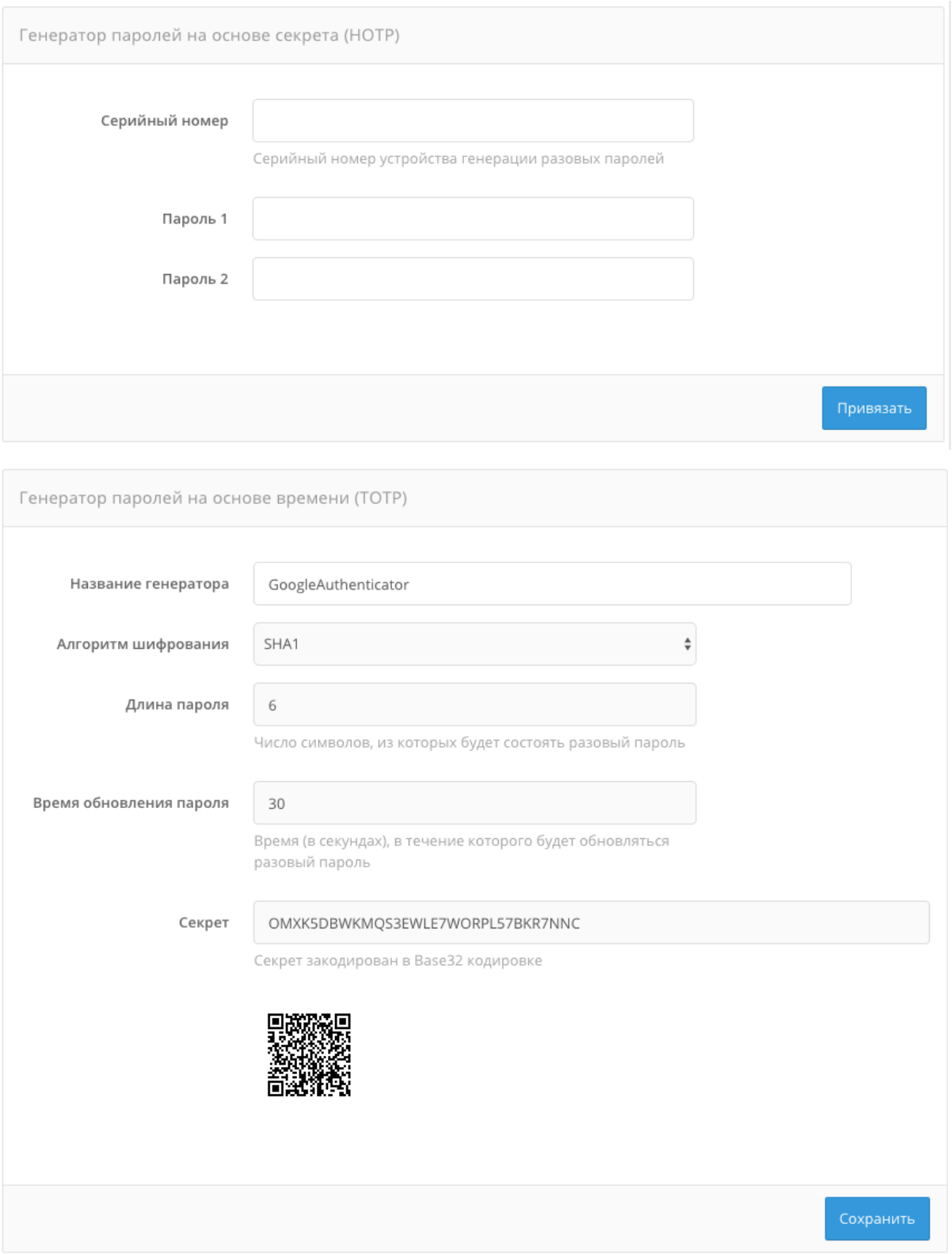
Просмотр и отвязка внешних поставщиков

В блоке «Привязанные учетные записи внешних систем» можно посмотреть перечень аккаунтов внешних поставщиков идентификации (социальных сетей, банков, ЕСИА, Mos ID и др.), привязанных к учетной записи найденного пользователя. Каждая привязка характеризуется уникальным идентификатором, где последняя часть – это внутренний идентификатор аккаунта в соответствующем поставщике идентификации. Например, в записи `esia:esia_1:1000347601` последняя часть `(1000347601)` – это идентификатор аккаунта в ЕСИА. При необходимости можно удалить связь с внешней учетной записью.



Привязка устройств для 2FA по разовому паролю

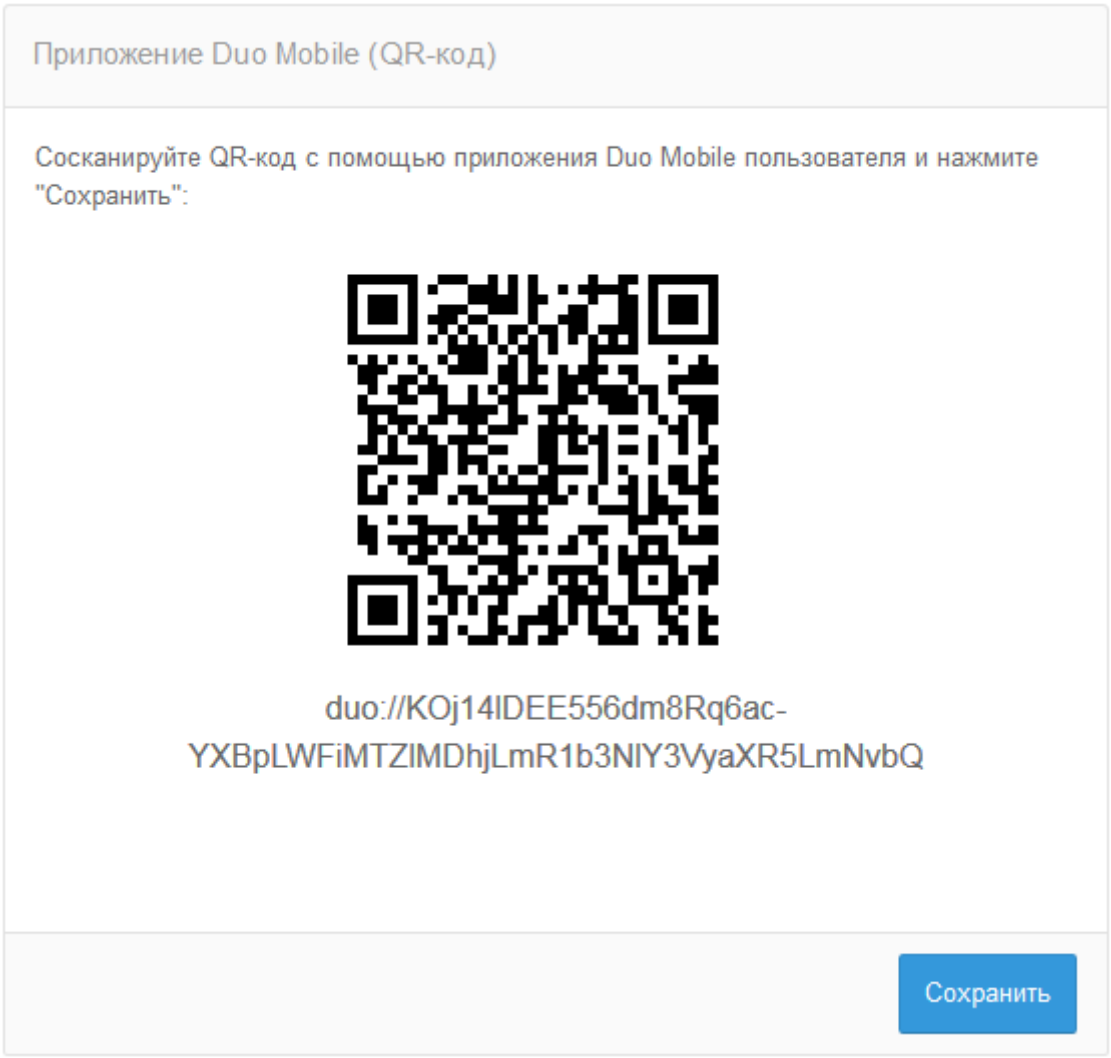
Администратор может привязать к учетной записи выбранного пользователя средство для проведения двухфакторной аутентификации. Например, можно привязать аппаратный HOTP/TOTP генератор по серийному номеру, либо привязать к учетной записи по QR-коду мобильное приложение, осуществляющее выработку TOTP-кодов.



Привязка Duo Mobile

Для проведения аутентификации средствами Duo Mobile необходимо провести привязку мобильного приложения к учетной записи пользователя. Рекомендуемый сценарий – пользователь самостоятельно привязывает свое мобильное приложение в веб-приложении *«Личный кабинет»*.

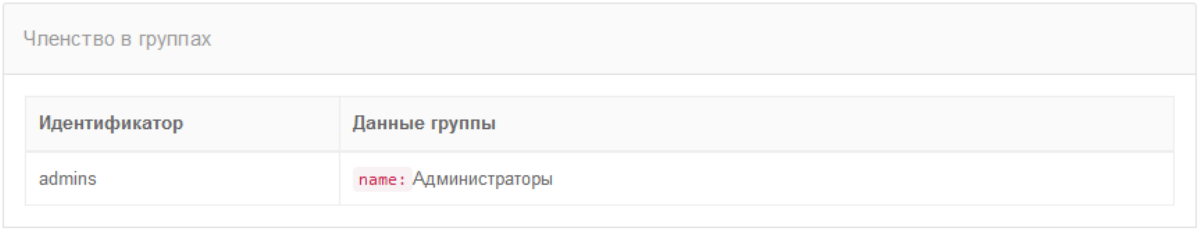
Альтернативный способ привязки – через консоль управления. Для этого необходимо в разделе *«Пользователи»* найти необходимую учетную запись и блок настроек *«Приложение Duo Mobile (QR-код)»*. В этом блоке следует нажать на кнопку *«Привязать Duo Mobile»*, далее сосканировать отображенный QR-код мобильным приложением Duo Mobile.



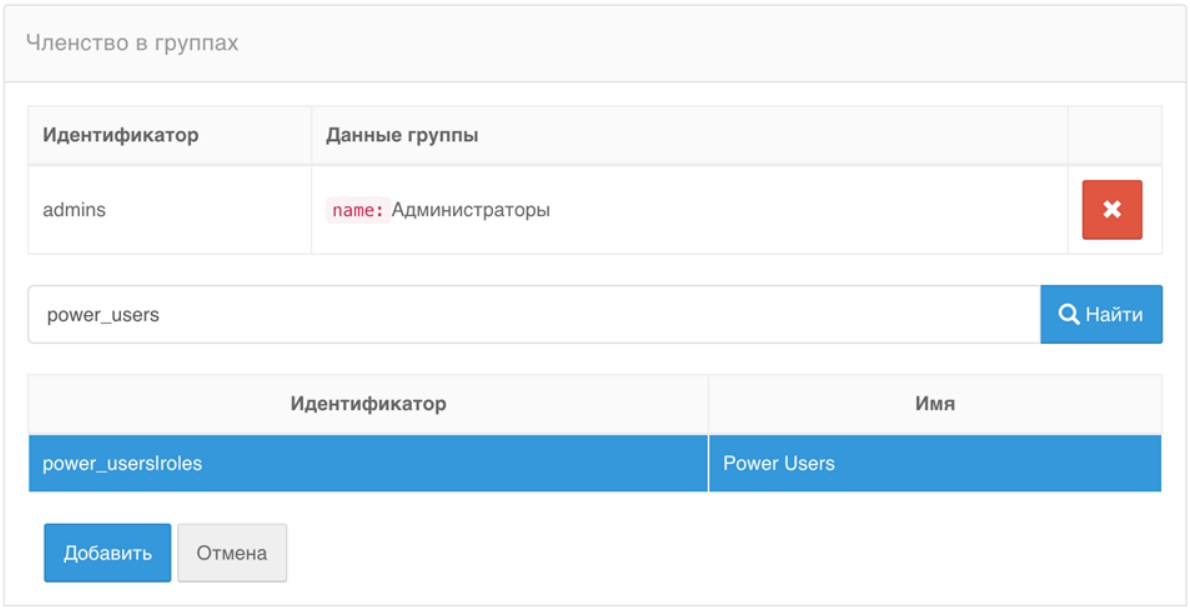
Управление членством в группах

Если пользователь включен в группы, то эта информация будет отображена в блоке *«Членство в группах»*. По каждой группе будет отображены следующие данные:

- идентификатор группы;
- значения атрибутов группы.



Можно исключить пользователя из группы с помощью кнопки удаления или добавить пользователя в другую группу с помощью ссылки *«Добавить в группу»*. Для добавления пользователя в группу нужно будет ввести значение атрибута, идентифицирующего группу, нажать кнопку *«Найти»*, выбрать подходящую группу из списка найденных, и нажать кнопку *«Добавить»*.

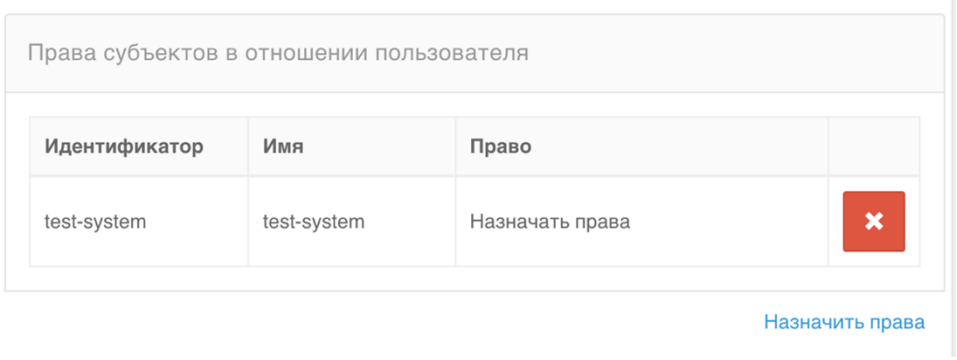


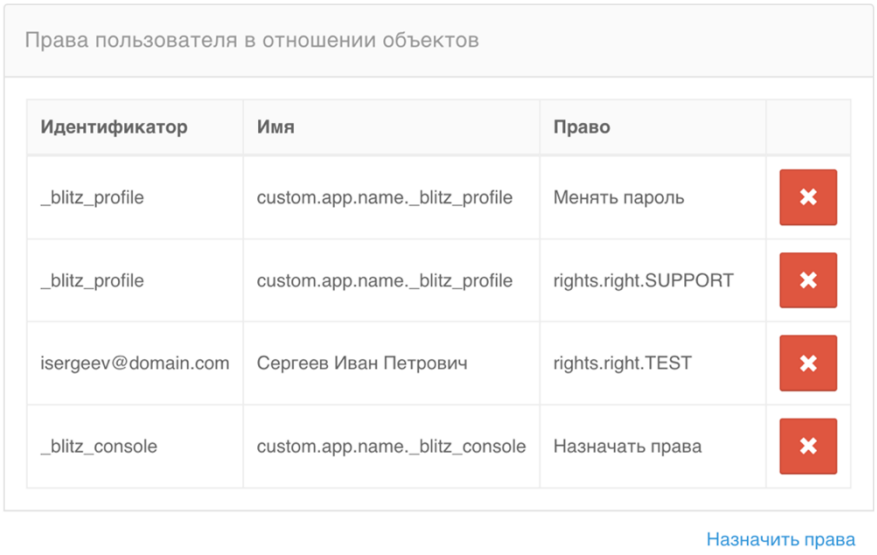
Просмотр, назначение и отзыв прав

Если в отношении пользователя есть права со стороны приложений или других учетных записей, то это будет отображено в блоке *«Права в отношении пользователя»*. Если пользователь имеет права в отношении объектов, например, других учетных записей, то это будет отображено в блоке *«Права пользователя в отношении объектов»*.

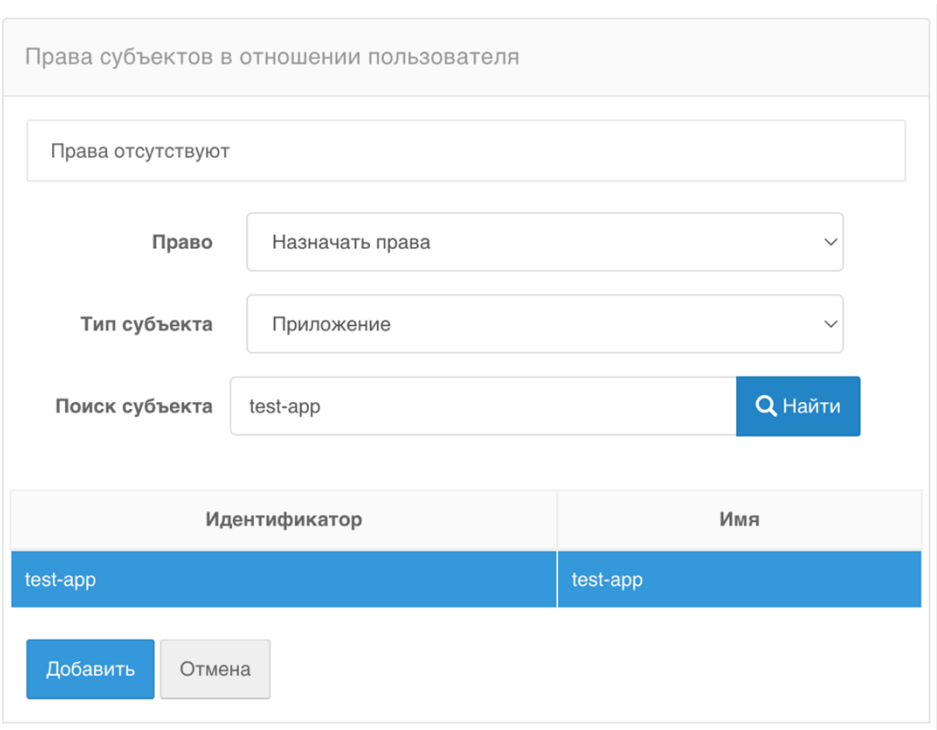
Каждое право характеризуется следующими параметрами:

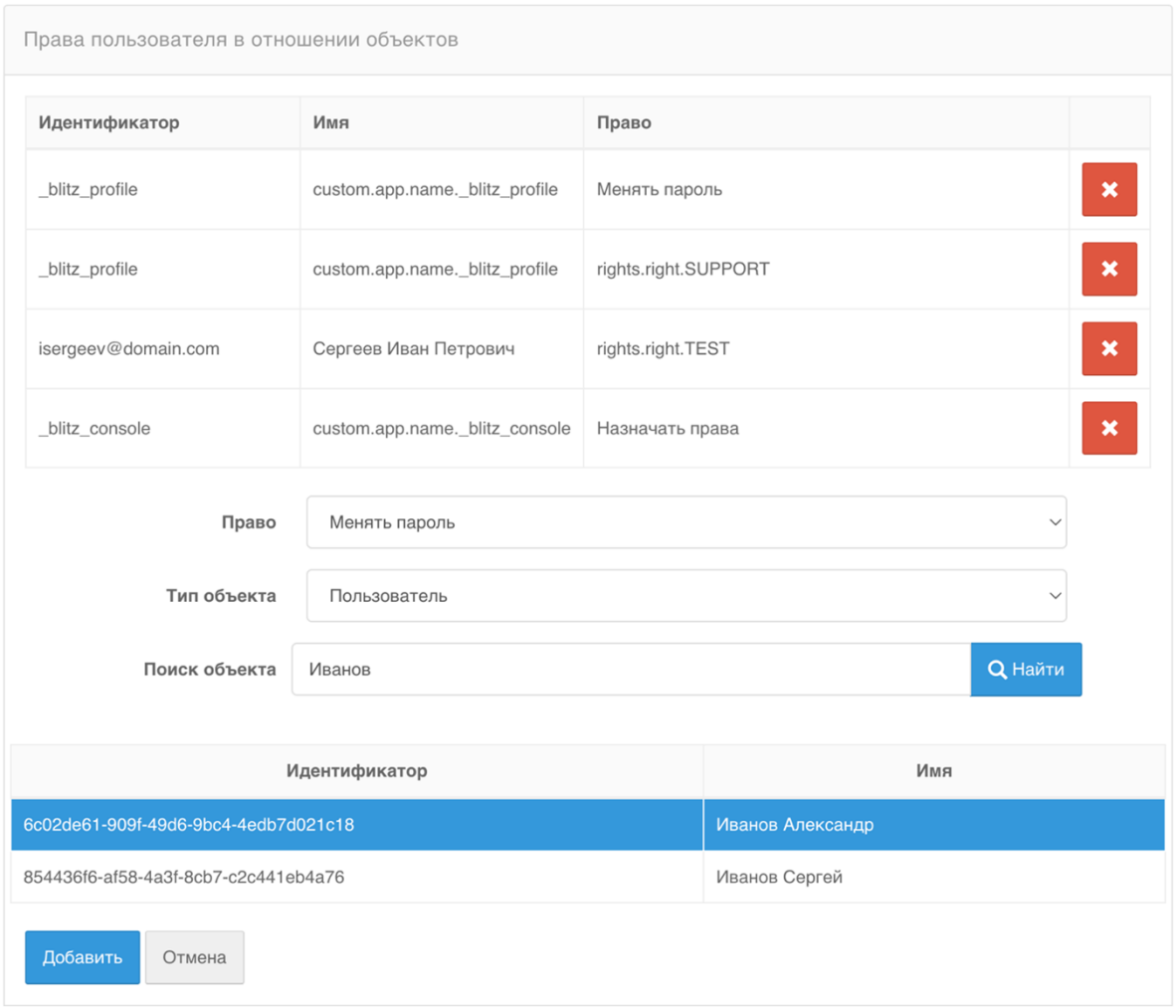
- идентификатор объекта;
- имя;
- право.





Отозвать право доступа можно с помощью кнопки удаления рядом с правом доступа. Назначить право доступа можно с помощью ссылки *«Назначить права»*. При этом надо будет выбрать назначаемое право доступа из списка, тип субъекта (пользователь или приложение) или объекта (пользователь, группа или приложение), найти и выбрать субъекта/объекта.

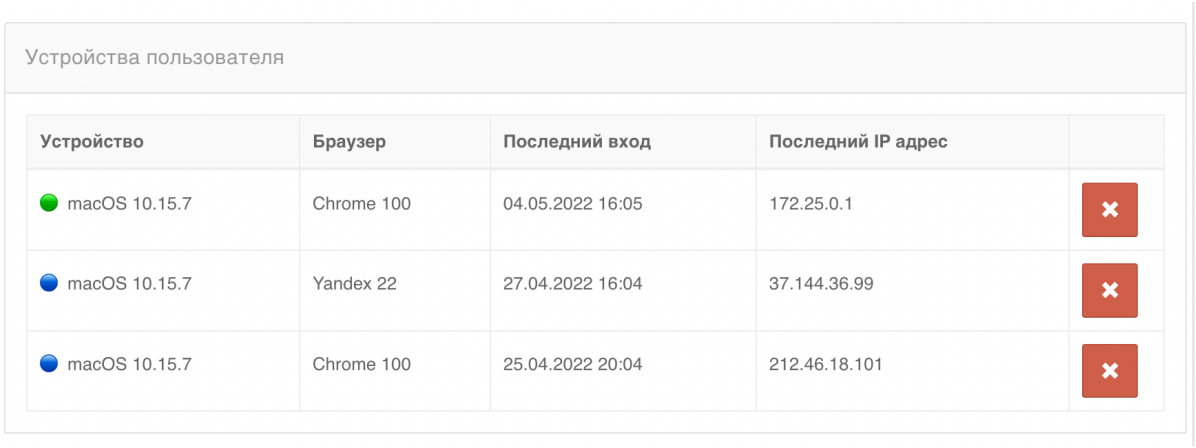




Запомненные устройства и браузеры

Администратор имеет возможность просмотреть устройства и браузеры, с которых пользователь осуществлял вход с использованием своей учетной записи. Описание устройств включает:

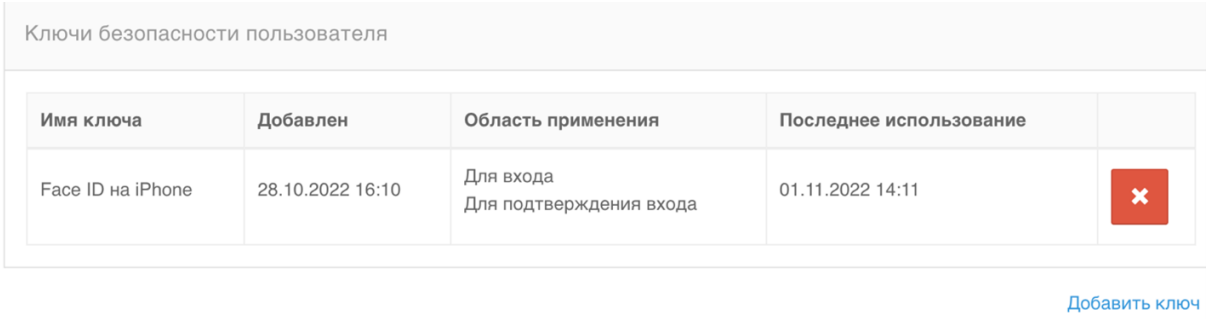
- признак того, запомнена ли на устройстве сессия входа и является ли устройство доверенным. Признак кодируется с помощью цвета:
  - серый – на устройстве не запомнена сессия входа и устройство не является доверенным;
  - желтый – на устройстве на запомнена сессия входа, но устройство является доверенным;
  - синий – на устройстве запомнена сессия входа, но устройство не является доверенным;
  - зеленый – на устройстве запомнена сессия входа и устройств является доверенным.
- имя и версия операционной системы устройства, определенные на основе `UserAgent`;
- имя и версия браузера, определенные на основе `UserAgent`;
- дата и время последнего входа с данного устройства и браузера;
- IP-адрес пользователя, который был определен при последнем входе с данного устройства и браузера.



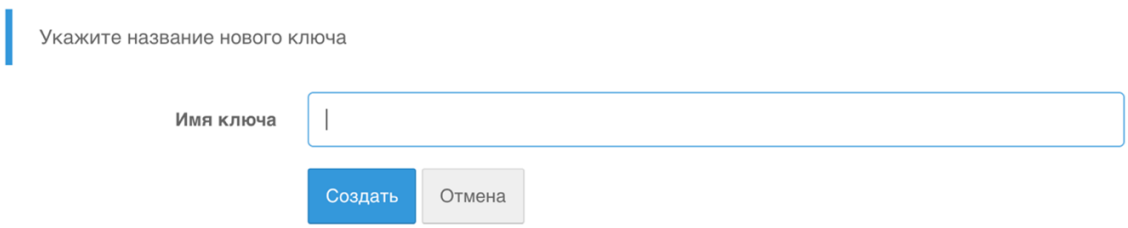
Ключи безопасности

Администратор имеет возможность просмотреть перечень ключей безопасности (Passkey, WebAuthn, FIDO2, U2F), зарегистрированных для учетной записи пользователя. Для каждого ключа безопасности указаны:

- имя ключа;
- дата и время регистрации ключа;
- область применения (для Passkey и FIDO2 – для входа и для подтверждения входа; для U2F – только для подтверждения входа);
- дата и время последнего использования ключа.



Администратор может зарегистрировать новый ключ безопасности с помощью ссылки «Добавить ключ». В обычном сценарии использования ключи безопасности себе добавляет сам пользователь в момент входа (онбординг) или через личный кабинет.



Возможность добавления ключа администратором может быть полезна в следующих сценариях:

- Администратор лично выдает пользователям аппаратный FIDO2/U2F ключ и привязывает его к учетной записи. Для доступа к приложениям компании используется двухфакторная аутентификация.
- Администратору в целях технической поддержки нужна возможность войти под учетной записью пользователя. Сброс от учетной записи пароля доставит пользователю неудобства – вместо этого

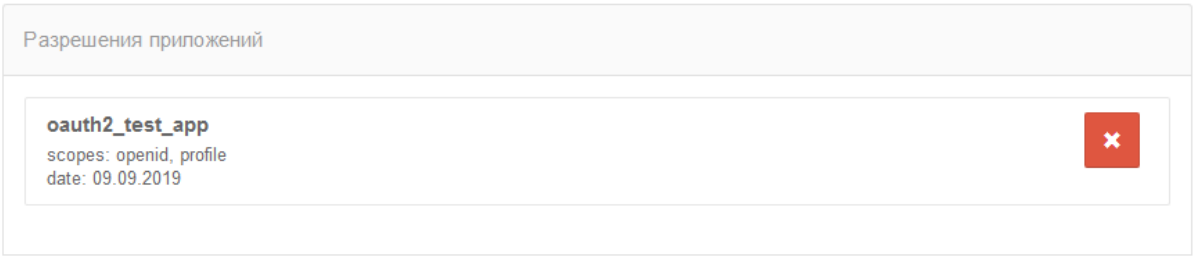
можно зарегистрировать ключ безопасности и использовать его для входа. Все действия по регистрации и удалению ключей безопасности регистрируются как события безопасности.

Выданные приложениям разрешения

Администратор имеет возможность просмотреть перечень разрешений, выданных пользователем приложениям.

Каждое разрешение описывается:

- идентификатор приложения;
- перечень разрешений (`scope`);
- дата выдачи разрешений.



Управление группами пользователей

Включение отображения групп в blitz.conf

Если в Blitz Identity Provider настроена возможность работы с группами пользователей, то в консоли управления появится раздел Группы.

Чтобы включить возможность просмотра групп пользователей, необходимо добавить блок настроек `blitz.prod.local.idp.groups` следующего вида:

```
"groups": {
  "profiles": [
    {
          "attrsMap": {
              "INN": "inn",
              "NAME": "orgname",
              "OGRN": "ogrn",
              "desc": "description",
              "members": {
                  "name": "uniqueMember",
                  "type": "strings"
              }
          },
          "attrsMeta2": [
              {
                  "INN": "inn"
              },
              {
                  "NAME": "orgname"
              },
              {
                  "OGRN": "ogrn"
              },
              {
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                    "desc": "description"
                },
                {
                    "members": {
                        "name": "uniqueMember",
                        "type": "strings"
                    }
                }
            ],
            "filter": "objectClass=organizationgroup",
            "groupStore": "389ds",
            "id": "orgs",
            "type": "mirror",
            "memberOfWithNested": true
        }
    ],
    "stores": {
        "list": [
            {
                "baseDN": "ou=external,ou=groups,dc=test",
                "desc": "Группы",
                "id": "389ds",
                "idAttrName": "cn",
                "ldapStore": "389ds",
                "memberOfAttrName": "memberOf",
                "membersAttrName": "uniqueMember",
                "newGroupAttrs": [
                    {
                        "attr": "objectclass",
                        "format": "strings",
                        "value": "top,groupOfUniqueNames,organizationgroup"
                    },
                    {
                        "attr": "dn",
                        "format": "string",
                        "value": "cn=${id},ou=external,ou=groups,dc=test"
                    }
                ],
                "searchScope": "SUB",
                "type": "ldap_based"
            }
        ]
    }
}
```

Особенности указания настроек:

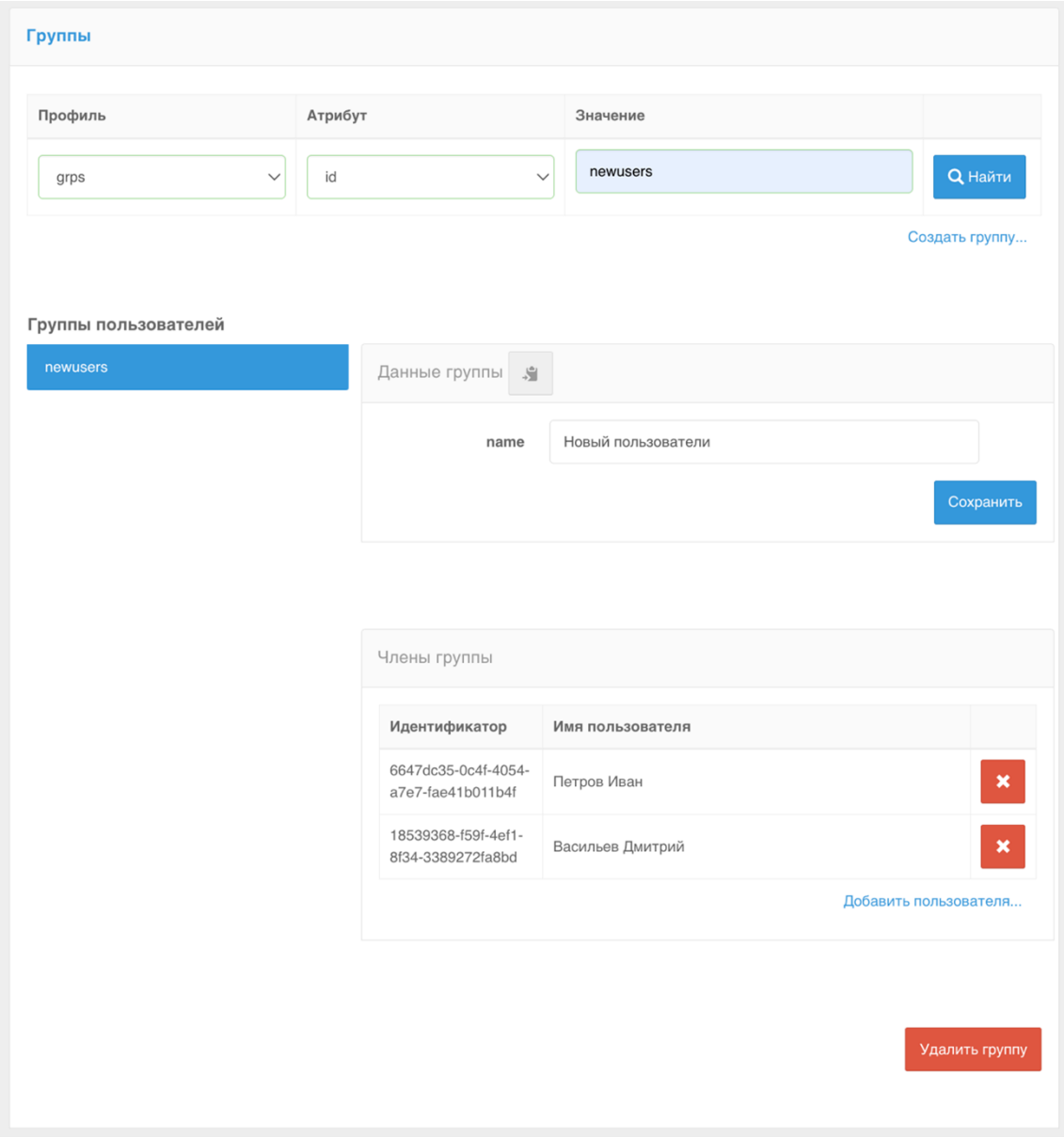
- в `profiles.groupStore`, `stores.list.id`, `stores.ldapStore` должен быть идентификатор LDAP-каталога, используемого для хранения пользователей;
- в `profiles.attrsMap` и в `stores.list.idAttrName` должны быть указаны атрибуты группы (класс `groups`), например `name`. Имена атрибутов при желании можно назвать и по-другому, поддерживаются только LDAP-атрибуты типа `String`;
- в `stores.list.baseDN` нужно проверить (и исправить если необходимо) путь для хранения организаций в LDAP. Если путь будет исправлен, то скорректировать также настройку `"value": "cn=${id},ou=external,ou=groups,dc=test"` соответствующим образом;
- в `profiles.memberOfWithNested` укажите значение `true` или `false` в зависимости от того, есть ли необходимость искать все группы у пользователя;
- для Microsoft AD параметр `membersAttrName` должен иметь значение `memberOf`.

Работа с группами

В разделе Группы можно осуществлять поиск групп по одному из сконфигурированных атрибутов, редактировать группы, создавать и удалять группы, управлять членством пользователей в группах.

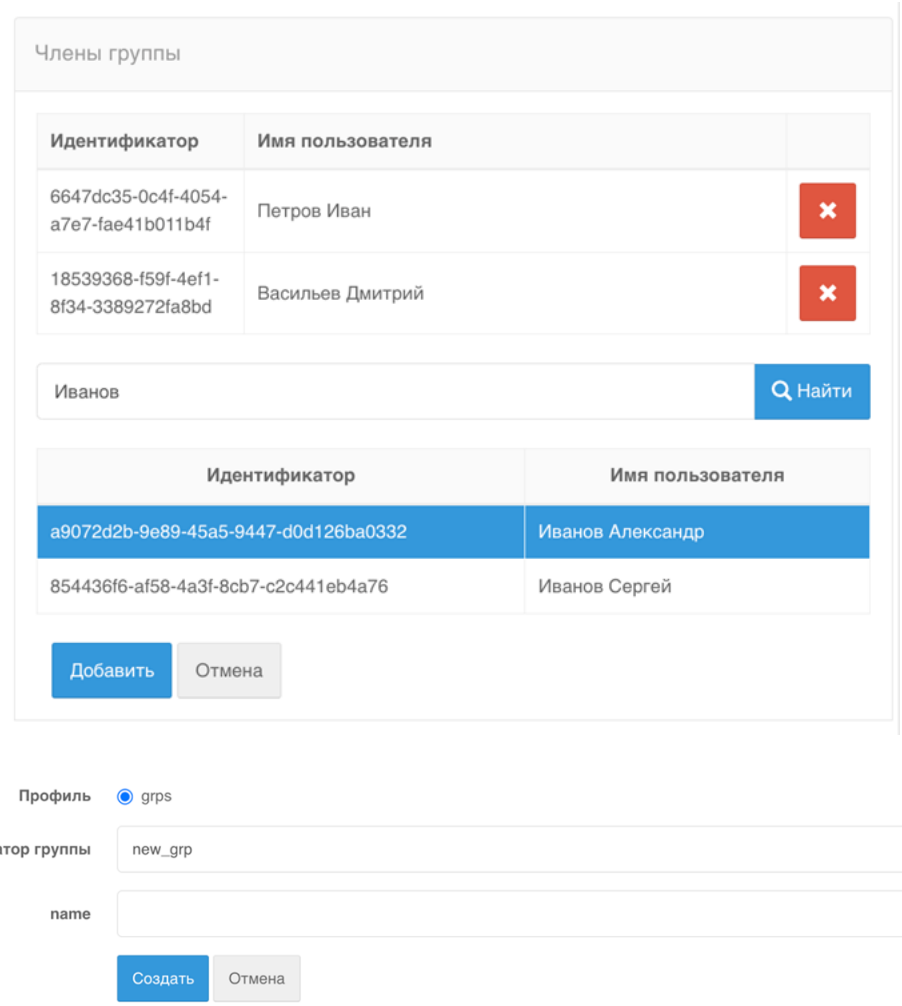
По каждой найденной группе отображаются ее атрибуты. Кроме того, в блоке Члены группы отображаются все пользователи, включенные в данную группу. По каждому пользователю отображается:

- идентификатор;
- имя пользователя – согласно шаблону, определенному в разделе Источники данных (`Имя пользователя в консоли`).



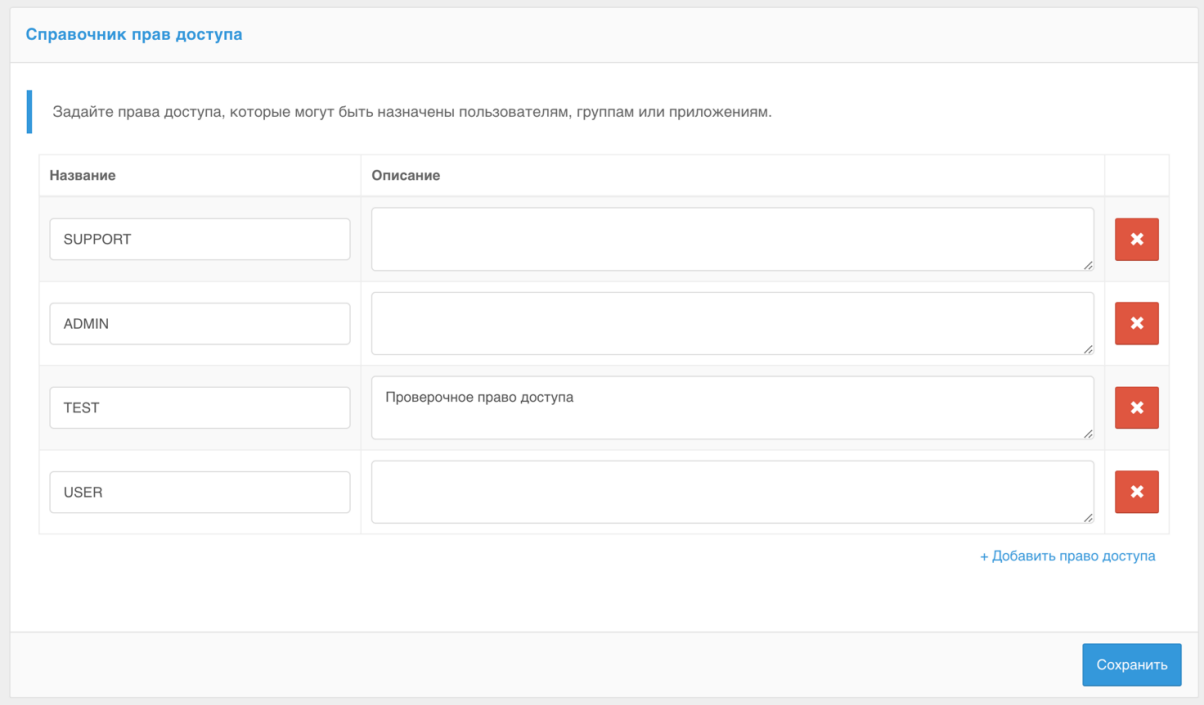
Доступны возможности по редактированию атрибутов группы, удалению группы, включению пользователей в группу с помощью ссылки Добавить пользователя..., исключению пользователя из группы, созданию новых групп пользователей с помощью ссылки Создать группу....

Включение пользователя в группу:



Управление правами доступа

Для ведения справочника прав доступа в Blitz Identity Provider используется раздел «Права доступа» консоли управления. Права доступа могут использоваться для контроля доступа пользователей в приложения, для контроля вызова приложениями защищаемых REST-сервисов, а также могут быть запрошены и использованы приложениями для осуществления контроля доступа пользователя к функциям приложений.



2.2.6 Уведомления и отправка сообщений

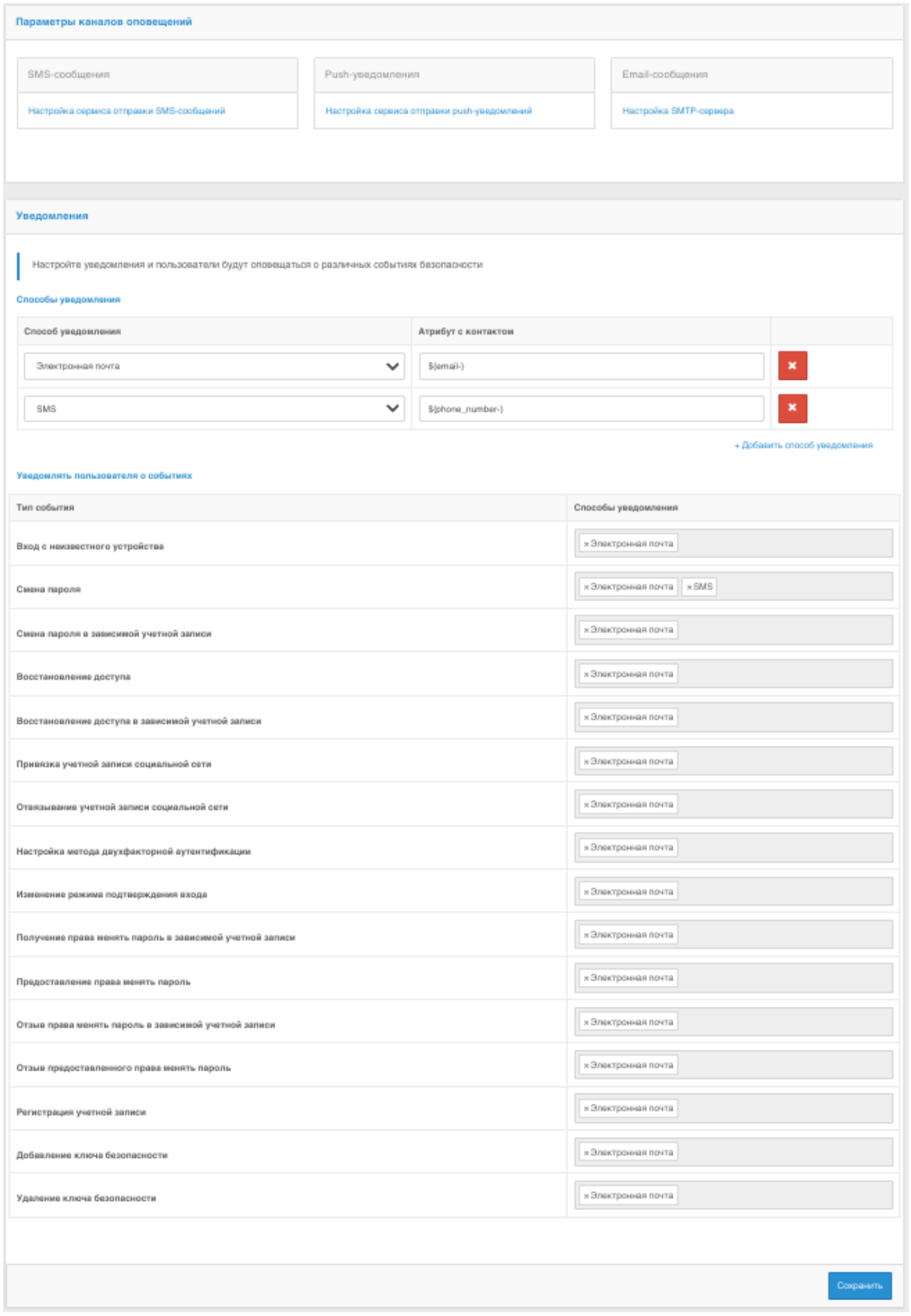
Для задания настроек уведомлений и подключения к системам отправки сообщений используется раздел *«Сообщения»* консоли управления Blitz Identity Provider. В этом разделе можно настроить уведомления и подключение к:

- сервису отправки SMS-сообщений;
- сервису отправки push-уведомлений;
- SMTP-серверу.

Для настройки уведомлений необходимо на основной странице раздела:

- выбрать канал для восстановления (электронная почта, мобильный телефон) и указать атрибут со значением этого контакта. Атрибут задается с помощью регулярного выражения, например, `${phone_number}` означает, что информация будет отправлена на телефон `phone_number`;
- выбрать события, по которым требуется отправлять уведомления. Возможно уведомление при следующих событиях:
  - вход с неизвестного устройства;
  - смена пароля;
  - смена пароля в зависимой учетной записи;
  - восстановление доступа;
  - восстановление доступа в зависимой учетной записи;
  - привязка учетной записи социальной сети;
  - отвязывание учетной записи социальной сети;
  - настройка метода двухфакторной аутентификации;
  - изменение режима подтверждения входа;
  - получение права менять пароль в зависимой учетной записи;
  - предоставление права менять пароль;

- отзыв права менять пароль в зависимой учетной записи;
- отзыв предоставленного права менять пароль;
- регистрация учетной записи;
- добавление нового ключа безопасности;
- удаление ключа безопасности.

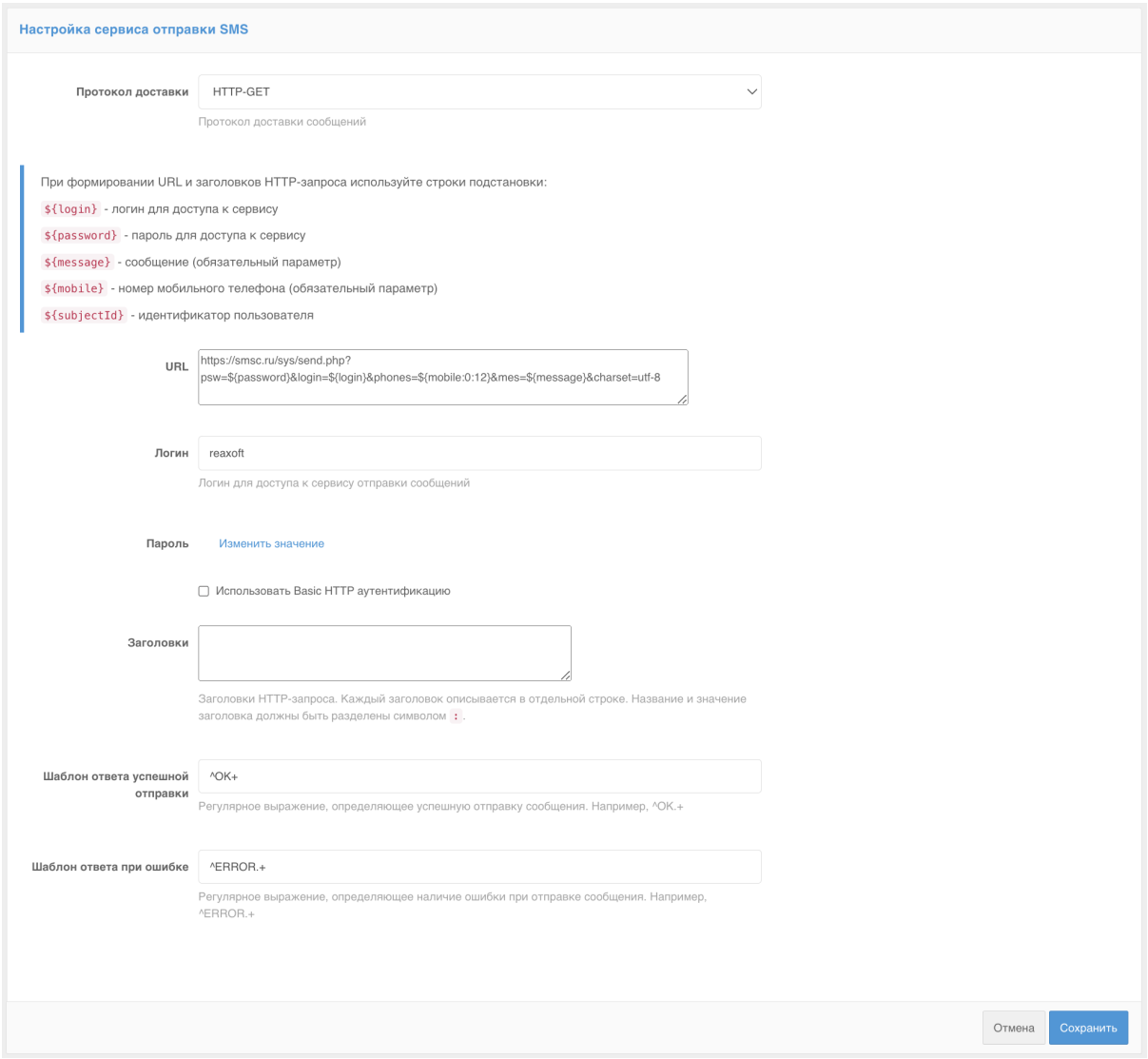

Параметры каналов оповещений
SMS-сообщения
Push-уведомления
Email-сообщения
Настройка сервиса отправки SMS-сообщений
Настройка сервиса отправки push-уведомлений
Настройка SMTP-сервера
Уведомления
Настройте уведомления и пользователи будут оповещаться о различных событиях безопасности
Способы уведомления
Способ уведомления
Атрибут с контактом
Электронная почта
${email-}
SMS
${phone_number-}
+ Добавить способ уведомления
Уведомлять пользователя о событиях
Тип события
Способы уведомления
Вход с неизвестного устройства
Смена пароля
Смена пароля в зависимой учетной записи
Восстановление доступа
Восстановление доступа в зависимой учетной записи
Привязка учетной записи социальной сети
Отвязывание учетной записи социальной сети
Настройка метода двухфакторной аутентификации
Изменение режима подтверждения входа
Получение права менять пароль в зависимой учетной записи
Предоставление права менять пароль
Отзыв права менять пароль в зависимой учетной записи
Отзыв предоставленного права менять пароль
Регистрация учетной записи
Добавление ключа безопасности
Удаление ключа безопасности
Сохранить

Подключение к SMS-шлюзу

Blitz Identity Provider необходима возможность отправлять SMS-сообщения, если используются следующие функции:

- аутентификация на основе отправки по SMS кода подтверждения (первый и второй фактор);
- информирование о важных событиях безопасности по SMS;
- изменение номера мобильного телефона через *«Профиль пользователя»*;
- восстановление забытого пароля с использованием мобильного телефона как канала подтверждения владения учетной записью;
- подтверждение номера мобильного телефона при регистрации пользователя.

Настройки задаются в консоли управления Blitz Identity Provider в разделе *«Сообщения»*.



Необходимо задать следующие настройки:

- вид протокола доставки (GET или POST);
- URL SMS-шлюза – задается в виде паттерна для формирования запроса к SMS-шлюзу для инициирования отправки им SMS. Пример настройки для SMS-шлюза:

```
https://smsc.ru/sys/send.php?psw=${password}&login=${login}&phones=${mobile}&mes=$
↪{message}&charset=utf-8
```

- логин и пароль для доступа к SMS-шлюзу. Логин и пароль могут быть переданы в качестве параметров GET-запроса или в виде HTTP-заголовка запроса (схема аутентификации HTTP Basic authentication);
- HTTP-заголовке запроса на SMS-шлюз;
- шаблон проверки ответа от шлюза, означающего успешную отправку. Задается в виде регулярного выражения;
- шаблон проверки ответа от шлюза, означающего ошибку отправки сообщения. Задается в виде регулярного выражения.

Подключение к сервису отправки push-уведомлений

Настройки push-уведомлений задаются в веб-приложении администрирования в разделе *«Сообщения»*.

Необходимо задать следующие настройки:

- вид протокола доставки (GET или POST);
- URL сервиса отправки push-уведомлений, например:

```
http://api.system.ru/json/v1.0/communication/mobile/push
```

- данные – сообщение, передаваемое в теле (`body`) запроса, например:

```
{"token":"${password}","title":"${title}","body":"${message}","msisdn":$
↪{subscriberId}}
```

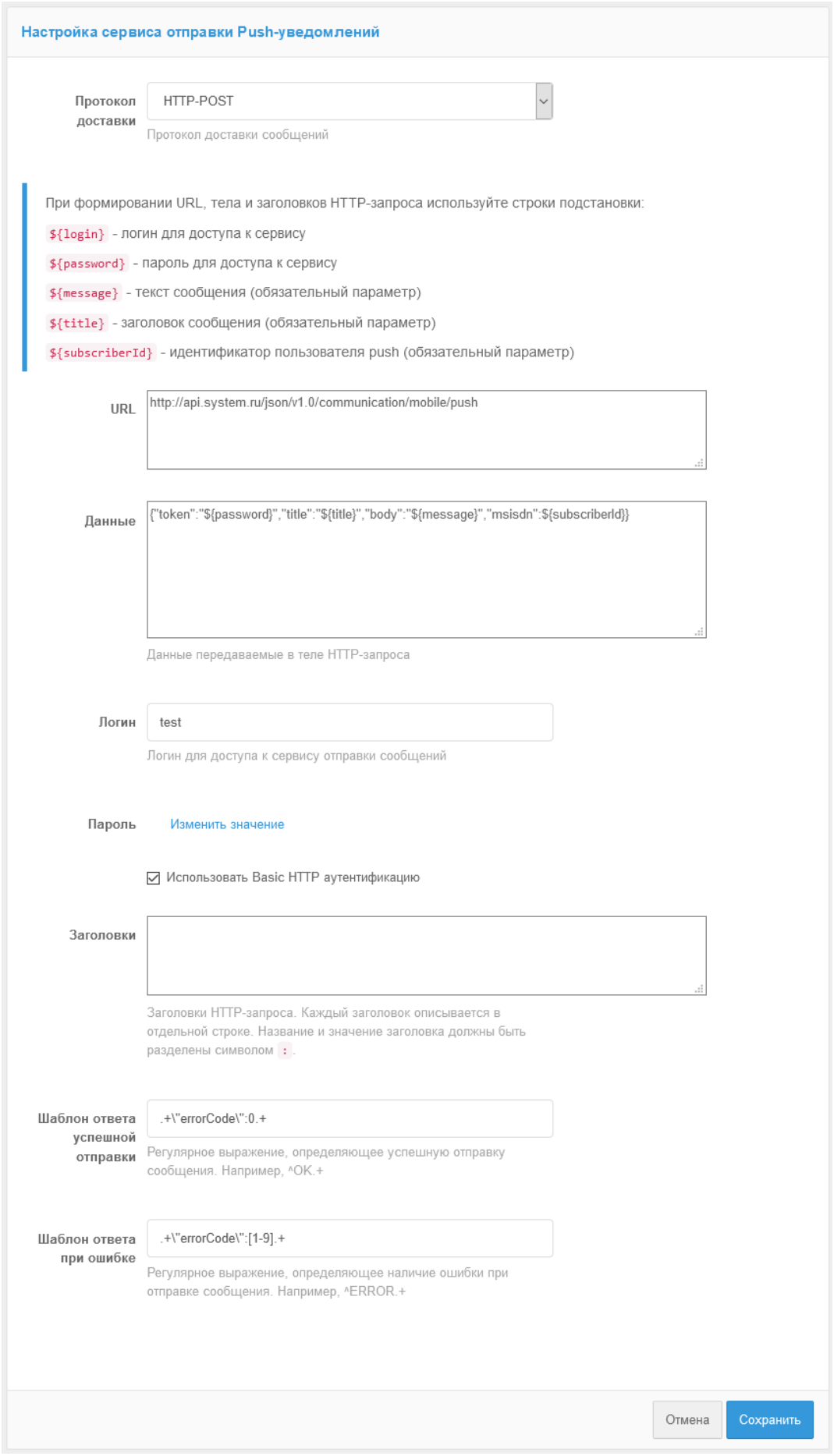
- логин и пароль для доступа к сервису. Логин и пароль могут быть переданы в качестве параметров GET-запроса или в виде HTTP-заголовка запроса (схема аутентификации HTTP Basic authentication);
- HTTP-заголовки запроса;
- шаблон проверки ответа от сервиса, означающего успешную отправку. Задается в виде регулярного выражения, например:

```
.+\"errorCode\":0.+
```

- шаблон проверки ответа от сервиса, означающего ошибку отправки сообщения. Задается в виде регулярного выражения, например:

```
.+\"errorCode\":[1-9].+
```

Пример настройки интеграции с сервисом отправки push-уведомлений отображен на рисунке ниже.

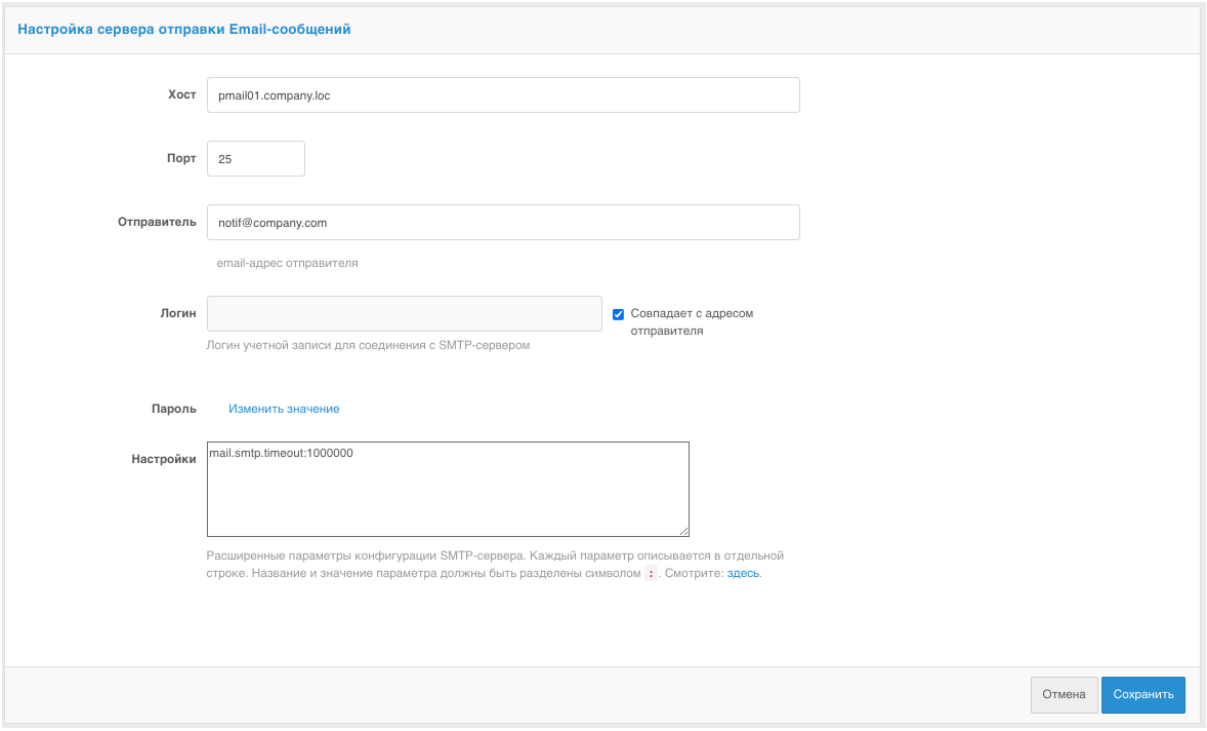

Настройка сервиса отправки Push-уведомлений
Протокол доставки
HTTP-POST
Протокол доставки сообщений
При формировании URL, тела и заголовков HTTP-запроса используйте строки подстановки:
${login} - логин для доступа к сервису
${password} - пароль для доступа к сервису
${message} - текст сообщения (обязательный параметр)
${title} - заголовок сообщения (обязательный параметр)
${subscriberId} - идентификатор пользователя push (обязательный параметр)
URL
http://api.system.ru/json/v1.0/communication/mobile/push
Данные
{"token":"${password}","title":"${title}","body":"${message}","msisdn":${subscriberId}}
Данные передаваемые в теле HTTP-запроса
Логин
test
Логин для доступа к сервису отправки сообщений
Пароль
Изменить значение
Использовать Basic HTTP аутентификацию
Заголовки
Заголовки HTTP-запроса. Каждый заголовок описывается в отдельной строке. Название и значение заголовка должны быть разделены символом :.
Шаблон ответа успешной отправки
.+\"errorCode\":0.+
Регулярное выражение, определяющее успешную отправку сообщения. Например, ^OK.+
Шаблон ответа при ошибке
.+\"errorCode\":[1-9].+
Регулярное выражение, определяющее наличие ошибки при отправке сообщения. Например, ^ERROR.+
Отмена
Сохранить

Подключение к SMTP-шлюзу

В Blitz Identity Provider необходимо настроить возможность отправлять email-сообщения, если используются следующие функции:

- информирование о важных событиях безопасности по email;
- изменение адреса электронной подписи через *«Профиль пользователя»*;
- восстановление забытого пароля с использованием email как канала подтверждения владения учетной записью;
- подтверждение адреса электронной почты при регистрации учетной записи пользователя.

Настройки задаются в консоли управления Blitz Identity Provider в разделе *«Сообщения»*.



Необходимо задать следующие настройки:

- имя хоста SMTP-шлюза;
- порт хоста SMTP-шлюза;
- необходимо или нет использовать TLS для защищенного подключения к шлюзу;
- email отправителя сообщений;
- логин учетной записи на SMTP-шлюзе, от имени которой Blitz Identity Provider будет производить отправку email (если логин совпадает с email отправителя, то следует отметить соответствующий чекбокс);
- пароль от учетной записи на SMTP-шлюзе, от имени которой Blitz Identity Provider будет производить отправку email;
- настройки – дополнительные параметры конфигурации взаимодействия с SMTP-шлюзом[55].

[55] https://javaee.github.io/javamail/docs/api/com/sun/mail/smtp/package-summary.html

2.3 Доступ в приложения и сетевые службы

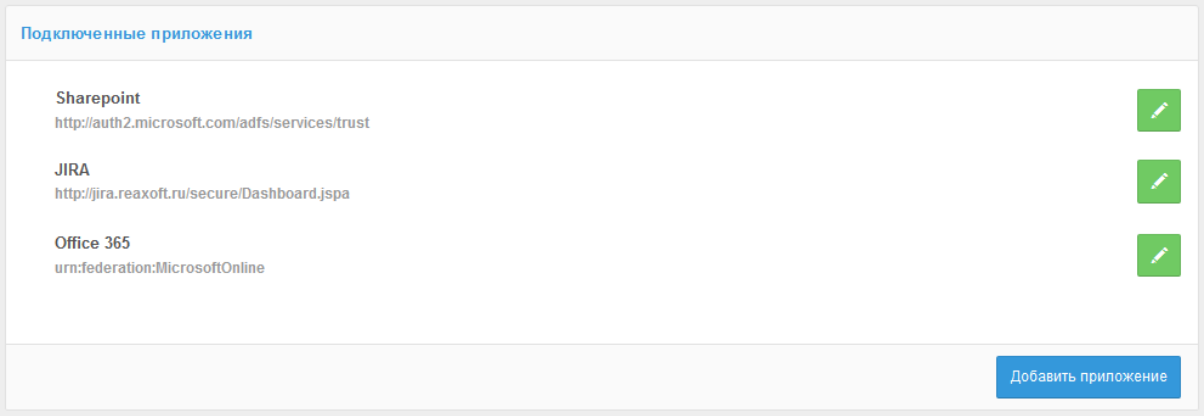
2.3.1 Регистрация приложений в Blitz Identity Provider

О приложениях

Регистрация приложений в Blitz Identity Provider необходима для того, чтобы приложения могли использовать предоставляемые Blitz Identity Provider сервисы:

- запрашивать идентификацию и аутентификацию пользователей;
- вызывать REST-сервисы Blitz Identity Provider.

Управление приложениями осуществляется в разделе Приложения консоли управления.



Создание учетной записи нового приложения

Для подключения нового веб-приложения необходимо перейти в раздел Приложения консоли и выбрать пункт Добавить приложение. Это действие запустит мастер подключения нового приложения, работа которого включает в себя следующие шаги:

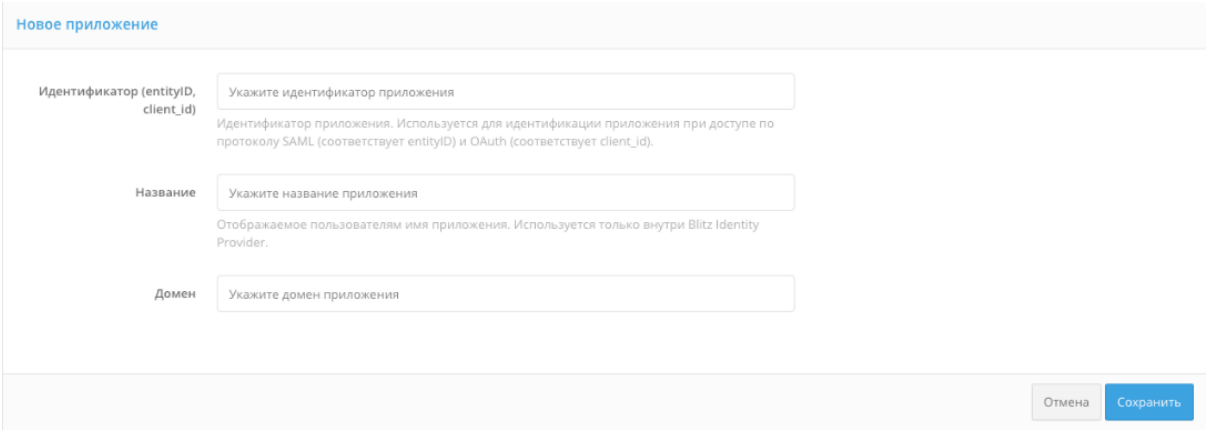
Шаг 1. Базовые настройки

Требуется указать идентификатор подключаемого приложения (при подключении по протоколу SAML идентификатор соответствует `entityID`, при подключении по OAuth 2.0 – `client_id`), его название и домен, т.е. URL, по которому доступно данное приложение.

Важно: При задании идентификатора для OAuth 2.0 недопустимо использовать двоеточие и тильду.

Название приложения используется в дальнейшем в Blitz Identity Provider при отображении на странице входа в случае инициирования приложением запроса на идентификацию пользователя.

Домен приложения используется при необходимости перенаправления пользователя в приложение из веб-страниц Blitz Identity Provider. Перенаправление осуществляется на указанный домен или на переданный в процессе взаимодействия с Blitz Identity Provider специализированный `redirect_uri`, но при этом выполняется сверка, что `redirect_uri` соответствует заданному в настройке приложения домену.



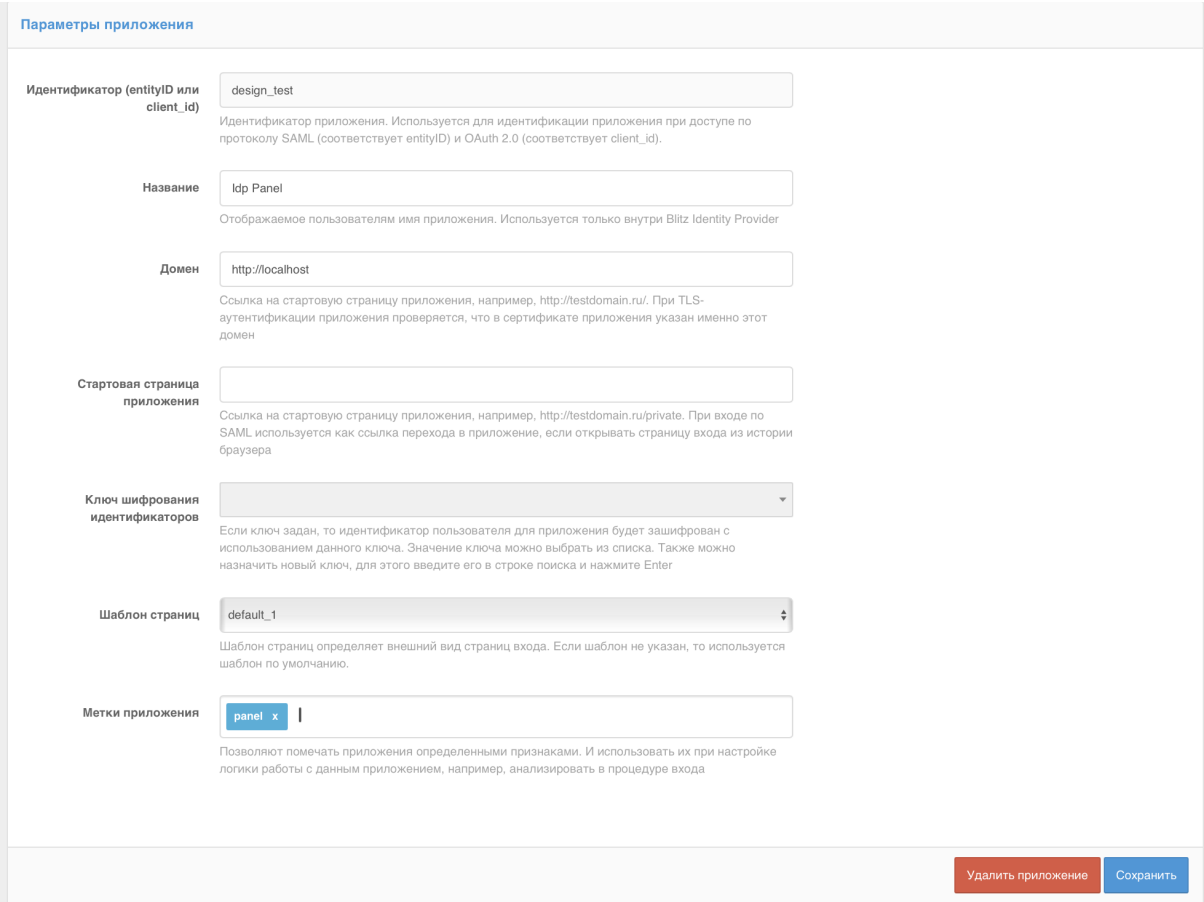
Шаг 2. Задание стартовой страницы приложения и выбор шаблона страницы входа

В поле `Стартовая страница приложения` рекомендуется задать ссылку на вход в приложение, инициирующую запрос идентификации и аутентификации.

В списке `Шаблон страниц` необходимо выбрать, на основе какого шаблона должна отображаться страница входа при попытке доступа пользователя в данное приложение. Инструкция по созданию нового шаблона входа *приведена здесь* (страница 290).

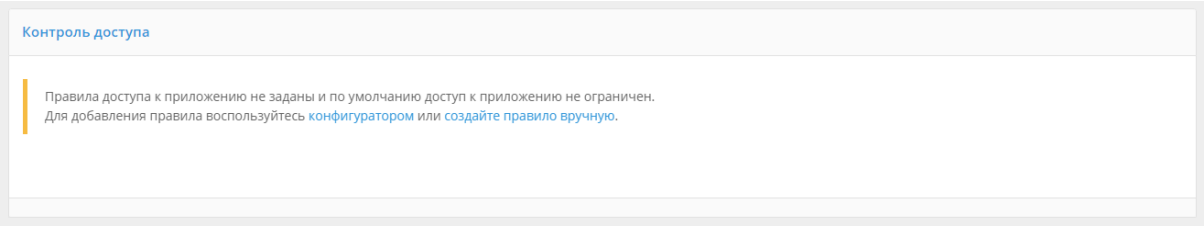
При необходимости можно указать ключ шифрования идентификаторов (`домен приватности`). Создание домена приватности обеспечивает уникальность идентификатора пользователя, полученного приложением по результатам аутентификации, т.е. этот идентификатор будет уникальным, но специфичным для данного приложения. Иными словами, если запрос на получение данных пользователя будет инициировать приложение из другого домена приватности, то оно будет получать другое значение идентификатора пользователя. При нажатии на поле будут отображены сконфигурированные ранее ключи шифрования, с возможность задать новый. Приложения, имеющие общий ключ шифрования, будут получать идентичный идентификатор пользователя.

На данном шаге вы также можете задать метки, чтобы далее использовать их при настройке логики работы с приложением, например, анализировать в *процедуре входа* (страница 284).

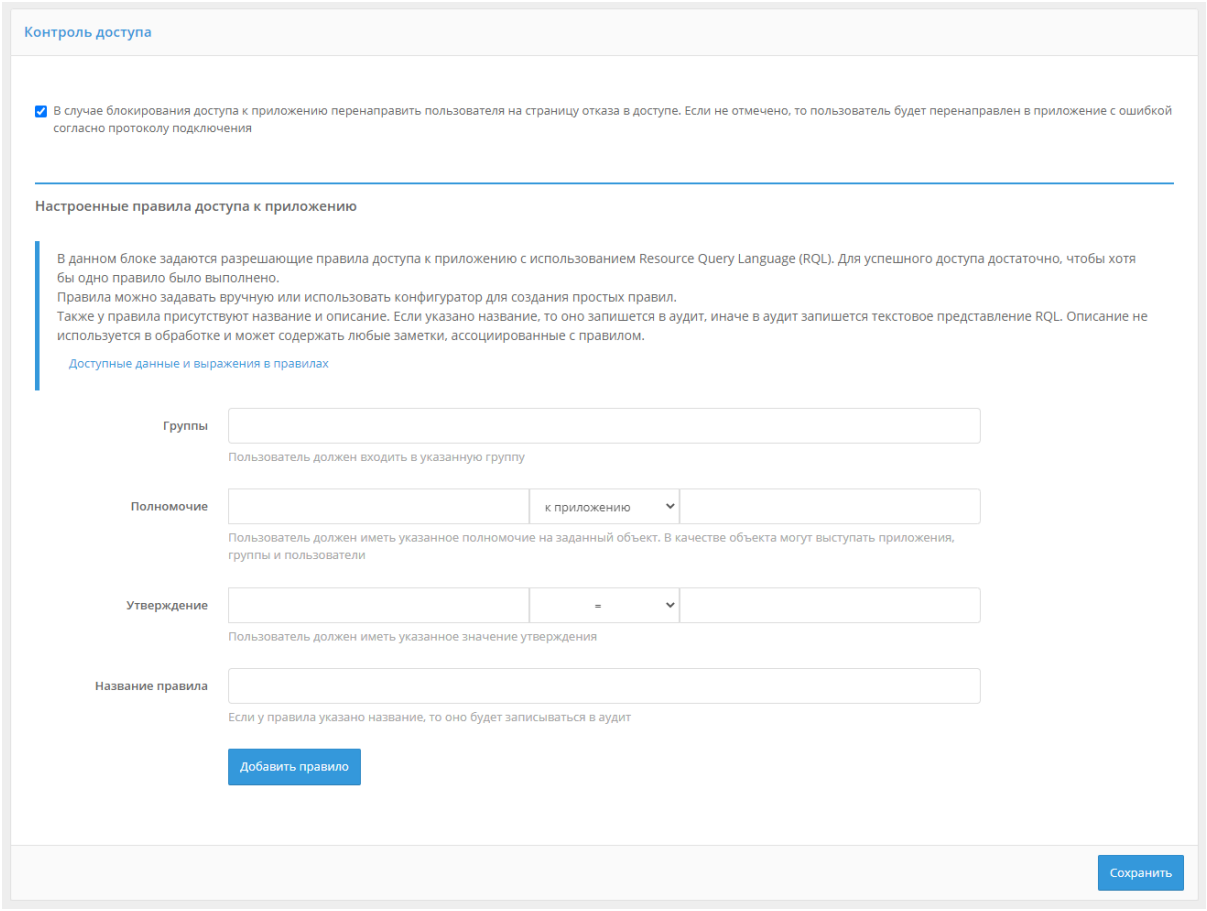


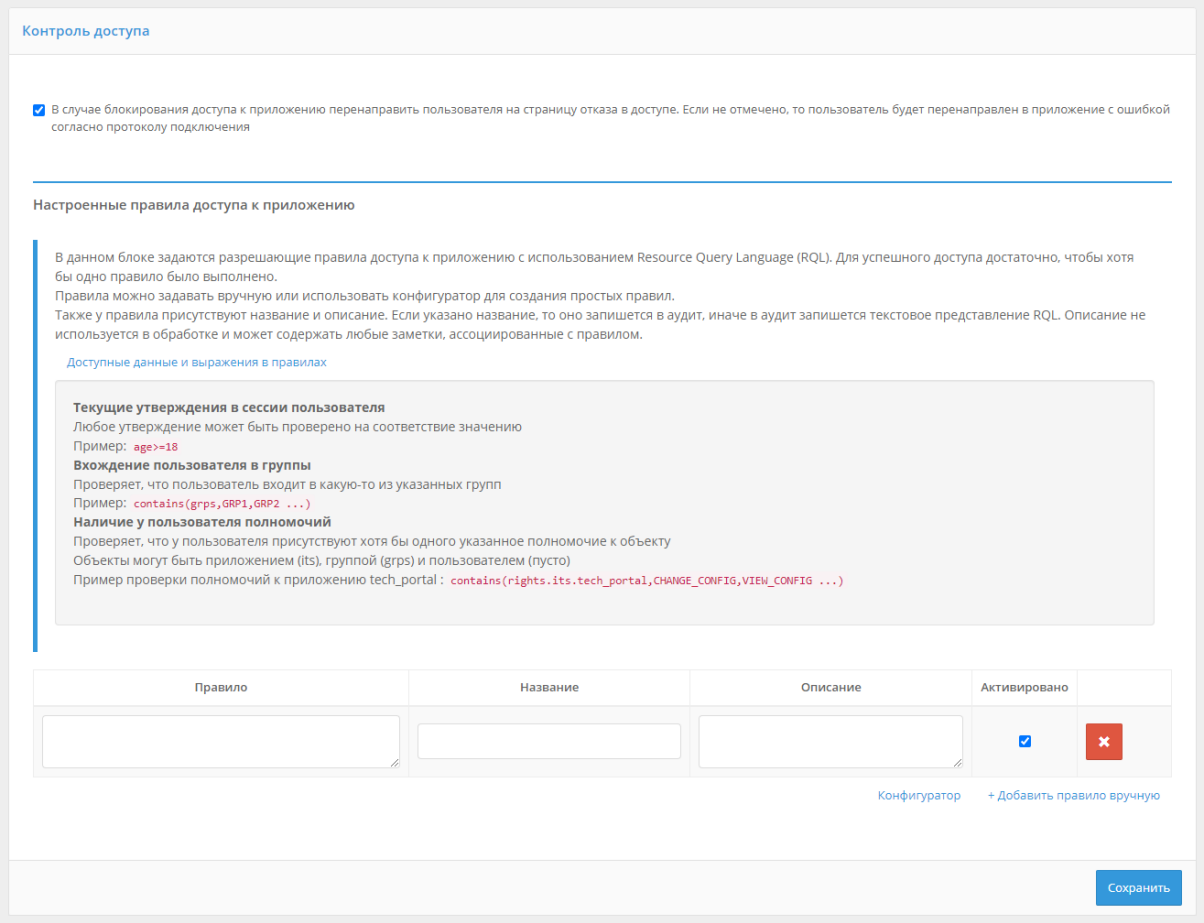
Шаг 3. Настройка правил доступа в приложения

Можно настроить правила, на основе которых Blitz Identity Provider будет принимать решение, впускать или нет пользователя в приложение.



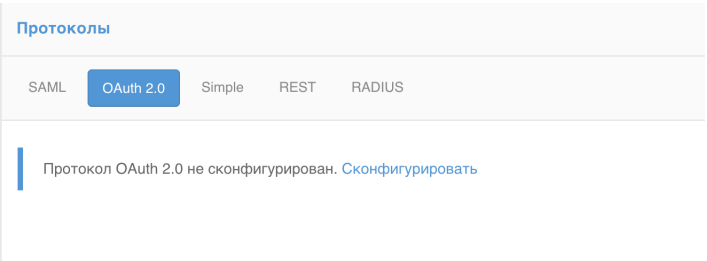
Правила контроля доступа можно добавить с помощью конфигуратора или вручную с помощью RQL-выражения (см. рисунки ниже). В правилах можно проверять, что пользователь включен в нужную группу пользователей (настройка `Группы` в конфигураторе или правило `contains(grps,GRP1,GRP2,…)`), имеет требуемое право доступа (настройка `Полномочие` в конфигураторе или правило `contains(rights.its.SYSTEM,RIGHT_1,RIGHT2,…)`) или имеет указанное значение атрибута (настройка `Утверждение` в конфигураторе или выражение с атрибутом).


Контроль доступа
В случае блокирования доступа к приложению перенаправить пользователя на страницу отказа в доступе. Если не отмечено, то пользователь будет перенаправлен в приложение с ошибкой согласно протоколу подключения
Настроенные правила доступа к приложению
В данном блоке задаются разрешающие правила доступа к приложению с использованием Resource Query Language (RQL). Для успешного доступа достаточно, чтобы хотя бы одно правило было выполнено.
Правила можно задавать вручную или использовать конфигуратор для создания простых правил.
Также у правила присутствуют название и описание. Если указано название, то оно запишется в аудит, иначе в аудит запишется текстовое представление RQL. Описание не используется в обработке и может содержать любые заметки, ассоциированные с правилом.
Доступные данные и выражения в правилах
Группы
Пользователь должен входить в указанную группу
Полномочие
к приложению
Пользователь должен иметь указанное полномочие на заданный объект. В качестве объекта могут выступать приложения, группы и пользователи
Утверждение
=
Пользователь должен иметь указанное значение утверждения
Название правила
Если у правила указано название, то оно будет записываться в аудит
Добавить правило
Сохранить



Шаг 4. Настройки протоколов подключения

Необходимо настроить один или несколько протоколов подключения приложения к Blitz Identity Provider.



Поддерживаются следующие протоколы подключения:

- SAML – для подключения приложений по SAML 1.0, 1.1, 2.0 и WS-Federation для идентификации и аутентификации пользователей.
- OAuth 2.0 – для подключения приложений по OAuth 2.0, OpenID Connect 1.0 (OIDC) для идентификации и аутентификации пользователей. В рамках этого протокола возможно конфигурирование динамической регистрации клиентов.
- Simple – для подключения веб-приложений для осуществления идентификации и аутентификации с помощью подстановки в приложение логина и пароля с proxy-сервера, если приложение не поддерживает возможности подключения по SAML/OIDC.
- REST – для подключения приложений, использующих REST-сервисы Blitz Identity Provider по регистрации/изменению учетных записей, управлению устройствами аутентификации пользователей.
- RADIUS – для подключения к сетевым службам по протоколу RADIUS.

Если организация планирует разработку или доработку собственных приложений для подключения их к Blitz Identity Provider, то разработчикам необходимо ознакомиться с *Руководством по интеграции* (страница 383).

Если организация планирует подключить к Blitz Identity Provider приложения, имеющие штатную поддержку подключения по SAML 1.0, SAML 1.1, SAML 2.0, WS-Federation или OIDC (OpenID Connect 1.0, OAuth 2.0), то в последующих подразделах, описываются общие настройки на стороне Blitz Identity Provider подключения произвольного приложения с поддержкой SAML/OIDC.

2.3.2 Схемы работы технологий SSO

В данном разделе приведены схемы работы таких распространенных технологий единого входа, как OAuth 2.0 и SAML.

Подключение веб-приложения по OIDC

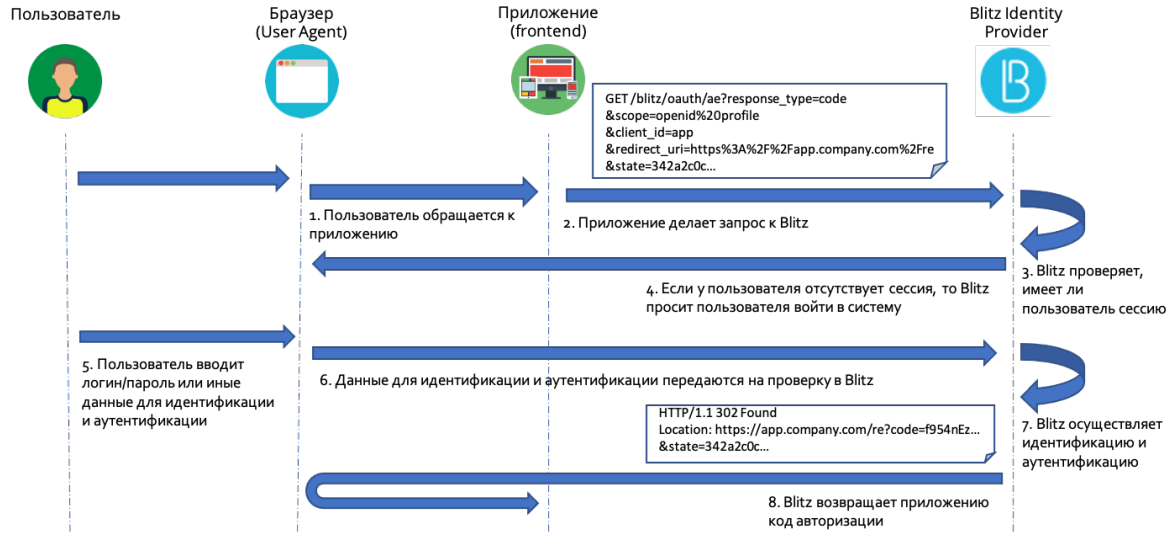
Взаимодействие веб-приложения с Blitz Identity Provider по OIDC включает в себя следующие этапы:

> **Примечание:** Данный процесс совпадает с моделью авторизации приложения Authorization Code Grant, предусмотренной спецификацией OAuth 2.0.

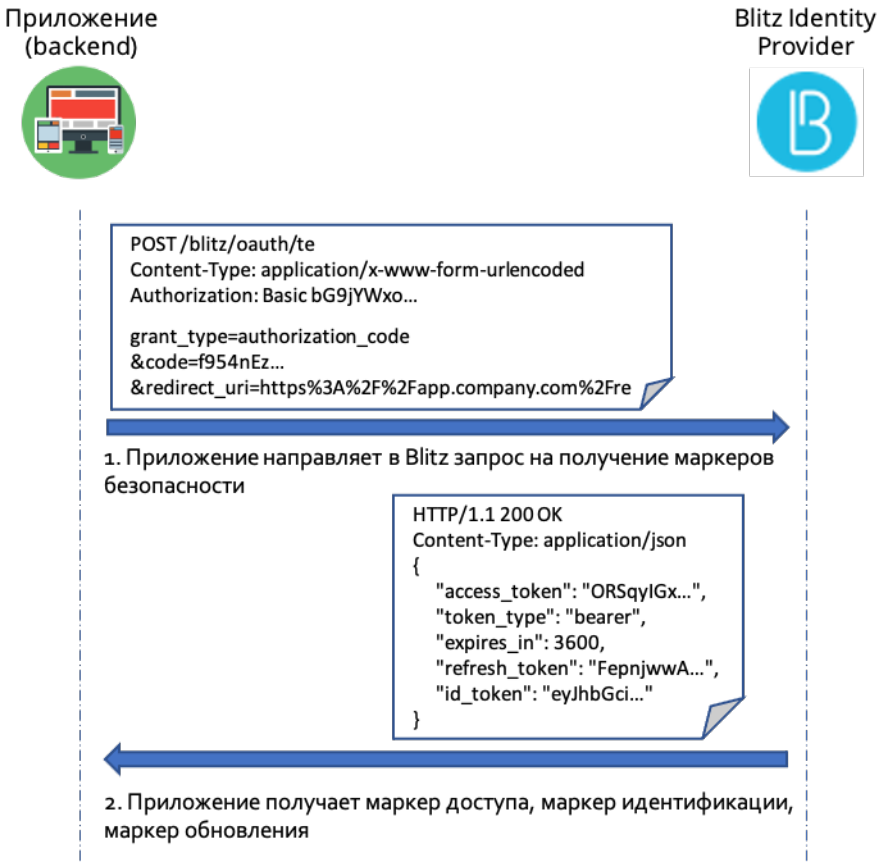
1. Приложение через веб-браузер отправляет запрос на идентификацию и аутентификацию пользователя в адрес Blitz Identity Provider.
2. Blitz Identity Provider проводит идентификацию/аутентифицирует пользователя.
3. Blitz Identity Provider получает согласие пользователя на передачу информации о нем в приложение (для приложений, которые размещены на домене `company.com`, согласие предоставляется автоматически без запроса пользователя).
4. Blitz Identity Provider через веб-браузер перенаправляет пользователя обратно в приложение и передает в приложение код авторизации.
5. Приложение с использованием кода авторизации формирует запрос на получение маркера идентификации, маркера обновления, маркера доступа.
6. Приложение получает ответ, содержащий необходимые маркеры.
7. Приложение запрашивает данные пользователя по маркеру доступа. При необходимости приложение может провести проверку маркера идентификации и извлечь из этого маркера идентификатор пользователя и дополнительные атрибуты.

На рисунках представлены процессы получения кода авторизации, маркеров, данных пользователя.

Получение кода авторизации:



Получение маркеров безопасности:



Получение данных пользователя:

Подключение мобильного приложения по OIDC

Взаимодействие мобильного приложения с Blitz Identity Provider дополнительно к штатным средствам протокола OIDC/OAuth 2.0 использует спецификации:

- RFC 7591 OAuth 2.0 Dynamic Client Registration Protocol[56],
- RFC 7592 OAuth 2.0 Dynamic Client Registration Management Protocol[57].

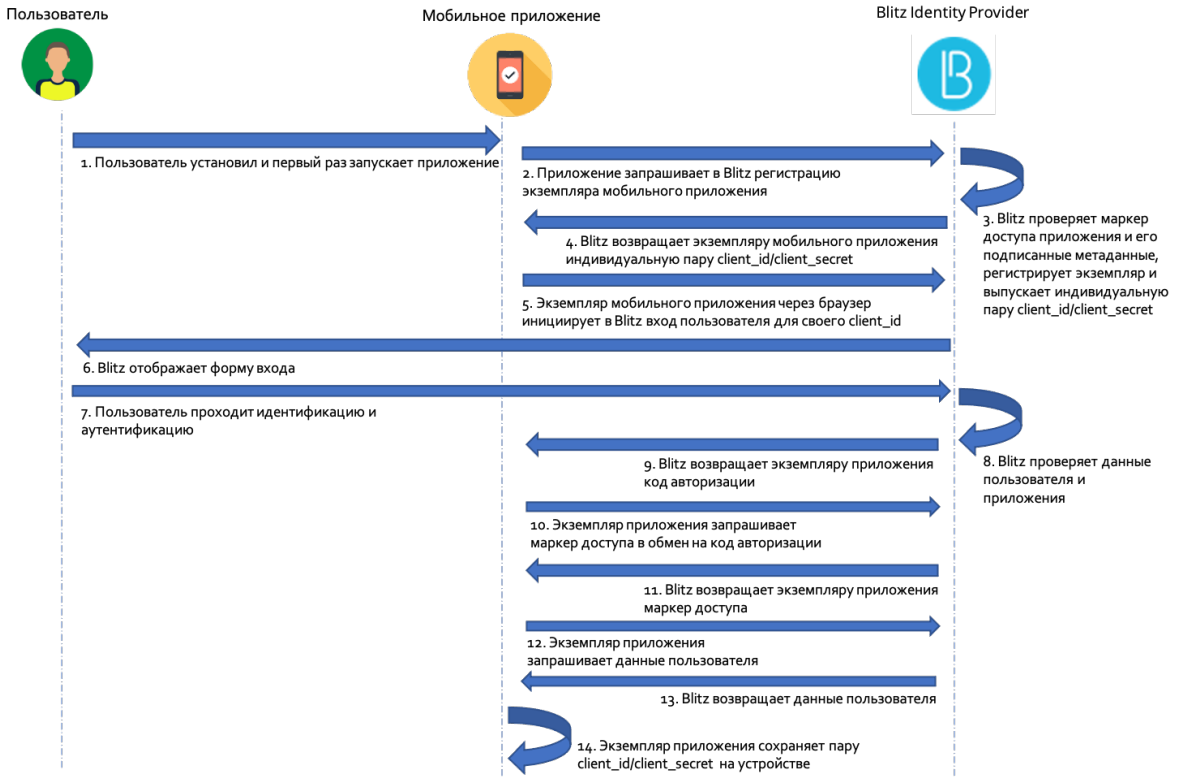
Взаимодействие мобильного приложения с Blitz Identity Provider включает следующие этапы:

1. Динамическая регистрация в Blitz Identity Provider экземпляра мобильного приложения. Получение экземпляром приложения от Blitz Identity Provider уникальной пары `client_id` / `client_secret`.
2. Первичный вход пользователя в мобильное приложение с помощью Blitz Identity Provider. Установка пользователем ПИН-кода или Touch ID/Face ID. Сохранение на устройстве шифрованной пары `client_id` / `client_secret`, полученной от Blitz Identity Provider.
3. Вторичные входы пользователя с помощью ПИН-кода или Touch ID/Face ID. Авторизация в Blitz Identity Provider с помощью шифрованной пары `client_id` / `client_secret`.
4. Удаление в Blitz Identity Provider полученной пары `client_id` / `client_secret` при логауте (смене учетной записи, выхода из учетной записи) пользователя из мобильного приложения.

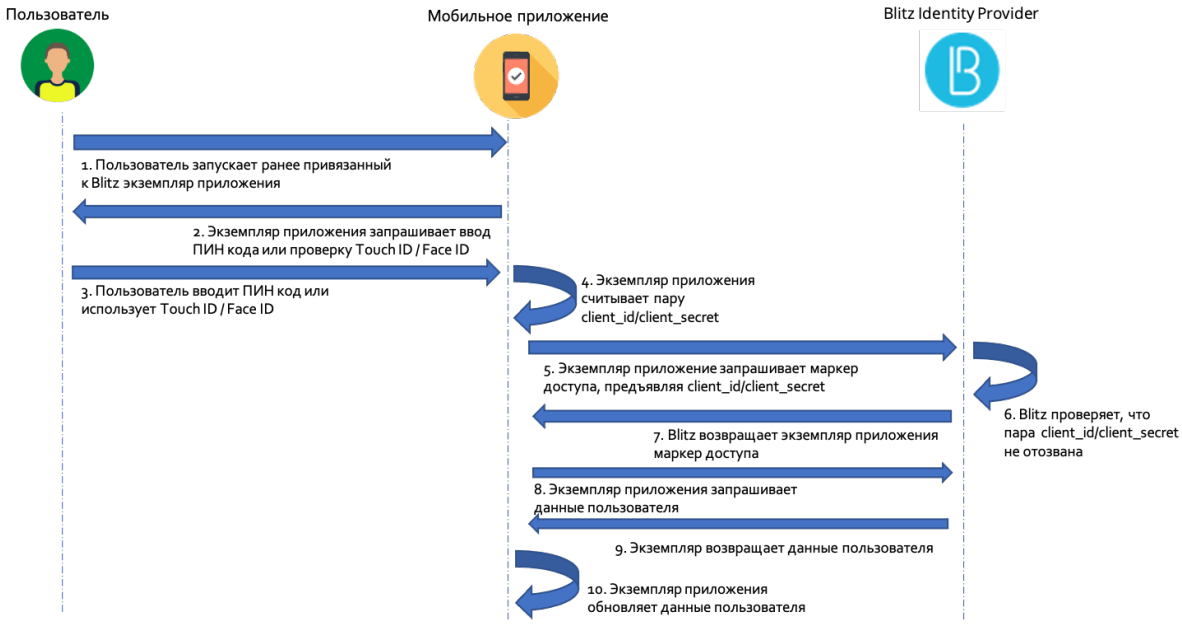
Схематично последовательность действий этапов 1-2 представлена на первом рисунке, а этапа 3 – на втором.

[56] https://tools.ietf.org/html/rfc7591
[57] https://tools.ietf.org/html/rfc7592

Первый вход пользователя в мобильное приложение:



Повторные входы пользователя в мобильное приложение:



Подключение приложения по SAML

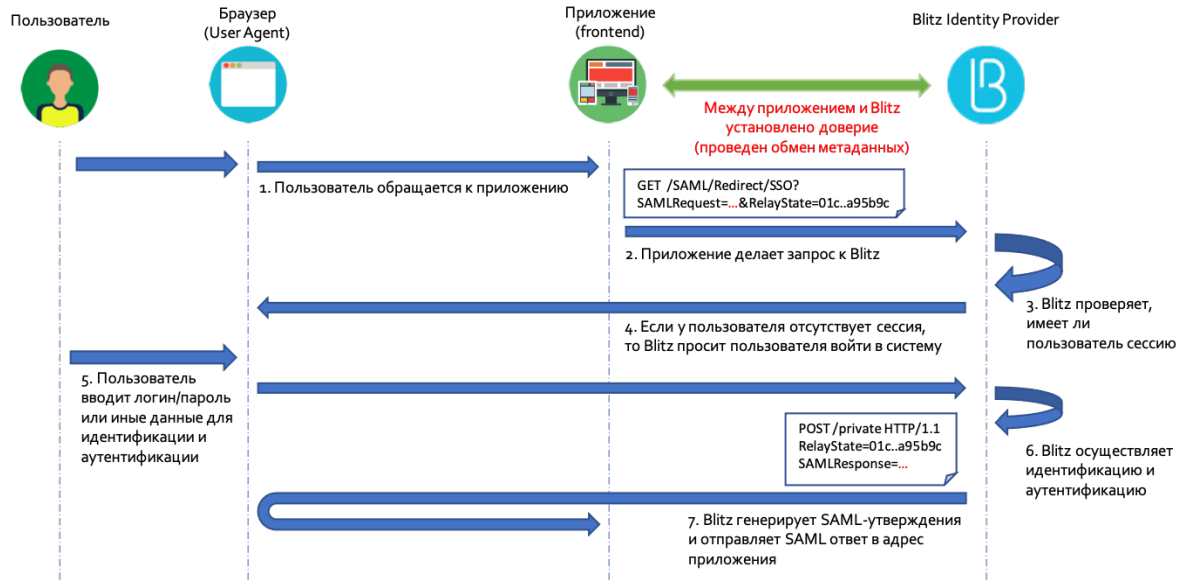
В процессе взаимодействия приложение (поставщик услуг) посылает в Blitz Identity Provider SAML-запрос на идентификацию пользователя (SAML Request). Запрос представляет собой оформленный в соответствии со стандартом SAML XML-документ. В запросе присутствует идентификатор запрашивающего идентификацию приложения, называемый `entityID`, а также дополнительная служебная информация. Сам запрос передается подписанным электронной подписью приложения. В качестве транспортного протокола для передачи сообщения используется протокол HTTPS, вызов поставщика идентификации осуществляется через HTTP Redirect. Это означает, что запрос от приложения к Blitz Identity Provider осуществляется опосредованно, через браузер пользователя, и прямое сетевое взаимодействие между приложением и Blitz Identity Provider при использовании SAML не требуется.

Получив SAML-запрос на идентификацию, Blitz Identity Provider идентифицирует принадлежность запроса определенному приложению, после чего отображает пользователю веб-страницу единого входа для проведения идентификации и аутентификации пользователя. В случае успешной идентификации и аутентификации пользователя Blitz Identity Provider передает приложению (поставщику услуг) SAML-ответ (SAML Response).В зависимости от заданных настроек взаимодействия запрос может быть подписанным и зашифрованным. Для формирования подписи и для шифрования используются стандарты XML Signature и XML Encryption. В качестве транспортного протокола для передачи сообщения с результатами идентификации используется протокол HTTPS, вызов поставщика услуг осуществляется через HTTP POST.

Получив от Blitz Identity Provider SAML-ответ, приложение проверяет его подпись, выполняет расшифровку, после чего извлекает из SAML-утверждений (SAML Assertions) идентификационные данные пользователя (идентификаторы, атрибуты, полномочия).

Процесс взаимодействия приложения и Blitz Identity Provider с использованием SAML приведен на рисунке.

Идентификация пользователя с использованием SAML

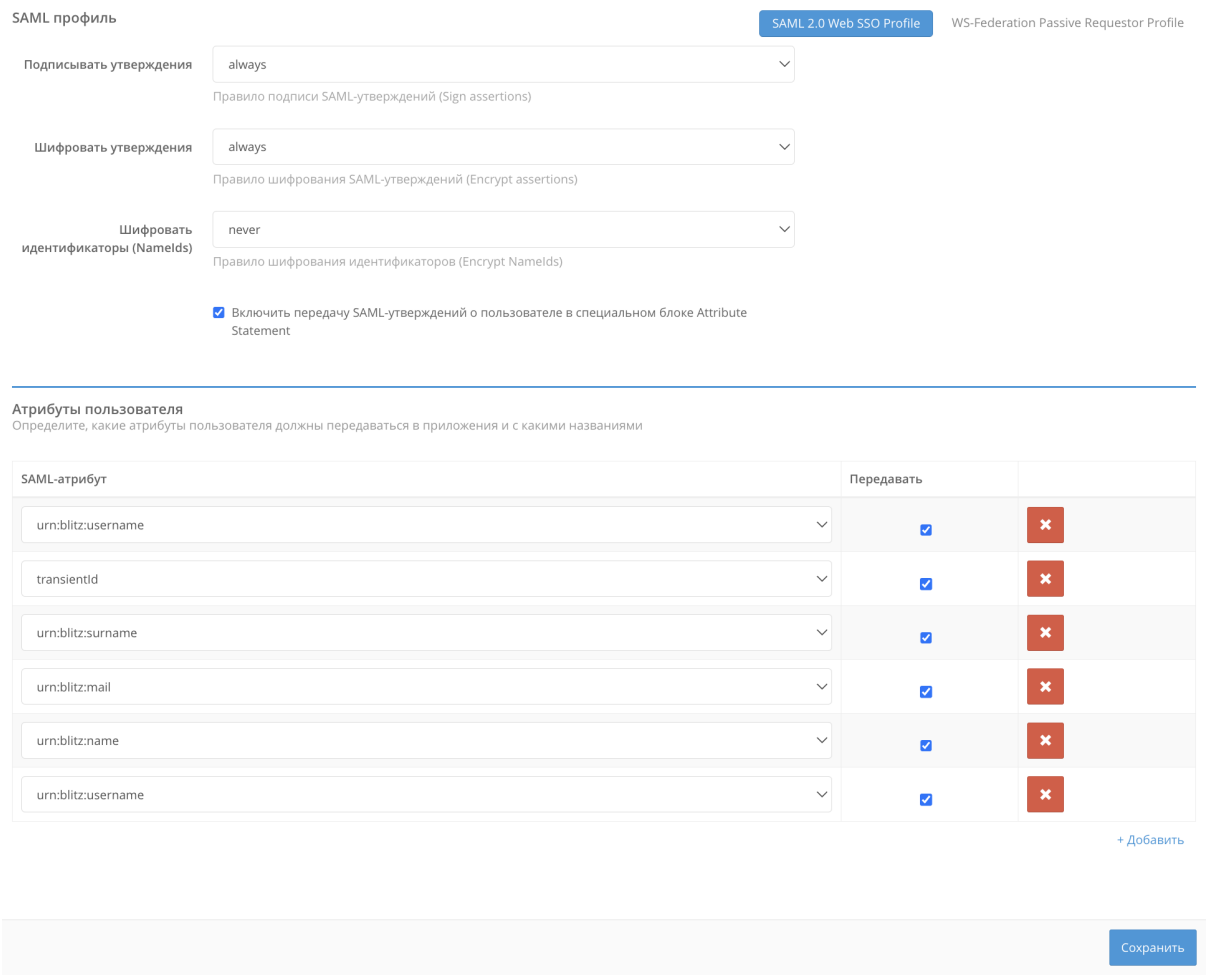


2.3.3 Настройка SAML и WS-Federation

Подключение по SAML 1.0/1.1/2.0

При подключении приложения по SAML необходимо задать следующие настройки:

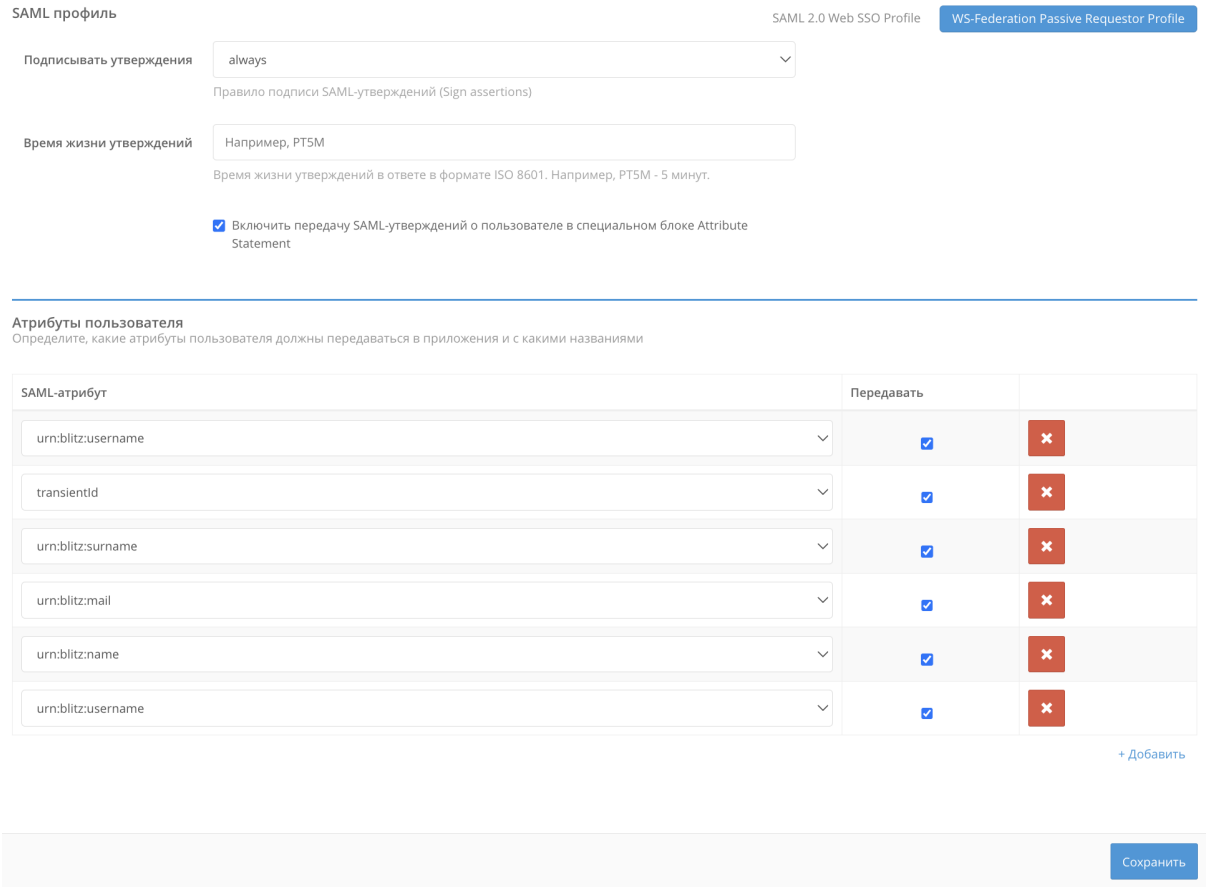
- *загрузить SAML-метаданные подключаемого приложения* (страница 236);
- убедиться, что переключатель SAML-профиля стоит в режиме SAML 2.0 Web SSO Profile;
- в блоке SAML-профиль нажать Сконфигурировать. В появившихся полях указать:
  - указать, нужно ли подписывать SAML-атрибуты (`SAML Assertions`) в ответах Blitz Identity Provider;
  - указать, нужно ли шифровать SAML-атрибуты в ответах Blitz Identity Provider;
  - указать, нужно ли шифровать SAML-идентификаторы (`SAML NameIds`) в ответах Blitz Identity Provider;
  - указать, нужно ли включать в ответ перечень утверждений с атрибутами пользователей;
- указать, какие SAML-атрибуты пользователя из Blitz Identity Provider передавать в приложение. SAML-атрибуты должны быть предварительно *сконфигурированы* (страница 236) в разделе SAML консоли управления.



Подключение по WS-Federation

При подключении приложения по WS-Federation необходимо задать следующие настройки:

- *загрузить SAML-метаданные подключаемого приложения* (страница 236);
- переключатель SAML-профиля установить в режим WS-Federation Passive Requestor Profile;
- в блоке SAML-профиль нажать Сконфигурировать. В появившихся полях указать:
  - указать, нужно ли подписывать утверждения (`Assertions`) в ответах Blitz Identity Provider;
  - указать время жизни утверждений в ответе. Необходимо использовать формат ISO 8601 для указания продолжительности периода[58], например, `PT5M` – 5 минут;
  - указать, нужно ли включать в ответ перечень утверждений с атрибутами пользователей;
- указать, какие атрибуты пользователя из Blitz Identity Provider передавать в приложение. Атрибуты должны быть предварительно *сконфигурированы* (страница 236) в разделе SAML консоли управления.

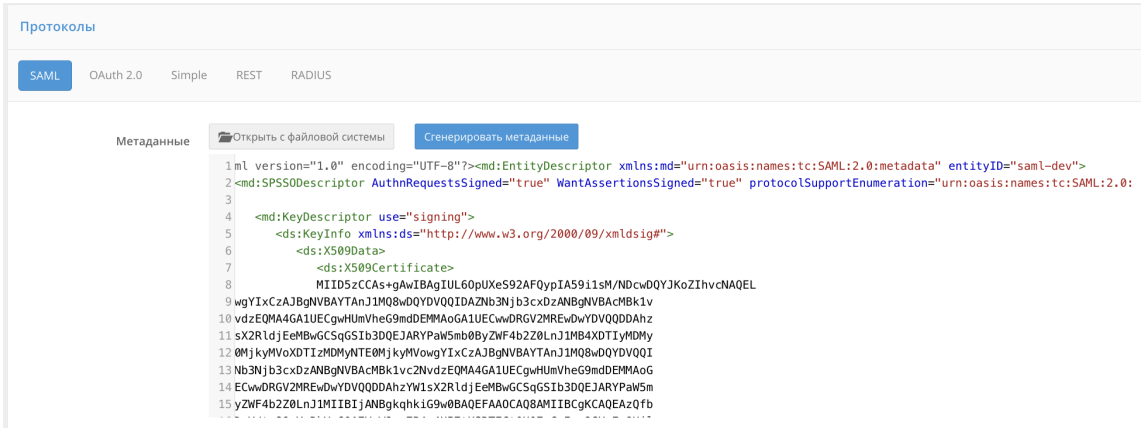


[58] http://www.ifap.ru/library/gost/86012001.pdf

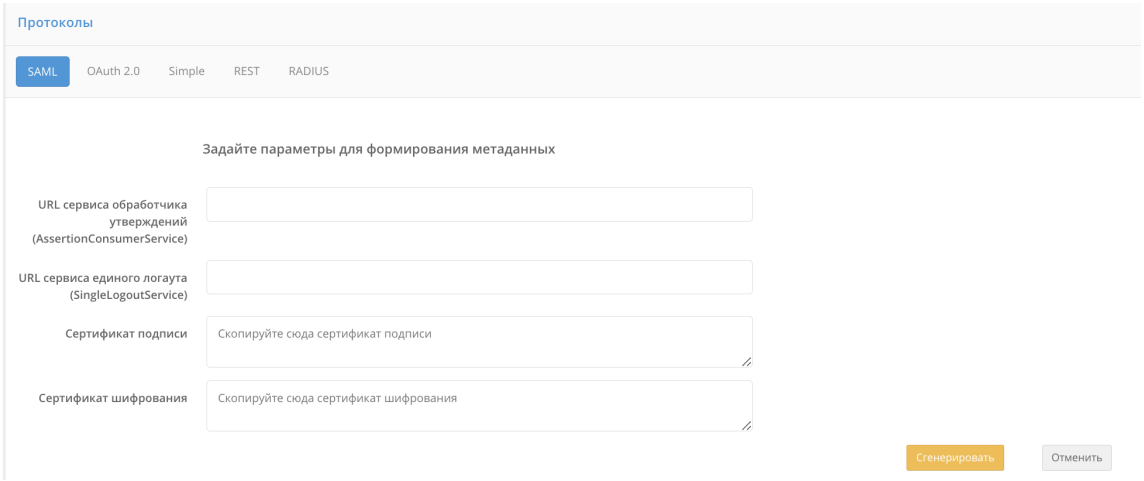
Загрузка SAML-метаданных

Для загрузки SAML-метаданных приложения можно использовать любой из способов:

- Для загрузки готового XML-файла нажмите Открыть с файловой системы.



- Для использования конструктора метаданных нажмите Сгенерировать метаданные. Введите следующие данные:
  - URL сервиса обработчика утверждений (`AssertionConsumerService`),
  - URL сервиса единого логаута (`SingleLogoutService`),
  - Сертификат подписи,
  - Сертификат шифрования.



  Нажмите Сгенерировать. В результате файл метаданных будет автоматически сгенерирован на основании введенных данных.

Настройка SAML-атрибутов

Для регистрации SAML-атрибутов пользователя в Blitz Identity Provider используется раздел SAML консоли управления.

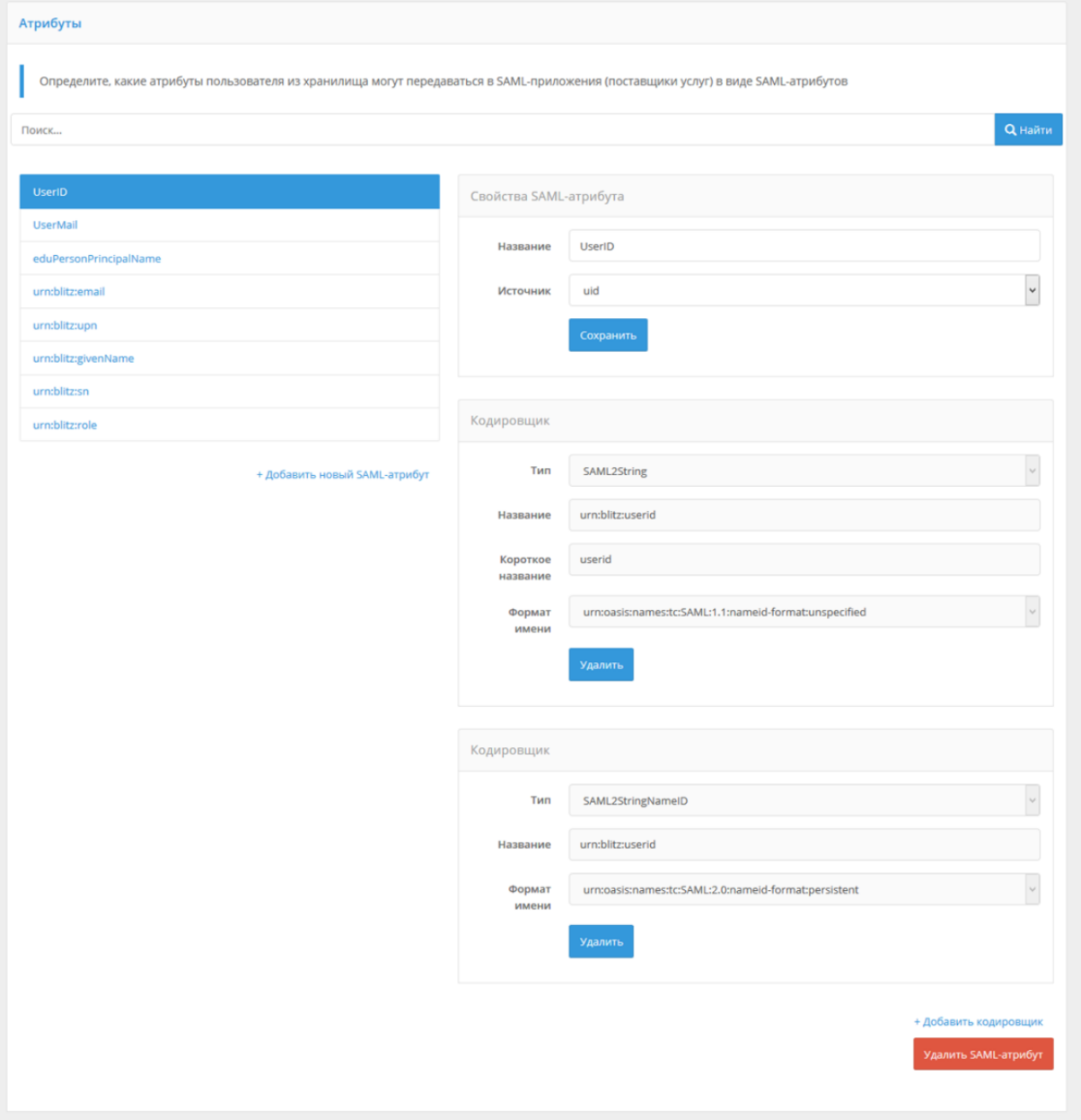
Для добавления нового SAML-атрибута необходимо:

1. Нажать на ссылку Добавить новый SAML-атрибут.
2. Ввести:
   - название SAML-атрибута (именно оно будет отображаться при подключении SAML-приложений);

- источник атрибута (отображаются все атрибуты, определенные в разделе Источники данных).

3. Нажать Добавить. Атрибут будет добавлен.
4. Определить кодировщики атрибутов. Для этого необходимо:
   - нажать на ссылку Добавить кодировщик;
   - выбрать тип кодировщика; следует обратить внимание, что тип кодировщика зависит от версии протокола, с которой работает поставщик услуг (подключенное приложение);
   - название SAML-атрибута, которое будет передано поставщику услуг (в рамках данного типа кодировщика);
   - короткое название, которое будет передано поставщику услуг (в рамках данного типа кодировщика);
   - формат имени.

При необходимости можно определить несколько кодировщиков выбранного SAML-атрибута (для этого каждый кодировщик должен относиться к разным типам кодировщиков).



2.3.4 Настройка OAuth 2.0 и OpenID Connect 1.0

Настройка приложения

При подключении приложения по OAuth 2.0 или OpenID Connect 1.0 (OIDC) в блоке Настройки взаимодействия с приложением необходимо задать следующие настройки взаимодействия с приложением:

- указать секретный ключ (или использовать сгенерированный по умолчанию ключ) подключаемого приложения (`client_secret`), который должен использоваться подключенным приложением при обращении к Blitz Identity Provider (если не указан, то аутентификация приложения-клиента должна производиться иначе, например, с использованием proxy TLS);
- указать дополнительный секретный ключ (`client_secret`) подключаемого приложения. Рекомендуется для случаев, когда нужно обеспечить плавную смену `client_secret` для данного приложения;
- указать предопределенную ссылку возврата (`redirect_uri`) – URL, на который по умолчанию будет переадресован пользователь после прохождения авторизации (`redirect_uri`);
- указать допустимые префиксы ссылок возврата – префикс используется для проверки ссылок возврата (`redirect_uri`), переданных в запросах на идентификацию от приложений. Если в запросе на аутентификацию указана ссылка возврата и она не соответствует ни одному из указанных префиксов, то в аутентификации будет отказано;
- допустимые разрешения – разрешения (`scope`), которые имеет право запрашивать данное приложение;

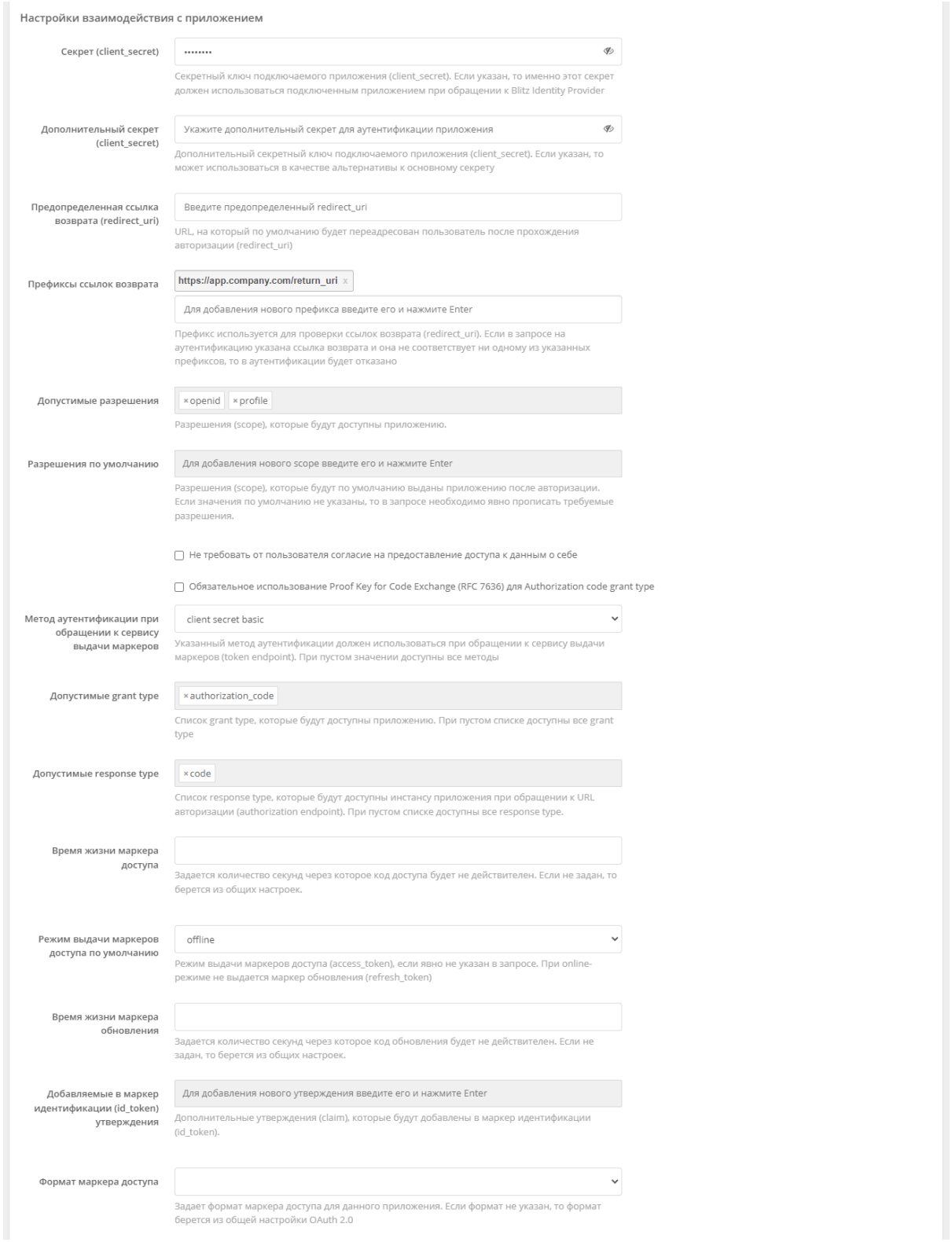
> **Примечание:** Blitz Identity Provider можно *настроить* (страница 143) таким образом, что будут сохраняться маркеры доступа пользователя от внешних поставщиков идентификации. Если приложение будет *получать по REST API* (страница 477) сохраненные маркеры доступа, для него необходимо выбрать системные разрешения `fed_tkn_any` (все внешние поставщики) и `fed_tkn_${fedPointType}_${fedPointName}` (внешний поставщик с типом `${fedPointType}` и именем `${fedPointName}`). Данные разрешения должны быть предварительно заведены в общих настройках протокола в разделе OAuth консоли.

- разрешения по умолчанию – разрешения (`scope`), которые будут по умолчанию выданы приложению после аутентификации. Если не указаны, то в запросе на аутентификацию всегда должны быть явно прописаны требуемые разрешения;
- отметить при необходимости опцию *«Не требовать от пользователя согласие на предоставление доступа к данным о себе»*. Если она отмечена, то при первом входе пользователя в систему не будет отображена страница согласия на предоставление данных этой системе;
- отметить при необходимости опцию *«Обязательное использование Proof Key for Code Exchange (RFC 7636) для Authorization code grant type»*, если запросы на аутентификацию должны валидироваться согласно RFC 7636;
- выбрать при необходимости метод аутентификации при обращении к сервису выдачи маркеров. Указанные методы аутентификации должны использоваться при обращении к сервису выдачи маркеров (`token endpoint`). При пустом значении доступны все методы;
- выбрать при необходимости допустимые `grant type`. Параметр определяет список `grant type`, которые будут доступны приложению. При пустом списке доступны все `grant type`;
- выбрать при необходимости допустимые `response type`. Параметр определяет список `response type`, которые будут доступны приложению при обращении к URL авторизации (`authorization endpoint`). При пустом списке доступны все `response type`;
- указать время жизни маркера доступа (в секундах). Если параметр не задан, то берется из общих настроек из раздела «OAuth 2.0»;
- указать режим выдачи маркеров доступа по умолчанию. Blitz Identity Provider предусматривает два режима выдачи маркеров доступа (`access_token`):

  - offline-режим – при запросе маркера доступа будет выдан также бессрочный маркер обновления (`refresh_token`), которые может быть использован для получения нового маркера доступа. Приложению рекомендуется использовать этот режим, если оно должно получать актуальные данные пользователя из Blitz Identity Provider за пределами времени действия пользовательской сессии. Например, если приложение делает почтовую рассылку и перед ее отправкой хочет получить актуальный адрес электронной почты из Blitz Identity Provider.
  - online-режим – будет выдан только маркер доступа. Приложению рекомендуется использовать этот режим, если ему достаточно получать актуальные данные пользователя в момент входа (в течение активной сессии пользователя).

Режим выдачи маркеров доступа может быть явно указан в запросе на проведение аутентификации; если он не указан, то используется режим по умолчанию.

- указать время жизни маркера обновления (в секундах). Если параметр не задан, то берется из общих настроек из раздела *«OAuth 2.0»*;
- указать добавляемые в маркер идентификации (`id_token`) утверждения. Если приложение взаимодействует с Blitz Identity Provider по протоколу OIDC (OpenID Connect 1.0), то в качестве одного из разрешений (`scope`) необходимо также указать `openid`. Тогда в обмен на авторизационный код при вызове `Token Endpoint` будут выданы не только маркер доступа (`access token`) и маркер обновления (`refresh token`), но и идентификационный маркер (`id_token`). В маркер идентификации будет включен идентификатор пользователя `sub`, а также дополнительные атрибуты, перечисленные в этой настройке. Возможно добавление как атрибутов, сконфигурированных в разделе *«Источники данных»*, так и дополнительных атрибутов (подробнее см. *Добавление атрибутов в маркер идентификации* (страница 244));
- выбрать формат маркера доступа – можно выбрать `opaque` или `JWT`. Если параметр не задан, то берется из общих настроек из раздела *«OAuth 2.0»*.

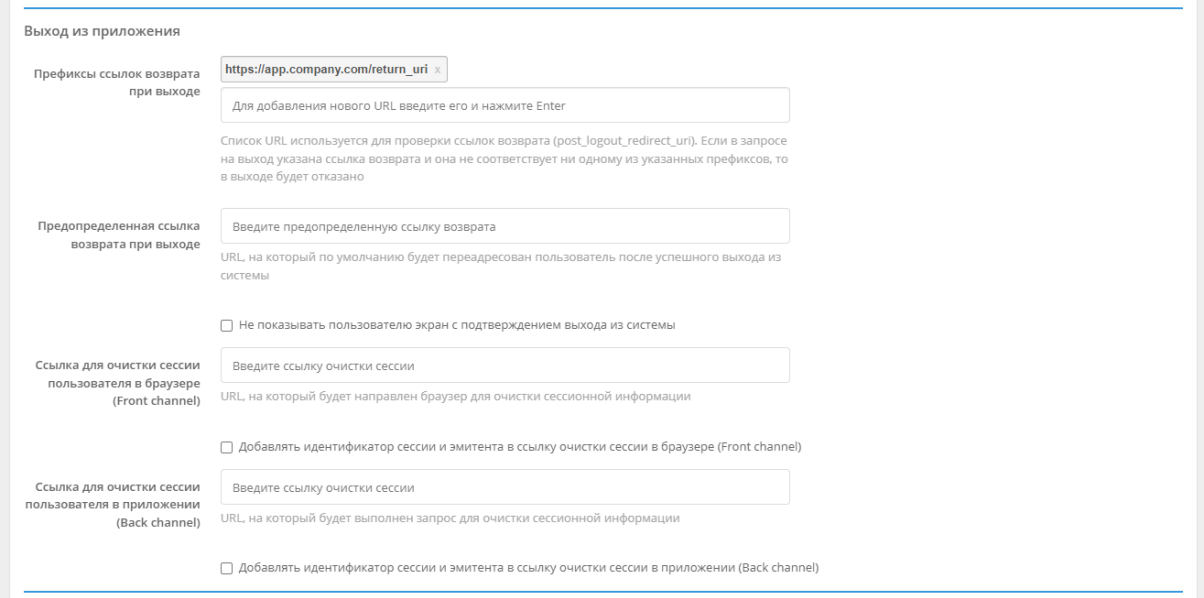


При использовании в приложении функции логаута[59] в блоке *«Выход из приложения»* необходимо задать следующие настройки:

- указать префиксы ссылок возврата при выходе. Необходимо перечислить префиксы допустимых URL страниц перенаправления пользователя после инициирования приложением логаута. Допустимо задать один или несколько префиксов ссылок возврата;

[59] https://openid.net/specs/openid-connect-rpinitiated-1_0.html#RPLogout

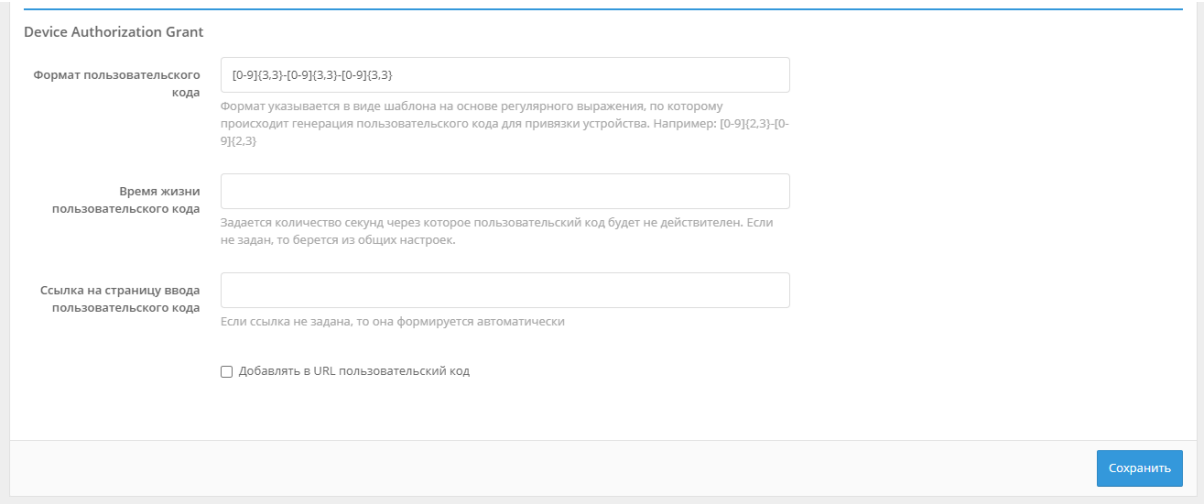
- предопределенная ссылка возврата при выходе – ссылка, на которую будет перенаправлен пользователь после логаута из приложения, если в параметрах вызова логаута от приложения не был передан адрес возврата `post_logout_redirect_uri`;
- отметить при необходимости опцию *«Не показывать пользователю экран с подтверждением выхода из системы»* – если эту настройку не отметить, то пользователю будет показан экран с запросом подтверждения выхода из приложения;
- ссылка для очистки сессии пользователя в браузере (`Front channel`) – указанный адрес обработчика приложения будет вызван из фрейма браузера в случае инициирования логаута пользователя;
- отметить при необходимости опцию *«Добавлять идентификатор сессии и эмитента в ссылку очистки сессии в браузере (Front channel)»* – в этом случае в браузерный обработчик логаута приложения будет передан идентификатор сессии (`sid`);
- ссылка для очистки сессии пользователя в приложении (`Back channel`) – указанный адрес обработчика приложения будет вызван с сервера Blitz Identity Provider в случае инициирования логаута пользователя;
- отметить при необходимости опцию *«Добавлять идентификатор сессии и эмитента в ссылку очистки сессии в приложении (Back channel)»* – в этом случае на адрес обработчика приложения, вызванный с сервера Blitz Identity Provider в случае инициирования логаута пользователя, будет передан `logout_token`, содержащий идентификатор сессии пользователя (`sid`).



При использовании приложением авторизации по спецификации Device Authorization Grant[60] (например, для подключения IOT-устройств, смарт ТВ, чат ботов, приложений голосовых помощников) в блоке *«Настройки взаимодействия с приложением»* в параметре *«Допустимые response type»* добавить вариант `device code`, а в параметре *«Допустимые grant type»* добавить вариант `urn:ietf:params:oauth:grant-type:device_code`. Также в блоке *«Device Authorization Grant»* необходимо задать следующие настройки:

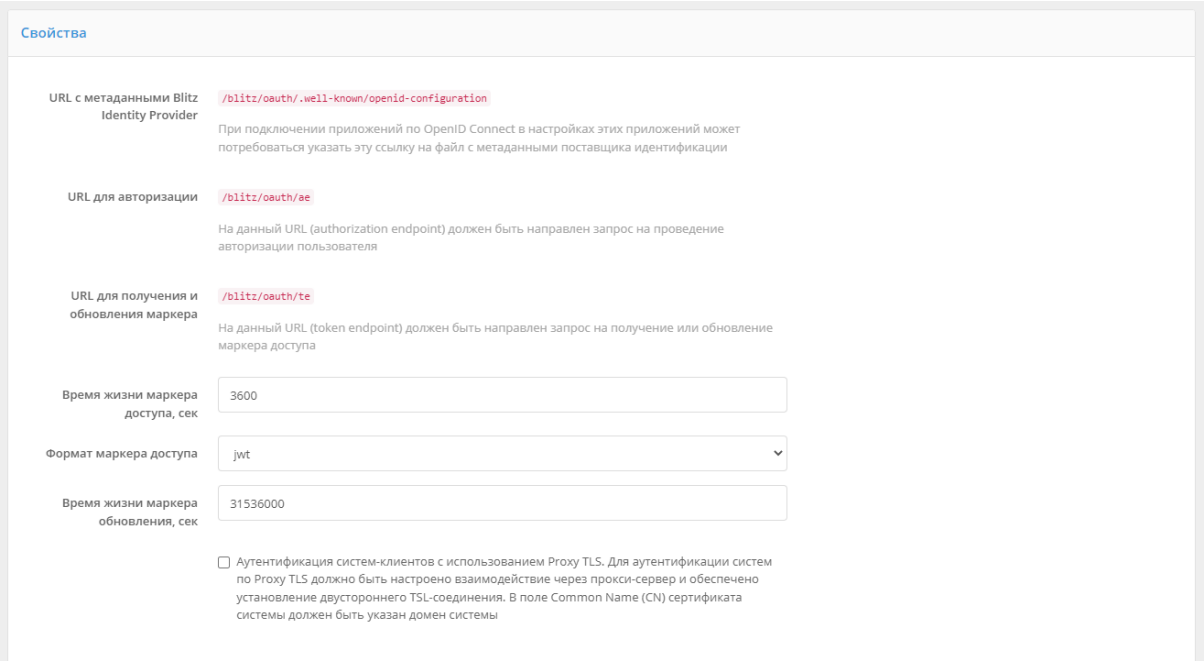
- формат пользовательского кода, для этого следует использовать регулярные выражения;
- время жизни пользовательского кода;
- ссылку на страницу ввода пользовательского кода;
- отметить при необходимости опцию *«Добавлять в URL пользовательский код»*. В этом случае Blitz Identity Provider при авторизации будет возвращать не только ссылку на станицу ввода пользовательского кода (например, `https://test.ru/device`), но еще и ссылку с кодом в качестве параметра (например, `https://test.ru/device?uc=676-267-324`).

[60] https://tools.ietf.org/html/rfc8628



Общие настройки OAuth 2.0

Для задания общих настроек OAuth 2.0, а также для конфигурирования набора разрешений (`scope`) используется раздел *«OAuth 2.0»* консоли управления.



В разделе *«OAuth 2.0»* консоли управления можно посмотреть различные URL обработчиков Blitz Identity Provider, связанных с OAuth 2.0 и OIDC:

- «URL с метаданными Blitz Identity Provider» – по этой ссылке размещены динамически обновляемые настройки (метаданные) Blitz Identity Provider (спецификация[61]). Разработчики приложений могут не прописывать все указанные ниже URL в конфигурации своего приложения, а использовать в настройках единую ссылку на эти метаданные;
- «URL для авторизации» – адрес обработчика OAuth 2.0 Authorization Endpoint для запросов через браузер на получение кода авторизации;
- «URL для получения и обновления маркера» – адрес обработчика OAuth 2.0 Token Endpoint для получения маркеров безопасности (`access_token`, `id_token`, `refresh_token`).

[61] https://tools.ietf.org/html/draft-ietf-oauth-discovery-10

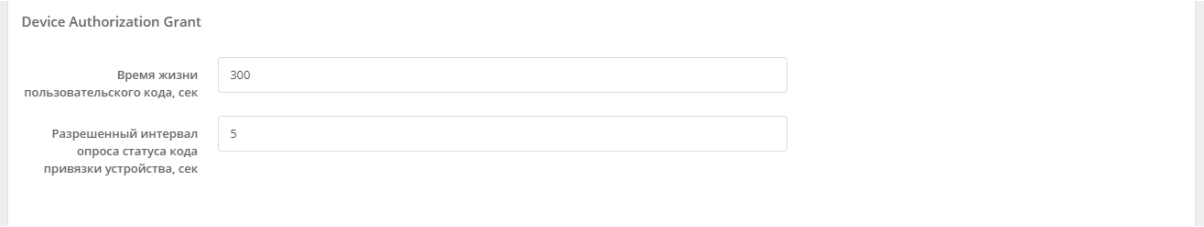
При необходимости можно:

- изменить *«Время жизни маркера доступа»*, используемое по умолчанию при выпуске маркеров для всех приложений;
- указать *«Формат маркера доступа»*, используемый по умолчанию при выпуске маркеров для всех приложений: строка (`opaque`) или JWT;
- изменить *«Время жизни маркера обновления»*, используемое по умолчанию при выпуске маркеров для всех приложений;
- отметить опцию *«Аутентификация систем-клиентов с использованием Proxy TLS»*. В этом случае должно быть настроено взаимодействие приложений с Blitz Identity Provider через прокси-сервер с установкой двустороннего TSL соединения. В поле *«Common Name (CN)»* сертификата системы должен быть указан домен системы подключаемого приложения.

В разделе *«Device Authorization Grant»* можно определить общие настройки для взаимодействия с приложениями по спецификации Device Authorization Grant. Здесь имеется возможность указать:

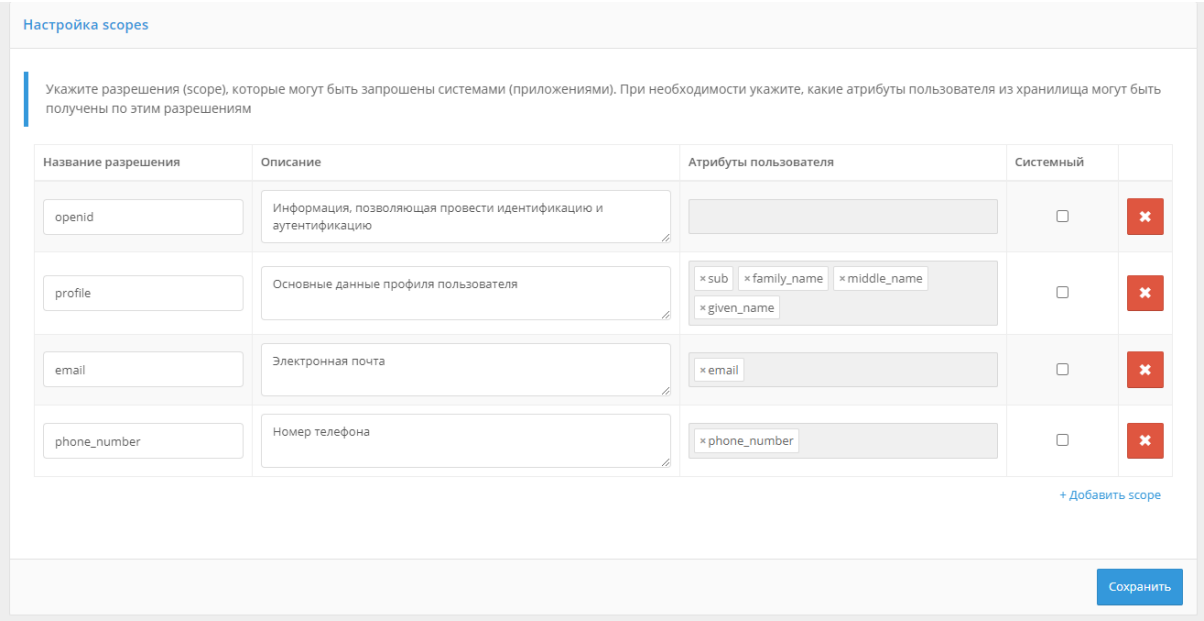
- время жизни пользовательского кода (в секундах);
- минимально разрешенный интервал опроса статуса кода привязки устройства в секундах. Если приложение опрашивает сервис Blitz Identity Provider чаще, чем указано в этом параметре, то будет возвращена ошибка.

При необходимости для каждого приложения можно указать свои настройки, связанные со спецификацией Device Authorization Grant.



Для корректной работы взаимодействия с приложениями по протоколу OAuth 2.0 необходимо определить разрешения (`scope`). Для этого нужно указать:

- название разрешения;
- описание разрешения (оно будет отображаться пользователю на странице согласия на предоставление доступа);
- атрибуты пользователя, которые будут предоставлены по данному разрешению (атрибуты должны быть определены в меню *«Источники данных»*);
- является ли разрешение системным – такие разрешения предоставляются приложениям только с использованием OAuth 2.0 Client Credentials Flow (не в контексте разрешения отдельного пользователя, а общие).



Внимание: Для корректной работы аутентификации по OpenID Connect 1.0 нужно убедиться, что разрешение с названием `openid` определено в этом разделе консоли. Также можно прописать атрибуты, передаваемые по этому разрешению. В этом случае указанные данные могут быть получены по маркеру доступа (`access token`), выданному на разрешение `openid`.

Важно: Blitz Identity Provider можно *настроить* (страница 143) таким образом, что будут сохраняться маркеры доступа пользователя от внешних поставщиков идентификации. Если подключенные по OAuth 2.0 приложения будут *получать по REST API* (страница 477) сохраненные маркеры доступа, в этом разделе консоли необходимо указать системные разрешения `fed_tkn_any` (все внешние поставщики) и `fed_tkn_${fedPointType}_${fedPointName}` (внешний поставщик с типом `${fedPointType}` и именем `${fedPointName}`). Данные разрешения должны также быть указаны в настройках протокола `OAuth` конкретного приложения.

Добавление атрибутов в маркер идентификации

Чтобы иметь возможность определять сессионные утверждения в процедуре входа, соответствующие утверждения также должны быть определены в конфигурационной файле. Для этого в раздел `blitz.prod.local.idp.login` конфигурационного файла необходимо добавить атрибут `sessionClaims` с перечнем утверждений, которые могут быть определены в процедуре.

Например, следующая запись позволяет определить атрибут `custom_attr`:

```
"sessionClaims" : [
   "custom_attr"
]
```

Приложения, подключенные по протоколу OpenID Connect 1.0, могут получать данные в маркере идентификации. Перечень атрибутов, которые будут переданы в маркере идентификации, должен быть задан в пункте *«Добавляемые в маркер идентификации (id_token) утверждения»* настроек протокола.

Помимо хранимых атрибутов, в маркер идентификации могут быть добавлены утверждения:

- полученные при входе пользователя по электронной подписи. Это могут быть данные о сертификате ключа электронной подписи, данные о физическом / юридическом лице из сертификата;

- полученные при входе через ЕСИА;
- определенные в процедуре входа.

Для получения утверждений из сертификата ключа электронной подписи необходимо отредактировать конфигурационный файл `blitz.conf`, добавив в блок настроек `blitz.prod.local.idp.login.methods.x509` добавить структуру следующего содержания:

```
"claims" : [
  {
      "name" : "attr_name",
      "value" : "cert_attr_name"
  }
],
```

В этой структуре `attr_name` – имя атрибута, которое будет использовано в маркере идентификации, а `cert_attr_name` – обозначение атрибута в сертификате (примеры доступных значении приведены в таблице).

Пример данных, получаемых из сертификата ключа электронной подписи

Обозначение атрибута в сертификате	Описание
SUBJECT.OGRN	ОГРН организации
SUBJECT.OGRNIP	ОГРНИП индивидуального предпринимателя
SUBJECT.INN	ИНН организации
SUBJECT.E	Служебный email должностного лица
SUBJECT.O	Имя организации
SUBJECT.ST	Регион организации
SUBJECT.L	Населенный пункт организации
SUBJECT.STREET	Улица, дом, номер офиса организации
SUBJECT.O	Подразделение должностного лица
SUBJECT.T	Должность представителя
SUBJECT.<OID>	Значением из атрибута с указанным `OID`. Например, `SUBJEC T.1.2.643.100.5` позволяет обратиться к атрибуту с `OI D 1.2.643.100.5`

Пример добавляемой в конфигурационный файл структуры:

```
"claims" : [
  {
      "name" : "org_OGRN",
      "value" : "SUBJECT.OGRN"
  },
  {
      "name" : "org_INN",
      "value" : "SUBJECT.INN"
  },
  {
      "name" : "org_email",
      "value" : "SUBJECT.E"
  },
  {
      "name" : "org_name",
      "value" : "SUBJECT.O"
  }
],
```

Чтобы утверждения из ЕСИА были доступны, необходимо отредактировать конфигурационный файл

blitz.conf, добавив в блок настроек blitz.prod.local.idp.federation.points.esia добавить структуру следующего содержания:

```
"claims" : [
  {
      "name" : "attr_name",
      "value" : "esia_attr_name"
  }
],
```

В этой структуре attr_name – имя атрибута, которое будет использовано в маркере идентификации, а esia_attr_name – обозначение атрибута при получении его из ЕСИА.

Пример данных, получаемых из ЕСИА

Обозначение атрибута, полученного из ЕСИА	Описание
oid	Уникальный идентификатор учетной записи ЕСИА
lastName	Фамилия
firstName	Имя
middleName	Отчество
birthDate	Дата рождения
gender	Пол
snils	СНИЛС
inn	ИНН
passport	Паспортные данные
birthPlace	Место рождения
email	Эл. почта
mobile	Моб. телефон

Пример добавляемой в конфигурационный файл структуры:

```
"claims" : [
                {
                    "name" : "esia_firstName",
                    "value" : "firstName"
                },
                {
                    "name" : "esia_lastName",
                    "value" : "lastName"
                },
                {
                    "name" : "esia_middleName",
                    "value" : "middleName"
                },
                {
                    "name" : "esia_birthDate",
                    "value" : "birthDate"
                }
                {
                    "name" : "esia_gender",
                    "value" : "gender"
                }
                {
                    "name" : "esia_snils",
                    "value" : "snils"
                }
                {
                    "name" : "esia_inn",
```

(continues on next page)

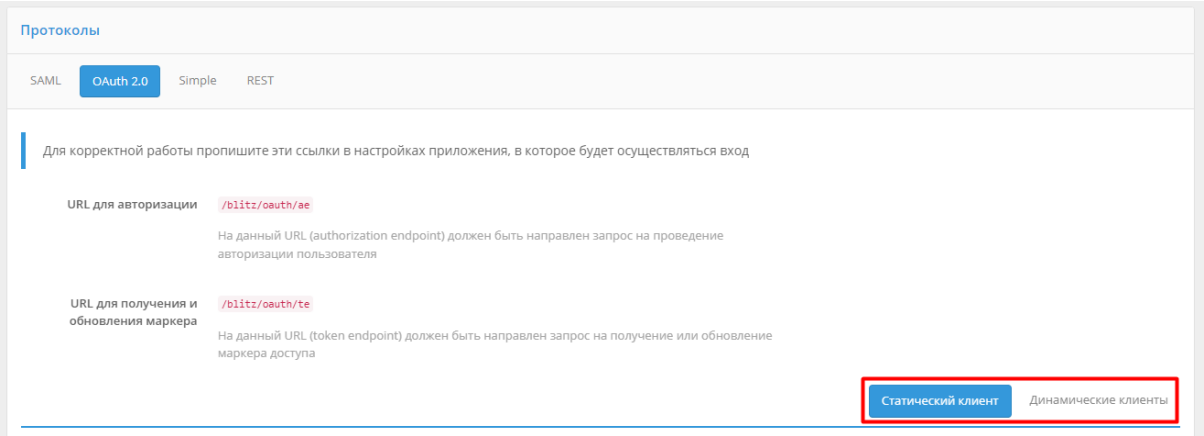
(продолжение с предыдущей страницы)

```
            "value" : "inn"
        }
        {
            "name" : "esia_passport",
            "value" : "passport"
        }
        {
            "name" : "esia_birthPlace",
            "value" : "birthPlace"
        }
        {
            "name" : "esia_email",
            "value" : "email"
        }
        {
            "name" : "esia_mobile",
            "value" : "mobile"
        }
    ],
```

Настройка динамической регистрации клиентов OAuth 2.0

Чтобы включить возможность динамической регистрации клиентов, необходимо выполнить следующие шаги:

- зарегистрировать приложение и настроить для него протокол подключения OAuth 2.0 согласно документации (см. *Общие настройки OAuth 2.0* (страница 242));
- в настройках OAuth 2.0 для данного приложения перейти на закладку *«Динамические клиенты»*.



Указать базовые настройки динамической регистрации клиентов:

- разрешить динамическую регистрацию клиентов;
- указать допустимые к прямой передаче утверждения. Эти утверждения допускается указывать в запросе на регистрацию экземпляра приложения. В случае их наличия в метаданных приложения (`software_statement`), приоритет будет отдан значению из метаданных. Рекомендуется разрешить передачу только типа устройства (`device_type`).

Создать первичные маркеры для приложения. Первичные маркеры используются для авторизации инстансов приложения при их регистрации.

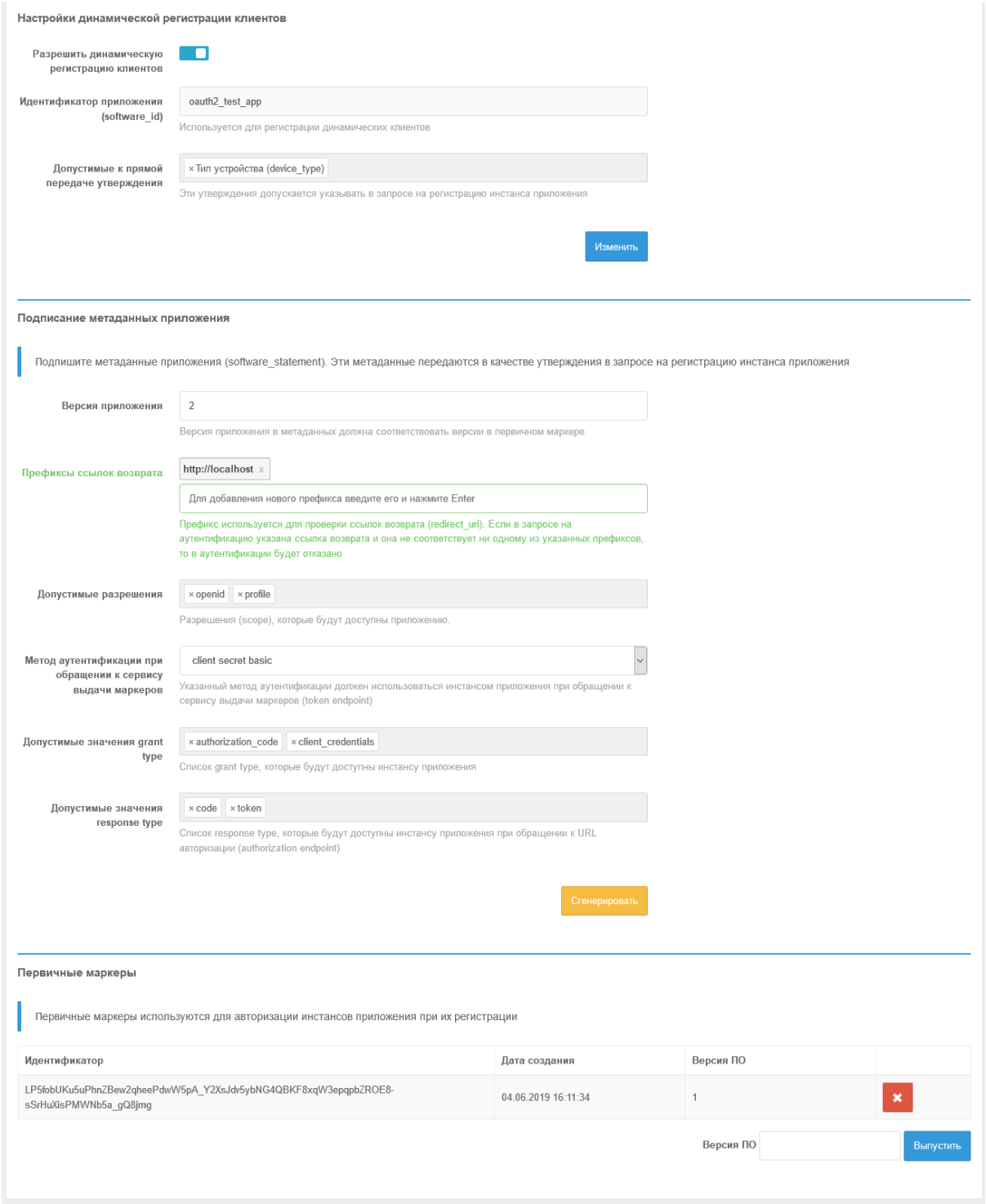
Сгенерировать метаданные приложения (`software_statement`). Эти метаданные передаются в качестве утверждения в запросе на регистрацию экземпляра приложения. В качестве атрибутов метаданных можно указать:

- версию приложения (обязательный атрибут). Версия приложения должна соответствовать версии первичного маркера, используемого приложением;
- префиксы ссылок возврата. Префикс используется для проверки ссылок возврата (`redirect_uri`). Если в запросе на аутентификацию указана ссылка возврата и она не соответствует ни одному из указанных префиксов, то в аутентификации будет отказано;
- допустимые разрешения – разрешения (`scope`), которые будут доступны приложению;
- метод аутентификации при обращении к сервису выдачи маркеров. Указанный метод аутентификации должен использоваться инстансом приложения при обращении к сервису выдачи маркеров (`token endpoint`);
- допустимые значения `grant type`. Список `grant type`, которые будут доступны экземпляру приложения;
- допустимые значения `response type`. Список `response type`, которые будут доступны экземпляру приложения при обращении к URL авторизации (`authorization endpoint`)

Следует учесть, что указанные атрибуты метаданных должны соответствовать параметрам OAuth 2.0, определенным для приложения (*«Статический клиент»*).

После подписания метаданных приложения их вместе с первичными маркерами следует передать разработчикам подключаемого приложения.

Пример настроек динамической регистрации клиента представлен на рисунке ниже.


Настройки динамической регистрации клиентов
Разрешить динамическую регистрацию клиентов
Идентификатор приложения (software_id)
oauth2_test_app
Используется для регистрации динамических клиентов
Допустимые к прямой передаче утверждения
× Тип устройства (device_type)
Эти утверждения допускается указывать в запросе на регистрацию инстанса приложения
Изменить
Подписание метаданных приложения
Подпишите метаданные приложения (software_statement). Эти метаданные передаются в качестве утверждения в запросе на регистрацию инстанса приложения
Версия приложения
2
Версия приложения в метаданных должна соответствовать версии в первичном маркере
Префиксы ссылок возврата
http://localhost ×
Для добавления нового префикса введите его и нажмите Enter
Префикс используется для проверки ссылок возврата (redirect_uri). Если в запросе на аутентификацию указана ссылка возврата и она не соответствует ни одному из указанных префиксов, то в аутентификации будет отказано
Допустимые разрешения
× openid × profile
Разрешения (scope), которые будут доступны приложению.
Метод аутентификации при обращении к сервису выдачи маркеров
client secret basic
Указанный метод аутентификации должен использоваться инстансом приложения при обращении к сервису выдачи маркеров (token endpoint)
Допустимые значения grant type
× authorization_code × client_credentials
Список grant type, которые будут доступны инстансу приложения
Допустимые значения response type
× code × token
Список response type, которые будут доступны инстансу приложения при обращении к URL авторизации (authorization endpoint)
Сгенерировать
Первичные маркеры
Первичные маркеры используются для авторизации инстансов приложения при их регистрации
Идентификатор
Дата создания
Версия ПО
LP5fobUKu5uPhnZBew2qheePdwW5pA_Y2XsJdr5ybNG4QBKF8xqW3epqpbZROE8-sSrHuXisPMWNb5a_gQ8jmg
04.06.2019 16:11:34
1
Версия ПО
Выпустить

2.3.5 Настройка Simple

Данный способ подключения приложения к Blitz Identity Provider можно применять при следующих условиях:

- Приложение нельзя подключить к Blitz Identity Provider с использованием стандартных протоколов SAML или OIDC.
- Приложение представляет собой веб-приложение, развернутое в собственной инфраструктуре (On-Premise). Доступ пользователей к приложениям можно организовать через реверсивный прокси-сервер.

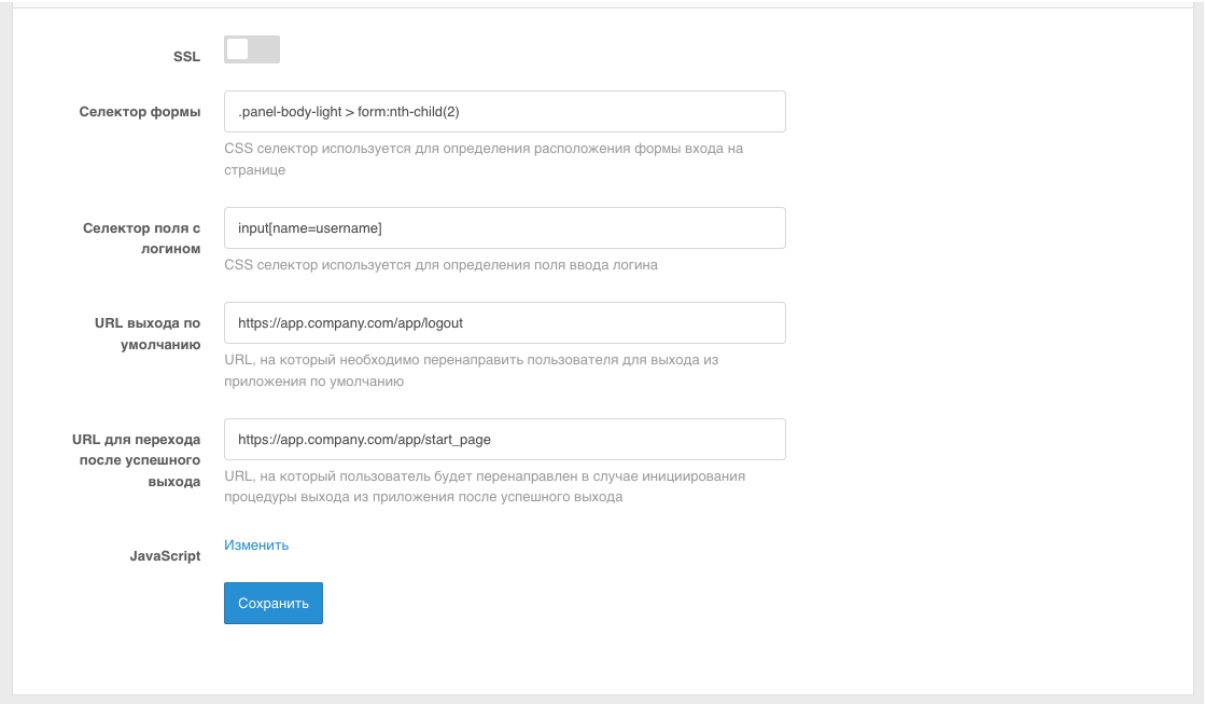
Чтобы подключить приложение к Blitz Identity Provider по протоколу Simple, необходимо:

1. В настройках приложения в консоль управления выбрать протокол Simple и задать его настройки:
   - SSL – настройка, указывающая, производится ли за прокси вызов подключаемого по Simple приложения по HTTP или по HTTPS. Рекомендуется в качестве прокси-сервера, защищающего приложение, использовать существующий веб-сервер приложения, и в таком случае соединение прокси-сервера с приложением будет осуществляться без TLS/SSL шифрования.
   - Селектор формы – задается CSS-селектор, позволяющий определить положение формы входа на странице подключаемого приложения.
   - Селектор поля с логином – задается CSS-селектор, позволяющий определить положение поля ввода логина на странице входа подключаемого приложения.
   - URL выхода по умолчанию (опциональная настройка) – указывает, какой адрес должен вызвать Blitz Identity Provider при необходимости инициировать логаут в подключенном по Simple приложении в случае единого логаута в Blitz Identity Provider.
   - URL для перехода после успешного выхода – указывает, какой адрес должен вызвать Blitz Identity Provider для перенаправления пользователя после успешного логаута, инициированного подключенным по Simple приложением.
   - JavaScript (опциональная настройка) – встраиваемый в страницу входа подключаемого по Simple приложения JS-код, позволяющий обработать полученный от приложения ответ с результатами входа (проверить, что вход произведен успешно) и показать об этом ошибку в Blitz Identity Provider.

   Пример значения:

```
var fm = document.querySelector('form[name=login]');

if (fm) {
    document.body.style.display = "none";
    var err = document.getElementById('lost-password');
    var errKey = err && err.innerHTML.indexOf('Incorrect password.') !== -1 ?
↪'incorrect_password' : 'unknown_error';
    var kvp = document.location.search.substr(1).split('&');
    kvp.push([encodeURI('error'),encodeURI(errKey)].join('='));
    window.location.search = kvp.join('&');
}

var aLogout = document.querySelector('#logout');
var href = aLogout ? aLogout.getAttribute("href") : null;
if (href) {
    var lp = encodeURIComponent(href);
    var slp = document.createElement('script');
    slp.setAttribute('src', 'https://idp.company.com/blitz/simple/slp?app=app_
↪id&lp=' + lp);
    document.head.appendChild(slp);
}
```

   Пример настроек протокола Simple для приложения представлен на рисунке ниже.



2. Задать настройки проксирования запросов к приложению на веб-сервере.

   Пример конфигурационного файла для веб-сервера nginx:

```
map "" $idp_host {
        default <server hostname>:9000;
}

map "$http_Blitz_Idp" $idp_post_login {
        default "0";
        "prepare-login" "1";
}

map "$arg_passive" $activLogout {
        default "1";
        "true" "0";
}

upstream oc-web {
    server <application server hostname>:<application port>;
}

server {
    listen 80;
    server_name <application domain name>;
    # enforce https
    return 301 https://$server_name$request_uri;
}

server {
    listen         443 ssl;
    server_name    <application domain name>;

    resolver               172.27.0.20 172.25.0.50 valid=300s;
    #resolver               8.8.8.8 valid=300s;

    #ssl_certificate        /etc/nginx/cert/<path to SSL certificate>.pem;
```

(continues on next page)

(продолжение с предыдущей страницы)

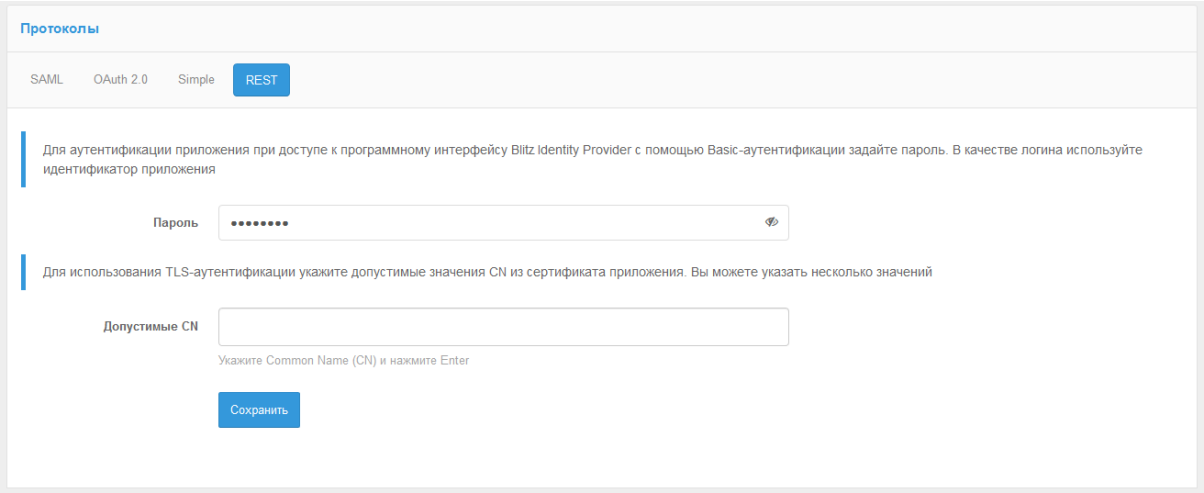
```
    #ssl_certificate_key    /etc/nginx/cert/<path to SSL certificate key>.pem;

    #ssl_certificate /etc/letsencrypt/live/app.company.com/fullchain.pem; #␣
↪managed by Certbot
    #ssl_certificate_key /etc/letsencrypt/live/app.company.com/privkey.pem; #␣
↪managed by Certbot

    access_log              /var/log/nginx/oc-acs.log full;
    error_log               /var/log/nginx/oc-err.log error;

    ### force timeouts if one of backend is died ##
    proxy_next_upstream error timeout invalid_header http_500 http_502 http_
↪503 http_504;

    ### Set headers ####
    proxy_set_header        Accept-Encoding   "";
    proxy_set_header        Host            $host;
    proxy_set_header        X-Real-IP       $remote_addr;
    proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header        X-Forwarded-Proto $scheme;
    add_header              Front-End-Https   on;
    proxy_redirect          off;

    proxy_set_header        Cookie "$http_cookie;domain2auth=$host";
    proxy_hide_header       Content-Security-Policy;

    add_header Content-Security-Policy "default-src 'self' https://$idp_host;␣
↪script-src 'self' https://$idp_host 'unsafe-eval'; img-src 'self' data:␣
↪https://$idp_host; style-src 'self' 'unsafe-inline'; font-src 'self' data:;␣
↪frame-src 'self'; connect-src 'self'";

    location ~ <path to login page of the application>$ {
            #if ($http_referer ~* "/blitz/simple") {
            #    set $idp_post_login "1";
            #}
            if ($http_referer ~* "<main server domain name>") {
                set $idp_post_login "1";
            }
            if ($idp_post_login = "1" ) {
                proxy_pass http://oc-web$request_uri;
            }
            if ($idp_post_login = "0" ) {
                proxy_pass http://$idp_host/blitz/simple/prepare$request_uri;
                break;
            }
    }
    location ~ /logout$ {
        if ($activLogout = "1") {
            return 302 https://<main server domain name>/blitz/simple/active_
↪logout?app=$host;
        }
        proxy_pass http://oc-web$request_uri;
    }
    location / {
        proxy_pass http://oc-web;
    }
}
```

2.3.6 Взаимодействие по REST API

Для вызова REST-сервисов Blitz Identity Provider необходимо настроить приложение, которое будет выступать в качестве системы-клиента REST-сервисов. Для этого нужно *зарегистрировать* (страница 224) новое приложение в разделе Приложения.

Далее перейти к настройкам приложения, в качестве протокола подключения указать REST и заполнить следующие данные:

- `Пароль` – будет использоваться использоваться при HTTP Basic-аутентификации, в качестве логина – идентификатор системы-клиента; если параметр не задан, то HTTP Basic-аутентификация не будет возможна для данной системы-клиента;
- `Допустимые CN` – перечень значений атрибута `CN` сертификата, используемого при TLS-аутентификации; если не заданы параметры, то TLS-аутентификация не будет возможна для данной системы-клиента.



Если для приложения не заданы настройки протокола подключения REST, то приложение не сможет использовать REST API сервера Blitz Identity Provider, защищаемые с использованием HTTP Basic авторизации.

2.3.7 Доступ к сетевым службам по RADIUS

Существует возможность настроить подключение пользователей к точкам сетевого доступа (RDP, VPN, Wi-Fi и др.) по протоколу RADIUS. Настройка подключения выполняется в описанной ниже последовательности.

Справка по RADIUS

Remote Authentication Dial In User Service (RADIUS) RFC 2865[62] — протокол, используемый для централизованного управления авторизацией, аутентификацией и учетом доступа в сетевые службы и оборудование. Через данный протокол выполняется взаимодействие между сервером и клиентом RADIUS. После запроса пользователем доступа в сетевую службу соответствующий клиент отправляет серверу запрос, в результате которого сервер проверяет наличие пользователя в базе данных. Если пользователь найден, сервер отправляет клиенту разрешение на его аутентификацию.

Сервером RADIUS выступает Blitz Identity Provider, клиентом — подключенная сетевая служба. В текущей реализации сервер выполняет поиск пользователей во всех подключенных хранилищах. Сетевые службы настраиваются в Blitz Identity Provider как приложения.

Сервер поддерживает следующие методы аутентификации:

- первый фактор: вход по логину и паролю;

[62] https://datatracker.ietf.org/doc/html/rfc2865

- второй фактор: подтверждение по коду из SMS, PUSH, TOTP, HOTP, email или через Личный кабинет пользователя.

Шаг 1. Конфигурирование сервера RADIUS

Для конфигурирования сервера RADIUS в Blitz Identity Provider выполните следующие действия:

1. В консоли управления перейдите в раздел RADIUS.
2. Последовательно настройте конфигурацию сервера.

Общие настройки

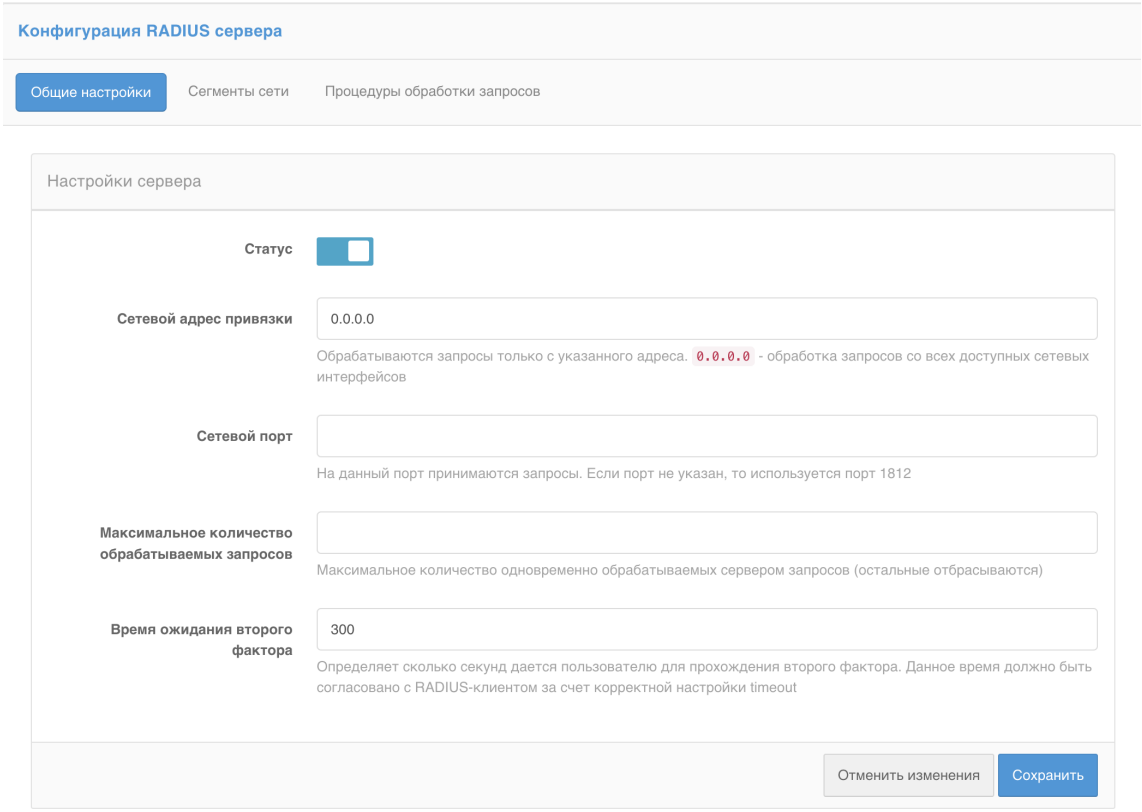
На данной вкладке указываются общие настройки сервера RADIUS.

- `Статус`: включение сервера.
- `Сетевой адрес привязки`: список адресов, с которых сервер обрабатывает запросы.

Совет: Для обработки запросов со всех доступных сетевых интерфейсов установите `0.0.0.0`.

- `Сетевой порт`: порт RADIUS, на который принимаются запросы. Если порт не указан, то используется порт `1812`.
- `Максимальное количество обрабатываемых запросов`: максимальное количество одновременно обрабатываемых сервером запросов (остальные отбрасываются).
- `Время ожидания второго фактора`: время в секундах, которое дается пользователю для прохождения второго фактора.

Внимание: Данное время должно быть согласовано с RADIUS-клиентом за счет корректной настройки времени ожидания ответа RADIUS-сервера.



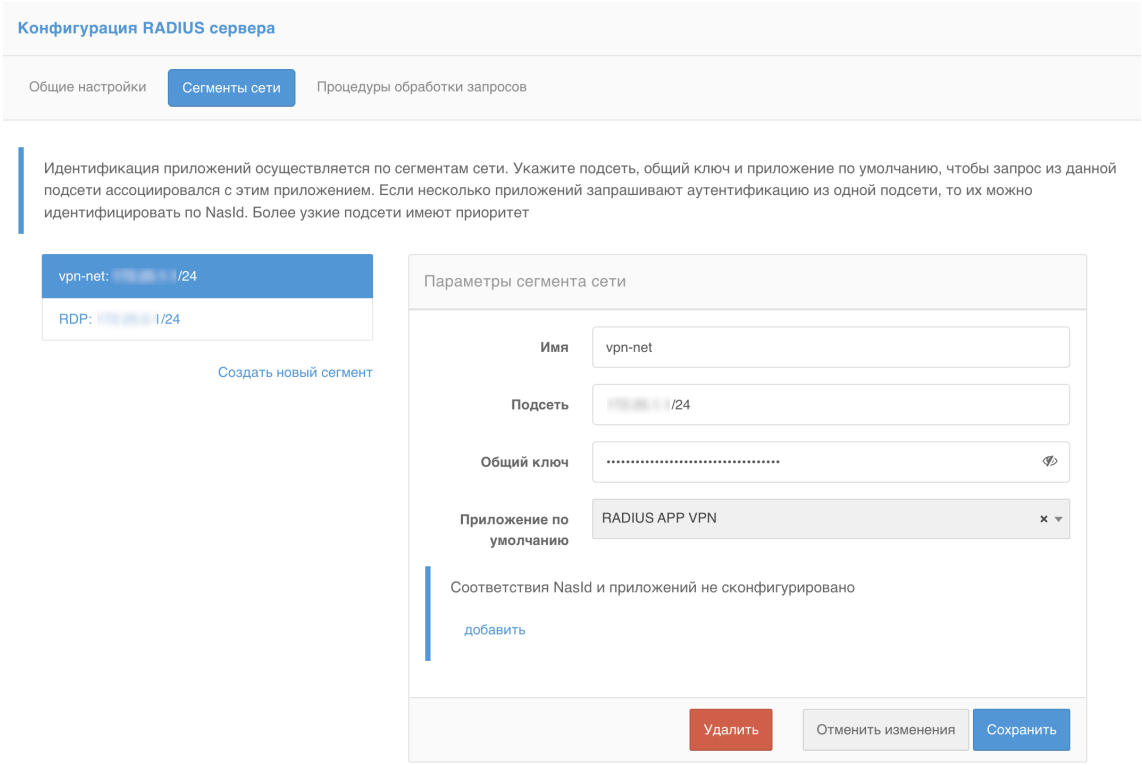
Нажмите Сохранить.

Сегменты сети

Идентификация приложений осуществляется по сегментам сети. Укажите подсеть, общий ключ и приложение по умолчанию, чтобы запрос из данной подсети ассоциировался с этим приложением. Если несколько приложений запрашивают аутентификацию из одной подсети, то их можно идентифицировать по `NasId`.

Внимание: Подсети с более узким префиксом имеют приоритет.

- `Имя`: введите произвольное имя сегмента сети.
- `Подсеть`: введите префикс подсети, запросы из которой будут ассоциироваться с приложением.
- `Общий ключ`: сгенерируйте и введите ключ, который нужно будет *ввести на стороне сетевой службы* (страница 260).
- `Приложение по умолчанию`: выберите приложение, с которым будет ассоциироваться запрос из данной подсети. Если приложений несколько, оно будет выступать приложением по умолчанию.
- Соответствие `NasId` и приложений: если предполагается, что из одной подсети запрашивать аутентификацию будет несколько приложений, задайте `NasId`, по которым сервер RADIUS будет их идентифицировать.



Нажмите Сохранить.

Процедуры обработки запросов

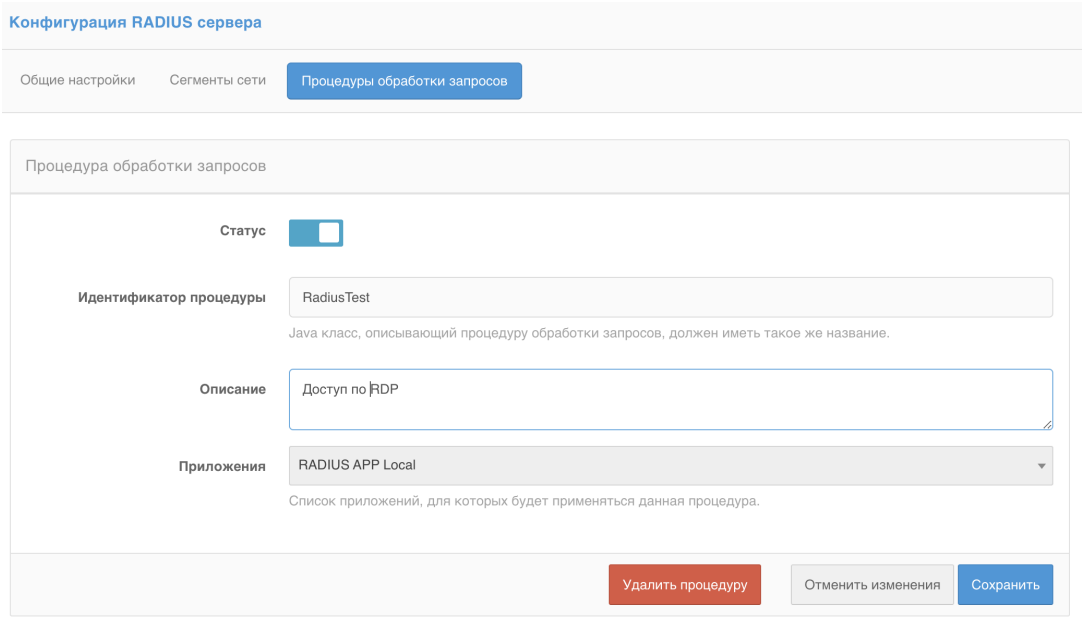
Данная вкладка содержит список процедур на Java, которые будут обрабатывать запросы из подключенных приложений. Процедуры определяют фактор аутентификации и реализуют другие политики доступа в сетевые ресурсы. В простейшем случае процедуры включают первый либо второй фактор. Можно создать несколько процедур в зависимости от требований к безопасности различных сетевых точек.

Для создания процедуры обработки запросов выполните следующие действия:

1. Нажмите Создать новую процедуру обработки запросов.
2. Задайте настройки:
   - `Статус`: включение процедуры.
   - `Идентификатор процедуры`: задайте идентификатор процедуры.

     Внимание: Java класс, описывающий процедуру обработки запросов, должен иметь такое же название.

   - `Описание`: введите описание процедуры.



3. Нажмите Сохранить.
4. Введите исходный код процедуры:
   - Для управления процессом обработки RADIUS запросов необходимо написать на языке Java класс, реализующий интерфейс `RadiusFlow`.
   - В случае использования второго фактора аутентификации вызовите `RadiusResult.more("method")`, где `method` принимает одно из следующих значений: `sms`, `push`, `totp`, `hotp`, `email`, `prfc` (подтверждение в Личном кабинете пользователя).

Примечание: При подтверждении через Личный кабинет в нем появляется сообщение о попытке входа, в котором пользователь должен нажать Подтвердить.

Внимание: Для того чтобы фактор сработал, Личный кабинет должен быть открыт с обязательным прохождением двух факторов аутентификации.

Список 11: Пример процедуры 2FA через подтверждение в Личном кабинете

```
package com.identityblitz.idp.radius.flow;

public class RadTest2 implements RadiusFlow {

  public String loginN12(final String login) {
    return login;
  }

  public RadiusResult next(final RadiusContext context) {
    if (context.factor() == 1) {
      //return RadiusResult.more("sms");

      return RadiusResult.more("prfc");
    }

    return  RadiusResult.authenticated(context.subject());
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  }

}
```

- В случае использования первого фактора деактивируйте условие `if (context.factor() == 1)`.

Список 12: Пример процедуры 1FA

```
package com.identityblitz.idp.radius.flow;

public class TestRadius implements RadiusFlow {

  public String loginN12(final String login) {
    return login;
  }

  public RadiusResult next(final RadiusContext context) {

    return  RadiusResult.authenticated(context.subject());
  }

}
```

- В процедуре можно отображать выбор способа подтверждения `RadiusResult.challenge`, а также инструкцию для прохождения второго фактора `RadiusResult.dialog`.

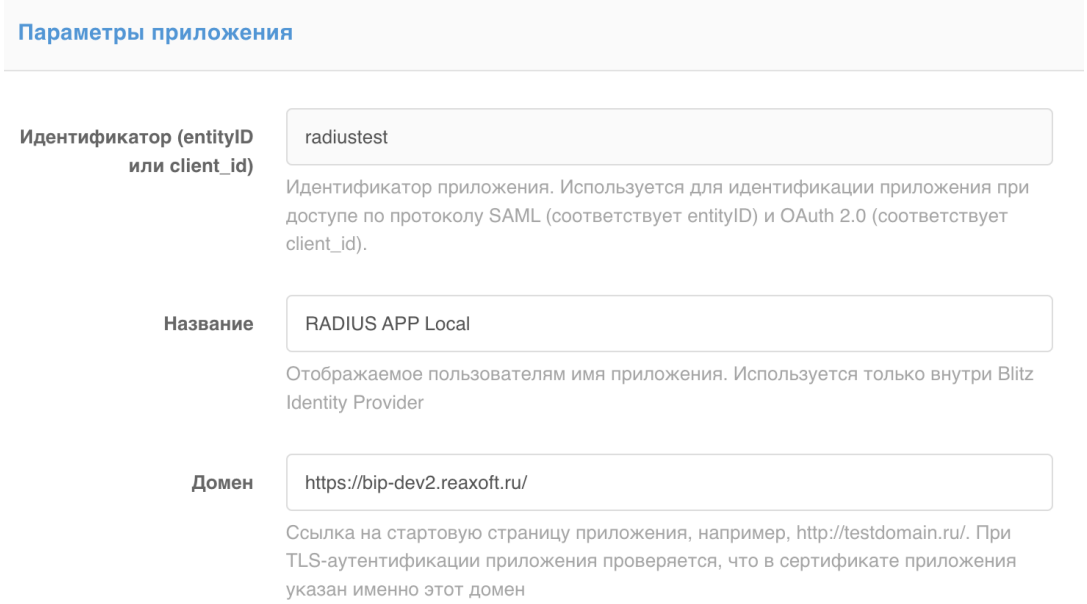
```
  private final Logger logger = LoggerFactory.getLogger("com.
↪identityblitz.idp.flow.radius");

  public String loginN12(final String login) {
    return login;
  }

  public RadiusResult next(final RadiusContext context) {
    if (context.factor() == 1) {
      return RadiusResult.challenge(Challenges.password());
    }
    return  RadiusResult.authenticated(context.subject());
  }

  public RadiusResult dialog(final RadiusContext context,
                             final String message,
                             final java.util.Map<String, String>␣
↪answers,
                             final String answer) {
    if(message.equals("challengeChoose")) {
      final String challenge = answers.get(answer);
        if(challenge != null) return RadiusResult.challenge(Challenges.
↪byName(challenge));
        else return RadiusResult.dialog(message, answers);
    } else {
      return RadiusResult.rejected("unsupportedMessage");
    }
  }

}
```

5. Для компиляции нажмите Сохранить.

Шаг 2. Настройка приложения

Для настройки приложения выполните следующие действия:

1. В консоли управления перейдите в раздел Приложения. *Создайте* (страница 224) приложение с базовыми настройками.
   - `Идентификатор (entityID или client_id)`,
   - `Название`,
   - `Домен`: домен сетевой службы.



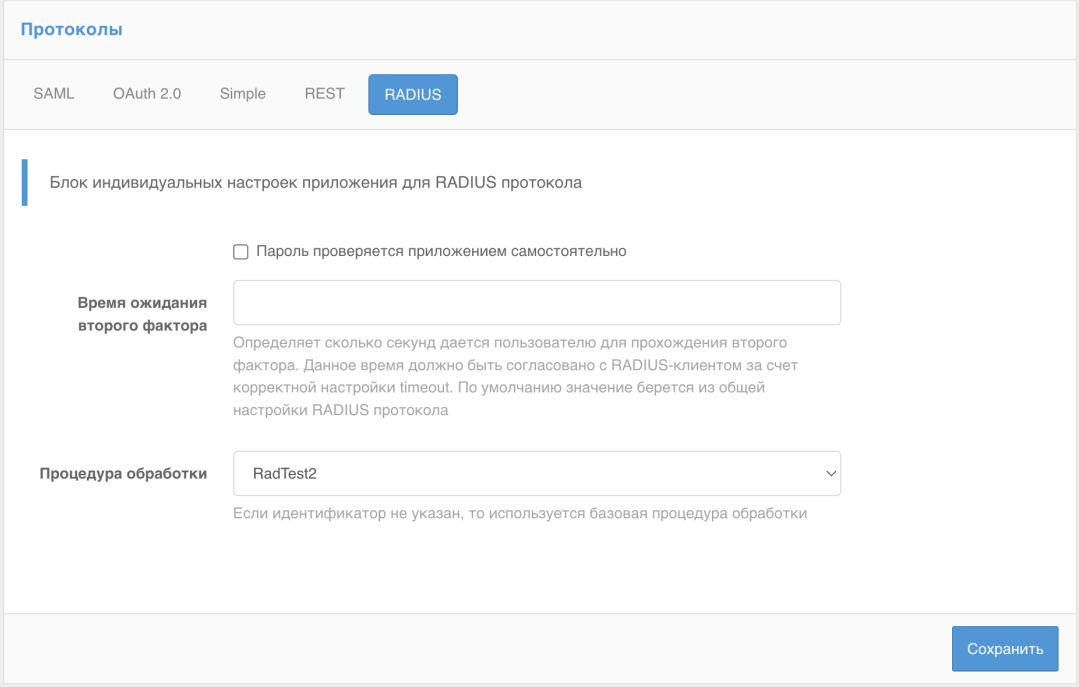
   Нажмите Сохранить.

2. В секции Протоколы приложения на вкладке RADIUS задайте следующие настройки:
   - Поставьте флажок `Пароль проверяется приложением самостоятельно`, если Blitz Identity Provider будет использоваться для второго фактора аутентификации.
   - `Время ожидания второго фактора`: время в секундах, которое дается пользователю для прохождения второго фактора. Если параметр не задан, будет взято значение из настроек сервера RADIUS.

     Внимание: Данное время должно быть согласовано с RADIUS-клиентом за счет корректной настройки времени ожидания ответа RADIUS-сервера.

   - Выберите процедуру обработки запросов от приложения. В списке `Процедура обработки` отображаются все *созданные* (страница 254) на сервере RADIUS процедуры.

     Внимание: Внимательно настраивайте *интеграцию* (страница 260) на стороне сетевой службы. Если в приходящих от приложения запросах не определен `NasId`, приложение узнается Blitz Identity Provider как приложение по умолчанию для данного сегмента сети, даже если фактически это разные приложения. В этом случае будет выполняться процедура обработки запросов, установленная для приложения по умолчанию, а не та, которая выбрана.



Нажмите Сохранить.

Шаг 3. Настройка на стороне сетевой службы

Для завершения подключения введите следующие настройки на стороне сетевой службы:

- IP-адрес сервера с `blitz-idp`.
- Общий ключ, заданный в настройках *сегмента сети* (страница 254), соответствующего приложению (сетевой службе) на сервере RADIUS. По данному ключу сервер будет опознавать сетевую службу и запускать выбранную для нее процедуру обработки доступа.
- `NasId` (при необходимости).
- Время ожидания ответа от сервера RADIUS, соответствующее установленному на сервере времени ожидания второго фактора.

2.4 Кастомизация работы с помощью программирования на Java

2.4.1 Процедуры входа и их создание

О процедурах входа

Процедуры входа на Java применяются для настройки правил доступа пользователей к различным приложениям. С помощью процедур можно определить, например, какие приложения должны быть доступны каким пользователям, при каких условиях должна требоваться двухфакторная аутентификация и какие методы подтверждения входа может применять пользователь. Применение процедур входа позволяет организации исполнить принятые в ней политики контроля доступа к приложениям.

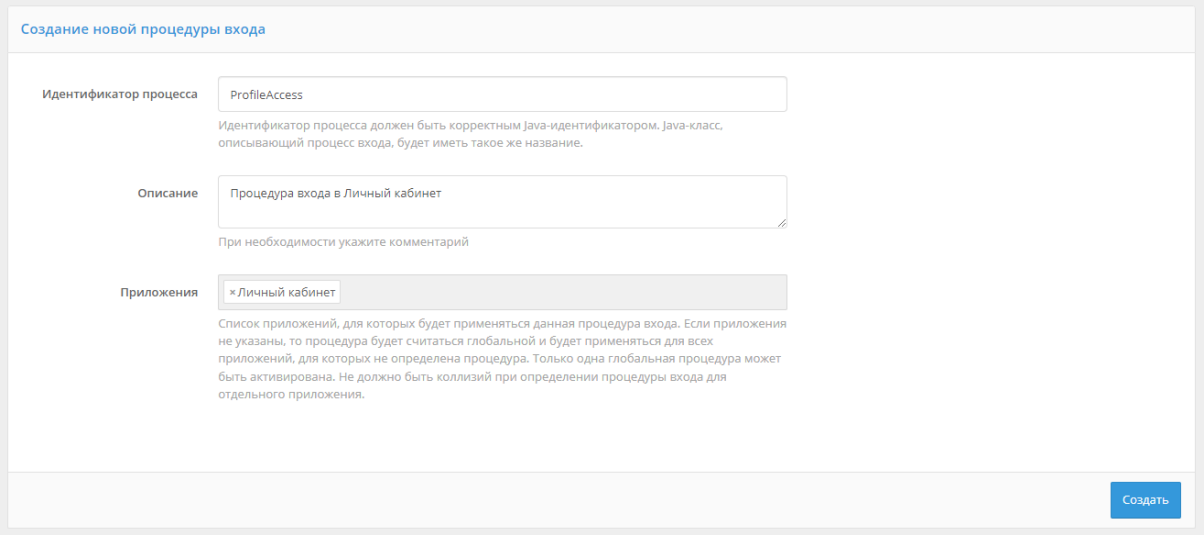
Управление процедурами входа осуществляется в разделе Процедуры входа консоли управления Blitz Identity Provider.

Создание процедуры

Создание процедуры входа включает в себя следующие шаги:

1. Указание базовых параметров процедуры:
    - идентификатор процесса (процедуры);
    - описание процедуры;
    - приложения – перечень приложений, для которых будет применяться данная процедура.

Важно: Для каждого приложения может быть создана только одна процедура. Если для данного приложения не создано процедуры, к нему будет применяться стандартная процедура входа (процедура входа по умолчанию). Если процедура создана без указания приложений, то она заменит стандартную процедуру входа.

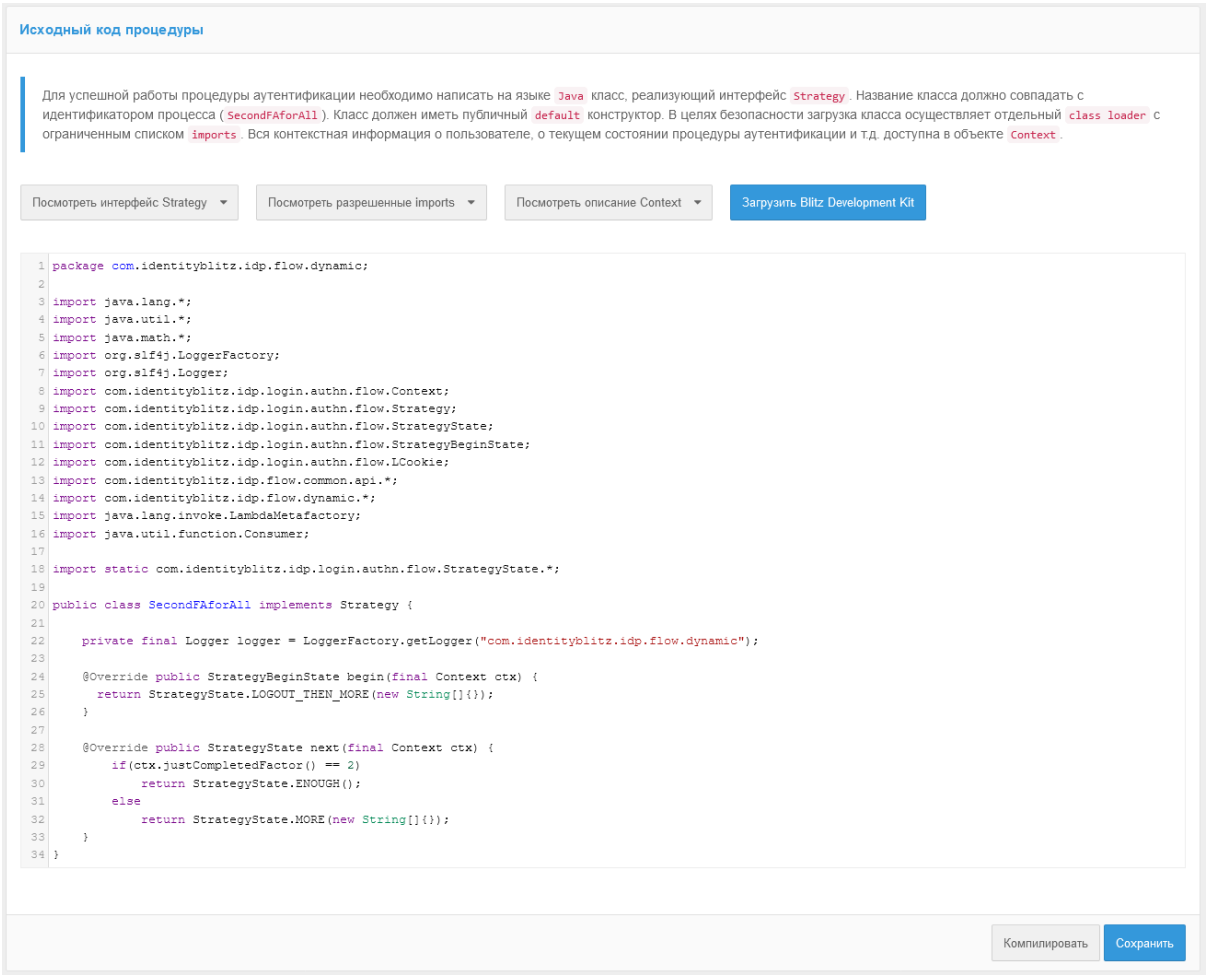


2. Написание исходного кода процедуры. Для успешной работы процедуры входа необходимо написать на языке Java класс, реализующий необходимый интерфейс `Strategy`. Вся контекстная информация о пользователе, о текущем состоянии процедуры аутентификации и т.д. доступна в объекте `Context`. Процедура состоит из двух блоков, которые определяют:
    - действия, предпринимаемые на начальном этапе процесса аутентификации. В этом блоке, например, можно определить, при каких условиях осуществлять переход в приложение в режиме SSO (если пользователь ранее был аутентифицирован);

- действия, предпринимаемые после первичной аутентификации пользователя. В этом блоке, например, можно определить, какие методы двухфакторной аутентификации при каких условиях использовать.

3. После написания кода необходимо нажать на кнопку «Компилировать». При наличии ошибок некорректные фрагменты кода будут выделены цветом и подписаны ошибки.
4. Если компиляция прошла успешно, можно сохранить процедуру.
5. Сохраненную процедуру можно активировать – для этого следует нажать на кнопку *«Активировать»* в шапке соответствующей процедуры.
6. Можно редактировать как активированную, так и деактивированную процедуру. После редактирования следует компилировать процедуру, после чего – сохранить. Если процедура была активирована, то новая скомпилированная процедура заменит старую.

Предупреждение: Если процедура активирована, то сохранить можно только ту процедуру, которую удается скомпилировать. Иными словами, если при редактировании активированной процедуры была выявлена ошибка, то кнопка «Сохранить» работать не будет, а при перезагрузке страницы изменения будут утеряны.



2.4.2 Готовые процедуры входа

В поставку входят несколько готовых процедур, которые могут быть при необходимости изменены:

- *принудительная двухфакторная аутентификация в приложение* (страница 263) (`Require2ndFactor`);
- *ограничение перечня доступных методов первого фактора при входе в приложение* (страница 264) (`FFmethods`);
- *предоставление доступа к приложению только при определенном значении атрибута* (страница 265) (`AccessByAttribute`);
- *запрет входа в приложение после истечения срока действия учетной записи* (страница 266) (`AccountExpiresCheck`);
- *разрешение входа в приложение только из определенных сетей* (страница 267) (`AllowedIPs`);
- *запрет работы в нескольких одновременных сессиях* (страница 268) (`RestrictSessions`);
- *сохранение в утверждениях claims перечня групп пользователя* (страница 269) (`AddGroupsToToken`);
- *отображение пользователю объявления при входе* (страница 270) (`InfoPipe`);
- *запрос ввода пользователем атрибута или актуализации телефона и email* (страница 271) (`PipeAttrActAdd`);
- *запрос ввода пользователем контрольного вопроса, если он не задан в учетной записи* (страница 274) (PipeSecQuestion);
- *регистрация ключа безопасности WebAuthn, Passkey, FIDO2 при входе* (страница 275) (`PipeWebAuthn`).
- *отображение пользователю списка выбора значений при входе* (страница 277) (`ChoicePipe`).

Далее приводятся листинги этих процедур. Для удобства отладки можно выводить информацию о состоянии аутентификации в лог, воспользовавшись функцией `logger.debug()`. Например, следующая команда выведет в лог заданный уровень аутентификации для пользователя:

```
logger.debug("requiredFactor="+ctx.userProps("requiredFactor"));
```

Принудительная двухфакторная аутентификация

Процедура `Require2ndFactor` требует двухфакторной аутентификации для доступа к приложению. Если пользователь переходит в приложение в рамках единой сессии, то при наличии одного пройденного фактора у него будет дополнительно проверен второй фактор, т.е. SSO в этом случае не сработает.

```
public class Require2ndFactor implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.
↪flow.dynamic");

    @Override public StrategyBeginState begin(final Context ctx) {
        if(ctx.claims("subjectId") != null){
            if (ctx.sessionTrack().split(",").length < 2)
                return StrategyState.MORE(new String[]{});
            else
                return StrategyState.ENOUGH();
        }
        else {
            return StrategyState.MORE(new String[]{});
        }
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    }

    @Override public StrategyState next(final Context ctx) {
        if(ctx.justCompletedFactor() == 1)
            return StrategyState.MORE(new String[]{});
        else
            return StrategyState.ENOUGH();
    }
}
```

Ограничение перечня доступных методов первого фактора

Процедура `FFmethods` позволяет при входе в приложение предлагать пользователю только определенные методы идентификации и аутентификации (аналогичную процедуру с иным перечнем методов, можно назначить другому приложению). Для обозначения методов аутентификации первого фактора в процедуре используются следующие идентификаторы:

- `password` – вход по логину и паролю;
- `x509` – вход по электронной подписи;
- `externalIdps` – вход через внешние поставщики идентификации (социальные сети и пр.);
- `spnego` – вход по сеансу операционной системы;
- `sms` – вход по коду подтверждения в SMS-сообщении.
- `knownDevice` – вход по известному устройству;
- `qrCode` – вход по QR-коду;
- `webAuthn` – вход с помощью ключей безопасности (WebAuthn, Passkey, FIDO2);
- `tls` – вход на основе переданного HTTP-заголовка.

```
public class FFmethods implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.
↪flow.dynamic");

    @Override public StrategyBeginState begin(final Context ctx) {
        if(ctx.claims("subjectId") != null)
            return StrategyState.ENOUGH();
        else
            return StrategyState.MORE(new String[]{"password","x509"});
    }

    @Override public StrategyState next(final Context ctx) {
        Integer reqFactor = (ctx.user() == null) ? null : ctx.user().
↪requiredFactor();
        if(reqFactor == null || reqFactor == 0)
            return StrategyState.ENOUGH();
        else {
            if(reqFactor == ctx.justCompletedFactor())
                return StrategyState.ENOUGH();
            else
                return StrategyState.MORE(new String[]{});
        }
    }
}
```

Вход только при определенном значении атрибута

Процедура `AccessByAttribute` использует атрибут `appList` для принятия решения о доступе пользователя к приложению. Для работы этой процедуры необходимо создать атрибут `appList` в виде массива (`Array of strings`). В качестве значений элементов этого массива следует использовать идентификаторы приложений. В результате доступ к приложению будет предоставлен, если среди значений `appList` у данного пользователя будет идентификатор этого приложения. Такая архитектура процедуры позволяет назначить ее сразу нескольким приложениям и регулировать доступ к ним при помощи одного атрибута.

```
public class AccessByAttribute implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.
↪flow.dynamic");

    @Override public StrategyBeginState begin(final Context ctx) {
        if(ctx.claims("subjectId") != null){
            int appListIdx = 0;
            boolean hasAccess = false;
            while (appListIdx > -1) {
                String app = ctx.claims("appList.[" + appListIdx + "]");
                logger.debug("app [" + appListIdx + "] = " + app);
                if (app == null){ appListIdx = -1; }
                else if (app.equals(ctx.appId())) { appListIdx = -1; hasAccess =␣
↪true; }
                else { appListIdx ++; logger.debug("AppList index = " +␣
↪appListIdx); }
            }
            if(hasAccess)
                return StrategyState.ENOUGH();
            else
                return StrategyState.DENY;
        }
        else
            return StrategyState.MORE(new String[]{});
    }

    @Override public StrategyState next(final Context ctx) {
        int appListIdx = 0;
        boolean hasAccess = false;
        while (appListIdx > -1) {
            String app = ctx.claims("appList.[" + appListIdx + "]");
            logger.debug("app [" + appListIdx + "] = " + app);
            if (app == null){ appListIdx = -1; }
            else if (app.equals(ctx.appId())) { appListIdx = -1; hasAccess = true;␣
↪}
            else { appListIdx ++; logger.debug("AppList index = " + appListIdx); }
        }
        if(!hasAccess)
            return StrategyState.DENY;
        Integer reqFactor = 0;
        if (ctx.user() != null) {
            reqFactor = ctx.user().requiredFactor();
        }
        if (reqFactor == 0)
            return StrategyState.ENOUGH();
        else {
            if (reqFactor == ctx.justCompletedFactor())
                return StrategyState.ENOUGH();
            else
                return StrategyState.MORE(new String[]{});
        }
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    }
}
```

Пример упрощенного варианта процедуры – допуск пользователя в приложение при условии, что адрес его электронной почты равен `ivanov@company.ru`:

```
@Override public StrategyBeginState begin(final Context ctx) {
  if(ctx.claims("subjectId") != null){
    if("ivanov@company.ru".equals(ctx.claims("email")))
      return StrategyState.ENOUGH();
    else
      return StrategyState.DENY;
  }
  else
    return StrategyState.MORE(new String[]{});
}

@Override public StrategyState next(final Context ctx) {
  if(!"ivanov@company.ru".equals(ctx.claims("email")))
    return StrategyState.DENY;
  Integer reqFactor = (ctx.user() == null) ? null : ctx.user().requiredFactor();
  if(reqFactor == null)
    return StrategyState.ENOUGH();
  else {
    if(reqFactor == ctx.justCompletedFactor())
      return StrategyState.ENOUGH();
    else
      return StrategyState.MORE(new String[]{});
  }
}
```

Запрет входа после истечения срока действия аккаунта

Процедура `AccountExpiresCheck` использует атрибут `accountExpires` для принятия решения о доступе пользователя к приложению. Для работы этой процедуры необходимо создать атрибут `accountExpires` с типом строка (`String`). В этот атрибут необходимо сохранить дату (в формате `гггг-ММ-дд ЧЧ:мм`, например `2021-09-23 13:58`), после наступления которой доступ в приложение будет заблокирован для данного пользователя. Если значение атрибута не указано, то пользователь будет допущен в приложение.

```
public class AccountExpiresCheck implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.
↪flow.dynamic");

@Override public StrategyBeginState begin(final Context ctx) {
  if ("login".equals(ctx.prompt())){
    List<String> methods = new ArrayList<String>(Arrays.asList(ctx.
↪availableMethods()));
    methods.remove("cls");
    return StrategyState.MORE(methods.toArray(new String[0]), true);
  } else {
    if(ctx.claims("subjectId") != null)
      return StrategyState.ENOUGH();
    else
      return StrategyState.MORE(new String[]{});
  }
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
@Override public StrategyState next(final Context ctx) {
  if (ctx.claims("accountExpires") != null && isExpired(ctx.claims("accountExpires
↪")))
    return StrategyState.DENY("account_expired", true);
  Integer reqFactor = (ctx.user() == null) ? null : ctx.user().requiredFactor();
  if(reqFactor == null || reqFactor == ctx.justCompletedFactor())
    return StrategyState.ENOUGH();
  else
    return StrategyState.MORE(new String[]{});
}

public static boolean isExpired(String strData) {
  try {
    Date now = new Date();
            Date date = new SimpleDateFormat("yyyy-M-d HH:mm").parse(strData);
            return now.after(date);
        } catch (ParseException e) {
            throw new RuntimeException(e);
        }
    }
}
```

Вход только из определенных сетей

Процедура `AllowedIPs` использует константу `ALLOW_IP` для принятия решения о доступе пользователя к приложению. В данной константе необходимо прописать перечень сетей, из которых возможен доступ в приложение, допустимо указать несколько сетей. При входе в приложение будет проверен IP адрес пользователя на предмет его соответствия одному из значений, включенных в константу. Если он соответствует, то пользователь будет допущен в приложение, если не соответствует – в доступе будет отказано.

```
public class AllowedIPs implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.
↪flow.dynamic");
    private final static String[] ALLOW_IP = {"179.218","180.219"};

    @Override public StrategyBeginState begin(final Context ctx) {
        if ("login".equals(ctx.prompt())){
            List<String> methods = new ArrayList<String>(Arrays.asList(ctx.
↪availableMethods()));
            methods.remove("cls");
            return StrategyState.MORE(methods.toArray(new String[0]), true);
        } else {
            if(ctx.claims("subjectId") != null)
                return StrategyState.ENOUGH();
            else
                return StrategyState.MORE(new String[]{});
        }
    }

    @Override public StrategyState next(final Context ctx) {
        if (!_allowed_ip(ctx.ip())) {
            return StrategyState.DENY("ip_not_allowed", true);
        }
        Integer reqFactor = (ctx.user() == null) ? null : ctx.user().
↪requiredFactor();
        if(reqFactor == null || reqFactor == ctx.justCompletedFactor()) {
```

(continues on next page)

(продолжение с предыдущей страницы)

```
            return StrategyState.ENOUGH_BUILDER()
                .build();
        } else
            return StrategyState.MORE(new String[]{});
    }

    private Boolean _allowed_ip(final String IP) {
      int IpListIdx = 0;
      boolean ipAllowed = false;
      while (IpListIdx > -1) {
        String ip_part = ALLOW_IP[IpListIdx];
        if (IP.startsWith(ip_part)) {
            ipAllowed = true;
            IpListIdx = -1;
        } else if (ALLOW_IP.length == (IpListIdx + 1)) {
            IpListIdx = -1;
        } else {
            IpListIdx ++;
        }
      }
        return ipAllowed;
    }
}
```

Запрет работы в нескольких одновременных сессиях

Процедура `RestrictSessions` запрещает работу в нескольких сессиях.

```
public class RestrictSessions implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.
↪flow.dynamic");

    @Override public StrategyBeginState begin(final Context ctx) {
      List<String> methods = new ArrayList<String>(Arrays.asList(ctx.
↪availableMethods()));
      if ("login".equals(ctx.prompt())){
        methods.remove("cls");
        return StrategyState.MORE(methods.toArray(new String[0]), true);
      } else {
        if(ctx.claims("subjectId") != null)
          return StrategyState.ENOUGH();
        else {
          methods.remove("cls");
          return StrategyState.MORE(methods.toArray(new String[0]));
        }
      }
    }

    @Override public StrategyState next(final Context ctx) {
      Integer reqFactor = (ctx.user() == null) ? null : ctx.user().
↪requiredFactor();
      if(reqFactor == null || reqFactor == ctx.justCompletedFactor()) {
        return StrategyState.ENOUGH_BUILDER().singleSession(true).build();
      } else
        return StrategyState.MORE(new String[]{});
    }
}
```

Сохранение в утверждениях (claims) перечня групп пользователя

Процедура `AddGroupsToToken` сохраняет в утверждение `grps` перечень групп пользователя. Чтобы эта процедура работала, должны быть выполнены условия:

- сконфигурирован атрибут `memberOf`, в котором отображаются группы пользователя;
- в конфигурационный файл добавлено сессионное утверждение `grps` (см. *Добавление атрибутов в маркер идентификации* (страница 244)).

При входе в приложение будет проверено наличие групп у пользователя в атрибуте `memberOf`, и если они там присутствуют, то они будут добавлены в утверждение `grps`.

```
public class AddGroupsToToken implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.
↪flow.dynamic");

    @Override public StrategyBeginState begin(final Context ctx) {
        if ("login".equals(ctx.prompt())){
            List<String> methods = new ArrayList<String>(Arrays.asList(ctx.
↪availableMethods()));
            methods.remove("cls");
            return StrategyState.MORE(methods.toArray(new String[0]), true);
        } else {
            if(ctx.claims("subjectId") != null)
                return StrategyState.ENOUGH();
            else
                return StrategyState.MORE(new String[]{});
        }
    }

    @Override public StrategyState next(final Context ctx) {
        Integer reqFactor = (ctx.user() == null) ? null : ctx.user().
↪requiredFactor();
        if(reqFactor == null || reqFactor == ctx.justCompletedFactor()) {
            List<String> grps = new ArrayList<String>();
            int groupListIdx = 0;
            while (groupListIdx > -1) {
              String group = ctx.claims("memberOf.[" + groupListIdx + "]");
              logger.debug("### group [" + groupListIdx + "] = " + group);
              if (group == null) {
                groupListIdx = -1;
              } else {
                grps.add(ctx.claims("memberOf.[" + groupListIdx + "]"));
                groupListIdx ++;
              }
            }
            LClaimsBuilder  claimsBuilder = ctx.claimsBuilder();
            if (grps.size() > 0) {
                claimsBuilder.addClaim("grps", grps);
            }
            LClaims claims = claimsBuilder.build();
            return StrategyState.ENOUGH_BUILDER()
                .withClaims(claims)
                .build();
        } else
            return StrategyState.MORE(new String[]{});
    }
}
```

Отображение пользователю объявления при входе

Можно настроить, чтобы при входе Blitz Identity Provider показал пользователю объявление. При этом пользователю могут быть показаны одна или две кнопки, а выбор пользователя можно будет проанализировать в процедуре входа.

Процедура

Процедура `InfoPipe` позволяет с периодичностью в 30 дней показывать пользователю при входе объявления. Перед использованием нужно выполнить следующие изменения:

- в функции `requiredNews()` скорректировать критерии отображения объявления – например, в примере настроено, что показывать раз в 30 дней в случае если в прошлый раз пользователь при показе объявления нажал кнопку отказа;
- в константе `DOMAIN` указать `URI`, по которому из браузера пользователя доступен Blitz Identity Provider;
- *настроить* (страница 271) в конфигурационном файле тип уведомления;
- *настроить* (страница 308) в сообщениях текст уведомления и названия кнопок.

```
public class InfoPipe implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.
↪flow.dynamic");
        private final static String DOMAIN = "example.com";

    @Override public StrategyBeginState begin(final Context ctx) {
        if ("login".equals(ctx.prompt())){
            List<String> methods = new ArrayList<String>(Arrays.asList(ctx.
↪availableMethods()));
            methods.remove("cls");
            return StrategyState.MORE(methods.toArray(new String[0]), true);
        } else {
            if(ctx.claims("subjectId") != null)
                return StrategyState.ENOUGH();
            else
                return StrategyState.MORE(new String[]{});
        }
    }

    @Override public StrategyState next(Context ctx) {
        if (ctx.user() == null || ctx.user().requiredFactor() == null ||
            ctx.user().requiredFactor().equals(ctx.justCompletedFactor()))
            if (requiredNews("user_agreement", ctx)) return showNews("user_
↪agreement", ctx);
            else return StrategyState.ENOUGH();
        else
            return StrategyState.MORE(new String[] {});
    }

    private boolean requiredNews(final String pipeId, final Context ctx) {
        Long readOn = ctx.user().userProps().numProp("pipes.info." + pipeId + ".
↪disagreedOn");
        return (readOn == null || Instant.now().getEpochSecond() - readOn >␣
↪30*86400);
    }

    private StrategyState showNews(final String pipeId, final Context ctx) {
        String uri = "https://" + DOMAIN + "/blitz/pipes/info/start?&pipeId=" +␣
```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪pipeId + "&appId=_blitz_profile";
        Set<String> claims = new HashSet<String>(){{
            add("instanceId");
        }};
        Set<String> scopes = new HashSet<String>(){{
            add("openid");
       }};
       return StrategyState.ENOUGH_BUILDER()
         .withPipe(uri, "<CLIENT_ID>", scopes, claims)
         .build();
    }
}
```

Добавление процедуры в blitz.conf

В конфигурационном файле `blitz.conf` добавьте раздел `blitz.prod.local.idp.built-in-pipes`, в котором назначьте вспомогательному приложению с типом `info` идентификатор `id`, определенный в процедуре, и тип объявления `type`. Возможны следующие типы объявлений:

- `news` – отображается одна кнопка,
- `agreement` – отображается две кнопки.

Пример конфигурации двух вспомогательных приложений `info` с идентификаторами `alarm` и `user_agreement`:

```
"built-in-pipes": {
    "choice": [],
    "info": [
        {
            "id": "alarm",
            "type": "news",
            "userReadAttr": "userInformed"
        },
        {
            "id": "user_agreement",
            "type": "agreement"
        }
    ]
}
```

Запрос ввода пользователем атрибута или актуализации телефона и email

Процедура `PipeAttrActAdd` позволяет запросить у пользователя ввод значения атрибута. Для мобильного телефона и для email реализована периодическая актуализация контакта. Для обычного атрибута (в примере используется `family_name`) разовое заполнение атрибута. В случае если пользователь не захотел заполнять атрибут, то следующий запрос ввода атрибута реализован спустя определенное время.

Перед использованием в процедуру нужно внести следующие изменения:

- в константе `DOMAIN` указать URI, по которому из браузера пользователя доступен Blitz Identity Provider;
- в константах `MOBILE_ATTR`, `EMAIL_ATTR`, `COMMON_ATTR` указать имена заполняемых атрибутов;
- в константе `SKIP_TIME_IN_SEC` указать время, не чаще которого пользователю будут предлагать заполнить атрибут;
- в константе `ACT_TIME_IN_SEC` указать время, не чаще которого пользователю будут предлагать актуализировать телефон или email;

- в константе `ASK_AT_1ST_LOGIN` изменить значение, если запрос заполнения контакта нужно выполнять при первом же входе (обычно первый вход происходит сразу после регистрации учетной записи, потому сделана настройка, чтобы пользователю при первом входе не предлагали сразу заполнить данные);
- в теле процедуры вместо `_blitz_profile` указать идентификатор другого приложения, если изменение атрибутов должно делаться от приложения, отличного от Личного кабинета;
- настроить в сообщениях тексты для атрибута из `COMMON_ATTR` (для email и телефона также можно скорректировать тексты по умолчанию) – см. *Сообщения вспомогательных приложений (pipes)* (страница 308).

```
public class PipeAttrActAdd implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.
↪flow.dynamic");
    private final static String DOMAIN = "example.com";
    private final static String MOBILE_ATTR = "phone_number";
    private final static String EMAIL_ATTR = "email";
    private final static String COMMON_ATTR = "family_name";
    private final static Integer SKIP_TIME_IN_SEC = 30*86400;
    private final static Integer ACT_TIME_IN_SEC = 30*86400;
    private final static Boolean ASK_AT_1ST_LOGIN = false;

    @Override public StrategyBeginState begin(final Context ctx) {
        if ("login".equals(ctx.prompt())){
            List<String> methods = new ArrayList<String>(Arrays.asList(ctx.
↪availableMethods()));
            methods.remove("cls");
            return StrategyState.MORE(methods.toArray(new String[0]), true);
        } else {
            if(ctx.claims("subjectId") != null)
                return StrategyState.ENOUGH();
            else
                return StrategyState.MORE(new String[]{});
        }
    }

    @Override public StrategyState next(final Context ctx) {
        Instant instant = Instant.now();
        Boolean new_device = false;
        if (ctx.ua().getNewlyCreated() && ctx.justCompletedFactor() == 1 && !ASK_
↪AT_1ST_LOGIN){
            logger.debug("User with sub={} is signing in, pid={}, on a new device",
                ctx.claims("subjectId"), ctx.id());
            new_device = true;
        }
        Integer reqFactor = ctx.user().requiredFactor();
        if(reqFactor == null || reqFactor == ctx.justCompletedFactor()) {
            Enough.Builder en_builder = StrategyState.ENOUGH_BUILDER();
            if (MOBILE_ATTR !=null && !new_device && requireActualizeAttr(MOBILE_
↪ATTR, ctx)) {
                String uri = "https://"+DOMAIN+"/blitz/pipes/attr/act?attr="
                    +MOBILE_ATTR+"&canSkip=true&appId=_blitz_profile&verified=true
↪";
                Set<String> clms = new HashSet<String>(){{
                    add("instanceId");
                    add(MOBILE_ATTR);
                }};
                Set<String> scps = new HashSet<String>(){{
                    add("openid");
                }};
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                logger.debug("User has no {} or a non-actualzed {}, so opening pipe
↪",
                    MOBILE_ATTR, MOBILE_ATTR);
                en_builder = en_builder.withPipe(uri, "_blitz_profile", scps,␣
↪clms);
            } else if (EMAIL_ATTR !=null && !new_device &&␣
↪requireActualizeAttr(EMAIL_ATTR, ctx)) {
                String uri = "https://"+DOMAIN+"/blitz/pipes/attr/act?attr="
                    +EMAIL_ATTR+"&canSkip=true&appId=_blitz_profile&verified=true";
                Set<String> clms = new HashSet<String>(){{
                    add("instanceId");
                    add(EMAIL_ATTR);
                }};
                Set<String> scps = new HashSet<String>(){{
                    add("openid");
                }};
                logger.debug("User has no {} or a non-actualzed {}, so opening pipe
↪",
                    EMAIL_ATTR, EMAIL_ATTR);
                en_builder = en_builder.withPipe(uri, "_blitz_profile", scps,␣
↪clms);
            } else if (COMMON_ATTR !=null && !new_device &&
                    requireActualizeAttr(COMMON_ATTR, ctx)) {
                String uri = "https://"+DOMAIN+"/blitz/pipes/attr/act?attr="
                    +COMMON_ATTR+"&canSkip=true&appId=_blitz_profile";
                Set<String> clms = new HashSet<String>(){{
                    add("instanceId");
                    add(COMMON_ATTR);
                }};
                Set<String> scps = new HashSet<String>(){{
                    add("openid");
                }};
                logger.debug("User has no {}, so opening pipe", COMMON_ATTR);
                en_builder = en_builder.withPipe(uri, "_blitz_profile", scps,␣
↪clms);
            }
            return en_builder.build();
        } else {
            return StrategyState.MORE(new String[]{});
        }
    }

    private Boolean requireActualizeAttr(final String attrName, final Context ctx)
↪{
        if (attrName.equals(MOBILE_ATTR) && (ctx.passedTrack().startsWith("1:sms")␣
↪||
            ctx.passedTrack().endsWith("sms"))) {
            logger.debug("User subjectId = {}, pid = {} used SMS, so no␣
↪actualization needed",
                ctx.claims("subjectId"), ctx.id());
            return false;
        }
        if (attrName.equals(EMAIL_ATTR) && ctx.passedTrack().endsWith("email")) {
            logger.debug(
                "User subjectId = {}, pid = {} used EMAIL while auth, so no␣
↪actualization needed",
                ctx.claims("subjectId"), ctx.id());
            return false;
        }
        Long skpTime = null;
        Long actTime = null;
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        long now = Instant.now().getEpochSecond();
        if (ctx.user().userProps().numProp("pipes.act."+attrName+".skippedOn") !=␣
↪null) {
            skpTime = ctx.user().userProps().numProp("pipes.act."+attrName+".
↪skippedOn");
        }
        if (skpTime != null && ((now - skpTime) < SKIP_TIME_IN_SEC)) {
            logger.debug(
                "User subjectId = {}, pid = {} has skipped update '{}' only '{}'␣
↪seconds ago, no actualization needed", ctx.claims("subjectId"), ctx.id(),␣
↪attrName, (now - skpTime));
            return false;
        }
        if (ctx.claims(attrName) == null) return true;
        else {
            if (ctx.user().attrsCfmTimes() != null) {
                actTime = ctx.user().attrsCfmTimes().get(attrName);
            }
            if (actTime == null) return true;
            else {
                logger.debug(
                    "User subjectId = {}, pid = {} has updated '{}' '{}' seconds␣
↪ago, actualization needed = {}", ctx.claims("subjectId"), ctx.id(), attrName,␣
↪(now - actTime), ((now - actTime) > ACT_TIME_IN_SEC));
                return ((now - actTime) > ACT_TIME_IN_SEC);
            }
        }
    }
}
```

Запрос ввода пользователем контрольного вопроса

Процедура `PipeSecQuestion` проверяет, задан ли у пользователя контрольный вопрос. Если вопрос не задан, процедура запрашивает его ввод пользователем.

Перед использованием в процедуру нужно внести следующие изменения:

- в константе `DOMAIN` указать `URI`, по которому из браузера пользователя доступен Blitz Identity Provider;
- в константе `CAN_SKIP` указать режим отображения: `true` – пользователь может пропустить заполнение; `false` – пользователь должен задать значение контрольного вопроса для завершения аутентификации.

```
public class PipeSecQuestion implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.
↪flow.dynamic");
        private final static String DOMAIN = "example.com";
    private final static Boolean CAN_SKIP = true;

    @Override public StrategyBeginState begin(final Context ctx) {
        if ("login".equals(ctx.prompt())){
            List<String> methods = new ArrayList<String>(Arrays.asList(ctx.
↪availableMethods()));
            methods.remove("cls");
            return StrategyState.MORE(methods.toArray(new String[0]), true);
        } else {
            if(ctx.claims("subjectId") != null)
                return StrategyState.ENOUGH();
```

(continues on next page)

(продолжение с предыдущей страницы)

```
            else
                return StrategyState.MORE(new String[]{});
        }
    }

    @Override public StrategyState next(final Context ctx) {
        Integer reqFactor = (ctx.user() == null) ? null : ctx.user().
↪requiredFactor();
        if (reqFactor == null || reqFactor.equals(ctx.justCompletedFactor())){
                if(requireAddSecQsn(ctx)) return addSecQsn(ctx);
            else return StrategyState.ENOUGH();
        }
        else  return StrategyState.MORE(new String[]{});
    }

    private Boolean requireAddSecQsn(final Context ctx) {
        String secQsn = (ctx.user() == null) ? null : ctx.user().
↪securityQuestion();
        Long agreedOn = (ctx.user() == null) ? null : ctx.user().userProps().
↪numProp("pipes.addSecQsn.agreedOn");
        Long disagreedOn = (ctx.user() == null) ? null : ctx.user().userProps().
↪numProp("pipes.addSecQsn.disagreedOn");
        if (secQsn != null) return false;
        else if (disagreedOn == null) return true;
        else {
            long now = Instant.now().getEpochSecond();
            return ((now - disagreedOn) > 1);
        }
    }

    private StrategyState addSecQsn(final Context ctx) {
        String uri = "https://"+DOMAIN+"/blitz/pipes/secQsn/start?canSkip="+CAN_
↪SKIP+"&appId=_blitz_profile";
        Set<String> claims = new HashSet<String>(){{
          add("instanceId");
        }};
        Set<String> scopes = new HashSet<String>(){{
            add("openid");
        }};
       return StrategyState.ENOUGH_BUILDER()
         .withPipe(uri, "_blitz_profile", scopes, claims)
         .build();
    }
}
```

Регистрация ключа безопасности (WebAuthn, Passkey, FIDO2) при входе

Процедура `PipeWebAuthn` позволяет запросить у пользователя регистрацию ключа безопасности (WebAuthn, Passkey, FIDO2) при входе.

Перед использованием в процедуру нужно внести следующие изменения:

- в константе `DOMAIN` указать URI, по которому из браузера пользователя доступен Blitz Identity Provider;
- в константе `SKIP_TIME_IN_SEC` указать время, не чаще которого пользователю будут предлагать заполнить атрибут;
- в константе `ASK_AT_1ST_LOGIN` изменить значение, если запрос выпуска ключа безопасности нужно выполнять при первом же входе (обычно первый вход происходит сразу после регистрации

учетной записи, потому сделана настройка, чтобы пользователю при первом входе не предлагали сразу заполнить данные);

- в теле процедуры вместо `_blitz_profile` указать идентификатор другого приложения, если изменение атрибутов должно делаться от приложения, отличного от Личного кабинета.

```
public class PipeWebAuthn implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.
↪flow.dynamic");
    private final static String DOMAIN = "example.com";
    private final static Integer SKIP_TIME_IN_SEC = 30*86400;
    private final static Boolean ASK_AT_1ST_LOGIN = true;

    @Override public StrategyBeginState begin(final Context ctx) {
        if ("login".equals(ctx.prompt())){
            List<String> methods = new ArrayList<String>(Arrays.asList(ctx.
↪availableMethods()));
            methods.remove("cls");
            return StrategyState.MORE(methods.toArray(new String[0]), true);
        } else {
            if(ctx.claims("subjectId") != null)
                return StrategyState.ENOUGH();
            else
                return StrategyState.MORE(new String[]{});
        }
    }

    @Override
    public StrategyState next(Context ctx) {
        Boolean new_device = false;
        if (ctx.ua().getNewlyCreated() && ctx.justCompletedFactor() == 1 && !ASK_
↪AT_1ST_LOGIN){
            logger.debug("User with sub={} is signing in, pid={}, on a new device",
                ctx.claims("subjectId"), ctx.id());
            new_device = true;
        }
        if (ctx.user() == null || ctx.user().requiredFactor() == null ||
            ctx.user().requiredFactor().equals(ctx.justCompletedFactor()))
            if (!new_device && requiredWebAuthn(ctx))
                return webAuthn(ctx);
            else
                return StrategyState.ENOUGH();
        else
            return StrategyState.MORE(new String[] {});
    }

    private boolean requiredWebAuthn(final Context ctx) {
        LBrowser br = ctx.ua().asBrowser();
        String deviceType = br.getDeviceType();
        String os = br.getOsName();
        List<WakMeta> keyList = null;
        logger.trace("User subjectId = {}, pid = {} is logging using device '{}'␣
↪and OS '{}', checking configured webAuthn keys", ctx.claims("subjectId"), ctx.
↪id(), deviceType, os);
        ListResult<WakMeta> keys = ctx.dataSources().webAuthn().
↪keysOfCurrentSubject();
        if (keys != null) {
            keyList = keys.filter(k -> deviceType.equals(k.addedOnUA().
↪deviceType()))
                .filter(k -> os.equals(k.addedOnUA().osName())).list();
        }
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        if (keys != null && keyList.size() > 0) {
            logger.debug("User subjectId = {}, pid = {} has '{}' webAuthn keys for␣
↪device '{}' and OS '{}'", ctx.claims("subjectId"), ctx.id(), keyList.size(),␣
↪deviceType, os);
            return false;
        } else {
            logger.debug("User subjectId = {}, pid = {} has no configured webAuthn␣
↪keys for device '{}' and OS '{}'", ctx.claims("subjectId"), ctx.id(), deviceType,
↪ os);
        }
        Long disagreedOn = ctx.user().userProps().numProp("pipes.addKey." +␣
↪deviceType + "." + os + ".disagreedOn");
        if (disagreedOn == null) {
            return true;
        } else if (Instant.now().getEpochSecond() - disagreedOn > SKIP_TIME_IN_
↪SEC) {
            logger.debug("User subjectId = {}, pid = {} has skipped Webauthn '{}'␣
↪seconds ago, so open webAuthn pipe", ctx.claims("subjectId"), ctx.id(), (Instant.
↪now().getEpochSecond() - disagreedOn));
            return true;
        } else {
            logger.debug("User subjectId = {}, pid = {} has skipped Webauthn '{}'␣
↪seconds ago, no need to open webAuthn pipe", ctx.claims("subjectId"), ctx.id(),␣
↪(Instant.now().getEpochSecond() - disagreedOn));
            return false;
        }
    }

    private StrategyState webAuthn(final Context ctx) {
        String uri = "https://"+DOMAIN+"/blitz/pipes/conf/webAuthn/start?&
↪canSkip=true&appId=_blitz_profile";
        Set<String> claims = new HashSet<String>(){{
            add("instanceId");
        }};
        Set<String> scopes = new HashSet<String>(){{
            add("openid");
        }};
        Map<String, Object> urParams = new HashMap<String, Object>();
        return StrategyState.ENOUGH_BUILDER()
            .withPipe(uri, "_blitz_profile", scopes, claims).build();
    }
}
```

Отображение пользователю списка выбора значений при входе

Можно настроить, чтобы при входе Blitz Identity Provider показал пользователю окно выбора из списка значений и сохранил результат выбора в атрибуте в учетной записи пользователя.

Процедура

Процедура `ChoicePipe` позволяет показывать пользователю при входе страницы выбора списка значений. Перед использованием необходимо внести следующие изменения:

- в константе `DOMAIN` вместо `<BLITZ-HOST>` указать URI, по которому из браузера пользователя доступен Blitz Identity Provider, а в константе `CLIENT_ID` вместо `<CLIENT_ID>` указать идентификатор приложения (с правами на `scope openid`), от имени которого будет выполняться вспомогательное приложение;
- *настроить в конфигурационном файле тип уведомления* (страница 279);
- *настроить в сообщениях текст уведомления и названия кнопок* (страница 308).

```
public class ChoicePipe implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.
↪flow.dynamic");

    private final static String DOMAIN = "<BLITZ-HOST>";
    private final static String CLIENT_ID = "<CLIENT_ID>";

    @Override public StrategyBeginState begin(final Context ctx) {
        if ("login".equals(ctx.prompt())){
            List<String> methods = new ArrayList<String>(Arrays.asList(ctx.
↪availableMethods()));
            methods.remove("cls");
            return StrategyState.MORE(methods.toArray(new String[0]), true);
        } else {
            if(ctx.claims("subjectId") != null)
                return StrategyState.ENOUGH();
            else
                return StrategyState.MORE(new String[]{});
        }
    }

    @Override
    public StrategyState next(Context ctx) {
        List<List<String>> choice = new ArrayList<List<String>>(){};
        choice.add(Arrays.asList("Value 1"));
        choice.add(Arrays.asList("Value 2"));
        try {
            if (ctx.user() == null || ctx.user().requiredFactor() == null
                    || ctx.user().requiredFactor().equals(ctx.
↪justCompletedFactor())) {
                String res = new ObjectMapper().writeValueAsString(choice);
                String choiceJson = Base64.getUrlEncoder().encodeToString(res.
↪getBytes("UTF-8"));
                return choice(ctx, choiceJson);
            }
            else
                return StrategyState.MORE(new String[] {});
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }

    private StrategyState choice(final Context ctx, final String choiceJson) {
        String uri = "https://" + DOMAIN + "/blitz/pipes/choice/start?appId=" +␣
↪CLIENT_ID + "&pipeId=select_value&choices=" + choiceJson;
        Set<String> claims = new HashSet<String>(){{
```

(continues on next page)

(продолжение с предыдущей страницы)

```
            add("instanceId");
        }};
        Set<String> scopes = new HashSet<String>(){{
            add("openid");
        }};
        return StrategyState.ENOUGH_BUILDER()
          .withPipe(uri, CLIENT_ID, scopes, claims)
          .build();
    }
}
```

Добавление процедуры в blitz.conf

В конфигурационном файле `blitz.conf` добавьте раздел `blitz.prod.local.idp.built-in-pipes`, в котором назначьте вспомогательному приложению с типом `choice` идентификатор `id`, определенный в процедуре, и имя атрибута `claim`, в который необходимо сохранять результат выбора.

Пример конфигурации вспомогательного приложения `choice`:

```
"built-in-pipes": {
    "choice": [
        {
            "id": "select_value",
            "claim": "role"
        }
    ]
}
```

2.4.3 Функции и методы различного назначения в процедурах входа

Данный раздел содержит примеры функций и методов, которые вы можете использовать при написании процедур входа в Blitz Identity Provider.

См.также:

Для вашего удобства Blitz Identity Provider также предоставляет набор *готовых процедур* (страница 263).

Получение геоданных пользователя

Процедуру входа можно использовать для получения данных о стране и городе, в которых находится пользователь, и на основании этого гибко настраивать правила входа, например, вводить запрет на вход из-за рубежа, активировать второй фактор аутентификации и др.

Для этого в процедурах входа используйте следующие классы и методы:

1. Класс `LGeoData` с функциями `getCountry()` и `getCity()`.

```
public class LGeoData {
    /**
    *  Get IP address country
    *
    * @return - country or null if country not specified.
    */
    public final String getCountry();
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    /**
    *  Get IP address city
    *
    * @return - city or null if city not specified.
    */

    public final String getCity();
}
```

2. Метод `geoData()` в `Context`.

```
/**
 * Get geo data of user IP address
 *
 * @return - geo data.
 */
 LGeoData geoData();
```

Важно: Для работы метода необходимо импортировать класс `LGeoData`.

```
import com.identityblitz.idp.login.authn.flow.LGeoData
```

Список 13: Пример кода, который выводит в лог страну и город пользователя

```
import com.identityblitz.idp.login.authn.flow.LGeoData;

LGeoData geoData = _ctx.geoData();
String country = geoData.getCountry();
logger.trace("IP location: country - {}, city - {}, factor - {}", country ,␣
↪geoData.getCity());
```

Список 14: Пример процедуры, включающей 2FA для определенных стран

```
package com.identityblitz.idp.flow.dynamic;

import java.lang.*;
import java.util.*;
import java.text.*;
import java.time.*;
import java.math.*;
import java.security.*;
import javax.crypto.*;
import org.slf4j.LoggerFactory;
import org.slf4j.Logger;
import org.codehaus.jackson.map.ObjectMapper;
import org.codehaus.jackson.type.TypeReference;
import com.identityblitz.idp.login.authn.flow.api.*;
import com.identityblitz.idp.login.authn.flow.Context;
import com.identityblitz.idp.login.authn.flow.Strategy;
import com.identityblitz.idp.login.authn.flow.StrategyState;
import com.identityblitz.idp.login.authn.flow.StrategyBeginState;
import com.identityblitz.idp.login.authn.flow.LCookie;
import com.identityblitz.idp.login.authn.flow.LUserAgent;
import com.identityblitz.idp.login.authn.flow.LBrowser;
```

(continues on next page)

(продолжение с предыдущей страницы)

```
import com.identityblitz.idp.login.authn.flow.LGeoData;
import com.identityblitz.idp.federation.matching.JsObj;
import com.identityblitz.idp.flow.common.api.*;
import com.identityblitz.idp.flow.dynamic.*;
import java.util.function.Predicate;
import java.util.stream.Stream;
import java.util.stream.Collectors;
import java.lang.invoke.LambdaMetafactory;
import java.util.function.Consumer;
import static com.identityblitz.idp.login.authn.flow.StrategyState.*;

public class EnableSecondFactorByCountry implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.
↪flow.dynamic");

    @Override public StrategyBeginState begin(final Context ctx) {
        if ("login".equals(ctx.prompt())){
            List<String> methods = new ArrayList<String>(Arrays.asList(ctx.
↪availableMethods()));
            methods.remove("cls");
            return StrategyState.MORE(methods.toArray(new String[0]), true);
        } else {
            if(ctx.claims("subjectId") != null)
                return StrategyState.ENOUGH();
            else
                return StrategyState.MORE(new String[]{});
        }
    }

    @Override public StrategyState next(final Context ctx) {
        Integer reqFactor = (ctx.user() == null) ? null : ctx.user().
↪requiredFactor();
        LGeoData geoData = ctx.geoData();
        String country = geoData.getCountry();
        logger.info("IP location: country - {}, city - {}, factor - {}", country ,␣
↪geoData.getCity());
        if(ctx.justCompletedFactor() == 1 && (country == null || !country.equals(
↪"Russia")))
            return StrategyState.MORE(new String[]{});
        else
            return StrategyState.ENOUGH();
    }
}
```

Сброс сессии пользователя

В процедуре входа можно вызвать сброс сессии пользователя при определенных условиях. Для этого используется функция `StrategyState.MORE_BUILDER()` со следующими методами:

- `setResetSession(reset: Boolean): true` - сбросить сессию, `false` - не сбрасывать (по умолчанию `false`).
- `isResetSession()`: позволяет узнать, установлен ли сброс сессии.

Пример ниже содержит скрипт для сброса сессии, если `ctx.prompt=login`:

```
package com.identityblitz.idp.flow.dynamic;

import java.lang.*;
```

(continues on next page)

(продолжение с предыдущей страницы)

```
import java.util.*;
import java.text.*;
import java.time.*;
import java.math.*;
import java.security.*;
import javax.crypto.*;
import org.slf4j.LoggerFactory;
import org.slf4j.Logger;
import org.codehaus.jackson.map.ObjectMapper;
import org.codehaus.jackson.type.TypeReference;
import com.identityblitz.idp.login.authn.flow.api.*;
import com.identityblitz.idp.login.authn.flow.Context;
import com.identityblitz.idp.login.authn.flow.Strategy;
import com.identityblitz.idp.login.authn.flow.StrategyState;
import com.identityblitz.idp.login.authn.flow.StrategyBeginState;
import com.identityblitz.idp.login.authn.flow.LCookie;
import com.identityblitz.idp.login.authn.flow.LUserAgent;
import com.identityblitz.idp.login.authn.flow.LBrowser;
import com.identityblitz.idp.login.authn.flow.LGeoData;
import com.identityblitz.idp.federation.matching.JsObj;
import com.identityblitz.idp.flow.common.api.*;
import com.identityblitz.idp.flow.dynamic.*;
import java.util.function.Predicate;
import java.util.stream.Stream;
import java.util.stream.Collectors;
import java.lang.invoke.LambdaMetafactory;
import java.util.function.Consumer;
import static com.identityblitz.idp.login.authn.flow.StrategyState.*;

public class ResetSession implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.
↪flow.dynamic");

    @Override public StrategyBeginState begin(final Context ctx) {
        if ("login".equals(ctx.prompt())){
            List<String> methods = new ArrayList<String>(Arrays.asList(ctx.
↪availableMethods()));
            methods.remove("cls");
            logger.info("### RESET_SESSION");
            return StrategyState.MORE_BUILDER().setResetSession(true).
↪addMethods(methods.toArray(new String[0])).build();
        } else {
            if(ctx.claims("subjectId") != null)
                return StrategyState.ENOUGH();
            else
                return StrategyState.MORE(new String[]{});
        }
    }

    @Override public StrategyState next(final Context ctx) {
        Integer reqFactor = (ctx.user() == null) ? null : ctx.user().
↪requiredFactor();
        if(reqFactor == null || reqFactor == ctx.justCompletedFactor())
            return StrategyState.ENOUGH();
        else
            return StrategyState.MORE(new String[]{});
    }
}
```

Кастомные ошибки и их вызов в скрипте

Blitz Identity Provider позволяет создавать кастомные ошибки и вызывать их в процедурах входа. Выполните следующие действия:

1. Руководствуясь *инструкцией* (страница 301), добавьте кастомное сообщение об ошибке в файл `messages` в директории `/usr/share/identityblitz/blitz-config/custom_messages`.

```
err.bad_gateway=Недоступно
```

2. Вызовите данную ошибку при получении `HTTP 502`.

```
if (result.status() == 502) {
                        return HttpLoop.error("bad_gateway",
                                              Collections.<String, String>
↪singletonMap("status", "" + result.status()));
}
```

Пример скрипта, вызывающего кастомную ошибку HTTP 502 для метода *Flash Call* (страница 137):

```
package flashcall;

import com.identityblitz.core.loop.http.HttpLoop;
import com.identityblitz.core.loop.http.HttpLoopRequest;
import com.identityblitz.core.loop.http.HttpLoopResult;
import com.identityblitz.core.loop.JsObj;
import java.util.Collections;

public class FlashCallFlow implements HttpLoop {

        public HttpLoopRequest run(final JsObj obj, final HttpLoopResult␣
↪result) {
                if (result == null) {
                        final String number = obj.asString("phone_number");
                        return HttpLoop.callBuilder("POST", "http://test.
↪flashcall.ru/api/v1")
                                              .withHeader("Token", "1234567890")
                                              .withBody(JsObj.empty.addString("id",
↪"test").addString("dst_number", number.substring(number.length() - 10)))
                                              .build(JsObj.empty);
                } else if (result.status() == 200) {
                        final JsObj body = result.body();
                        return HttpLoop.Ok(JsObj.empty.addString("code", body.
↪asString("SenderID")));
                } else if (result.status() == 502) {
                        return HttpLoop.error("bad_gateway",
                                              Collections.<String, String>
↪singletonMap("status", "" + result.status()));
                } else {
                        return HttpLoop.error("wrong_http_status",
                                              Collections.<String, String>
↪singletonMap("status", "" + result.status()));
                }
        }
}
```

Анализ меток приложений

Blitz Identity Provider позволяет присваивать приложениям *метки* (страница 224) и на их основании задавать в процедурах входа логику работы с отмеченными приложениями.

Для получения меток приложений в процедуре используется метод `ctx.appTags()` в `Context`.

Внимание: Для работы метода необходимо импортировать `java.util.Set`.

Пример процедуры получения метки `2F` и использования ее при включении входа по второму фактору:

```
package com.identityblitz.idp.flow.dynamic;

import java.lang.*;
import java.util.*;
import java.text.*;
import java.time.*;
import java.math.*;
import java.security.*;
import javax.crypto.*;
import org.slf4j.LoggerFactory;
import org.slf4j.Logger;
import org.codehaus.jackson.map.ObjectMapper;
import org.codehaus.jackson.type.TypeReference;
import com.identityblitz.idp.login.authn.flow.api.*;
import com.identityblitz.idp.login.authn.flow.Context;
import com.identityblitz.idp.login.authn.flow.Strategy;
import com.identityblitz.idp.login.authn.flow.StrategyState;
import com.identityblitz.idp.login.authn.flow.StrategyBeginState;
import com.identityblitz.idp.login.authn.flow.LCookie;
import com.identityblitz.idp.login.authn.flow.LUserAgent;
import com.identityblitz.idp.login.authn.flow.LBrowser;
import com.identityblitz.idp.login.authn.flow.LGeoData;
import com.identityblitz.idp.federation.matching.JsObj;
import com.identityblitz.idp.flow.common.api.*;
import com.identityblitz.idp.flow.dynamic.*;
import java.util.function.Predicate;
import java.util.stream.Stream;
import java.util.stream.Collectors;
import java.lang.invoke.LambdaMetafactory;
import java.util.function.Consumer;
import static com.identityblitz.idp.login.authn.flow.StrategyState.*;

public class UseAppTags implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.
↪flow.dynamic");

    @Override public StrategyBeginState begin(final Context ctx) {
        if ("login".equals(ctx.prompt())){
            List<String> methods = new ArrayList<String>(Arrays.asList(ctx.
↪availableMethods()));
            methods.remove("cls");
            return StrategyState.MORE(methods.toArray(new String[0]), true);
        } else {
            if(ctx.claims("subjectId") != null)
                return StrategyState.ENOUGH();
            else
                return StrategyState.MORE(new String[]{});
        }
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    }

    @Override public StrategyState next(final Context ctx) {
        Set<String> tags = ctx.appTags();
        logger.info("APP TAGS: " + tags);
        if (ctx.justCompletedFactor() == 1 && tags.contains("2F"))
            return StrategyState.MORE(new String[]{});
        else
            return StrategyState.ENOUGH();
    }
}
```

2.4.4 Кастомизация логики операций с хранилищами данных

Принцип кастомизации

Blitz Identity Provider позволяет кастомизировать логику операций с хранилищами данных. Для этого используется Java-класс с фиксированным именем и пакетом `com.identityblitz.idp.store.id.logic.dynamic`.

Существуют восемь пользовательских процедур, по одной для каждой операции с фиксированным именем класса:

- `searchUser` — `CustomSearchUsersLogic.java`
- `getUser` — `CustomGetUserLogic.java`
- `findUser` — `CustomFindUserLogic.java`
- `bindUser` — `CustomBindUserLogic.java`
- `changeUserPassword` — `CustomChangeUserPasswordLogic.java`
- `addUser` — `CustomAddUserLogic.java`
- `updateUser` — `CustomUpdateUserLogic.java`
- `deleteUser` — `CustomDeleteUserLogic.java`

Конфигурация

Для того чтобы настроить пользовательскую логику для нужных операций, выполните следующие действия:

1. Поместите Java-файлы с пользовательской логикой в каталог:

   ```
   /usr/share/identityblitz/blitz-config/dynamic/idstore/<operation_name_in_
   ↪lowercase>
   ```

   Например, для чтобы включить пользовательскую логику для `searchUsers` и `bindUser`, поместите файлы `CustomSearchUsersLogic.java` и `CustomBindUserLogic.java` в каталоги `/usr/share/identityblitz/blitz-config/dynamic/idstore/searchusers` и `/usr/share/identityblitz/blitz-config/dynamic/idstore/binduser` соответственно.

2. Откройте файл конфигурации `/usr/share/identityblitz/blitz-config/blitz.conf`.

   ```
   sudo vim /usr/share/identityblitz/blitz-config/blitz.conf
   ```

3. Добавьте новый блок `logic` в блок конфигурации `blitz.prod.local.idp.id-stores`. Блок должен содержать имена кастомизируемых операций, указанных в качестве ключа и секцию `{"enabled": true}` в качестве значения ключа.

Список 15: Кастомизация операций searchUsers и bindUser

```
{
  "logic": {
    "searchUsers": {
      "enabled": true
    },
    "bindUser": {
      "enabled": true
    }
  }
}
```

Написание пользовательской процедуры

Пользовательские процедуры для всех операций имеют одинаковую спецификацию, но используют собственный контекст и служебные функции. Каждый метод в процедурах соответствует определенному состоянию процесса выполнения операции. В методах необходимо реализовать логику перехода к следующему циклу (с последующим вызовом нового метода) или завершения операции.

Каждый метод в процедуре возвращает пару `LoopOutput` и `OperationState`. `LoopOutput` может быть:

1. терминальный – завершает логический цикл работы одним из следующих способов:
    - ошибка;
    - успех (результат успеха для определенной операции);
    - заключительная операция сохранения (выполнить операцию сохранения с некоторыми параметрами и завершить с результатом).
2. задача - требуется больше итераций цикла:
    - запрос к хранилищу на выполнение определенной операции;
    - запрос к внешнему веб-сервису.

На данный момент механизм работы пользовательских процедур находится на стадии бета-тестирования. Вы можете запросить подробную спецификацию Java и получить консультацию по возможностям кастомизации в своей среде у наших технических специалистов по адресу support@idblitz.ru.

2.4.5 Процедуры привязки аккаунтов внешних поставщиков

Помимо *базовой* (страница 186) настройки, привязку учетных записей для каждого внешнего поставщика идентификации можно настроить с помощью процедуры на языке программирования Java. Данный режим обеспечивает максимальную гибкость настройки и подходит для узкоспециализированных сценариев связывания учетных записей и сопоставления атрибутов.

Настройка доступна в разделе Поставщики идентификации -> Связывание учетных записей -> Расширенная настройка. Для написания собственной процедуры следуйте инструкциям в базовой процедуре, а также рекомендациям из данного раздела документации.

Связывание учетных записей

Базовая настройка | Расширенная настройка

Процедура связывания УЗ

> Для успешной работы процедуры связывания необходимо написать на языке `Java` класс, наследующий абстрактный класс `MatchingBlock`. Название класса должно быть `Esia_1Esia`. Класс должен иметь публичный `default` конструктор. В целях безопасности загрузка класса осуществляет отдельный `class loader` с ограниченным списком `imports`. Вся необходимая информация передается в параметры функций.

```
package com.identityblitz.idp.federation.matching.dynamic;

import java.lang.*;
import java.util.*;
import java.text.*;
import java.time.*;
import java.math.*;
import java.security.*;
import javax.crypto.*;
import org.slf4j.LoggerFactory;
import org.slf4j.Logger;
import com.identityblitz.idp.federation.*;
import com.identityblitz.idp.federation.matching.*;
import com.identityblitz.idp.flow.common.api.*;
import com.identityblitz.idp.flow.common.model.*;
import com.identityblitz.idp.federation.matching.dynamic.*;
import java.util.function.Consumer;
import java.util.stream.Stream;
import java.util.stream.Collectors;
import org.codehaus.jackson.map.ObjectMapper;
import org.codehaus.jackson.type.TypeReference;
import com.identityblitz.idp.extensions.types.JsObject;

import com.identityblitz.idp.federation.matching.*;
import com.identityblitz.idp.flow.common.api.HttpFactory;

/**
 * Класс наследуется от MatchingBlock и для корректного инстантирования должен иметь default конструктор.
 * Текущая сгенерированная реализация обеспечивает стратегию при которой пользователи не сопоставляются и не обновляются.
 */
public class Esia_1Esia extends MatchingBlock {

  private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.federation.matching.dynamic");

  /**
     * Итеративная функция определяющая соответствие внетренних УЗ и УЗ поставщика идентификации.
     * На каждой итерации функция может выполнить операцию find (найденые пользователи будут переданы в следующей итерации)
     * или завершить операцию со следующими решениями:
     *    matched - соответствующие пользователи найдены;
     *    matchError - ошибка определения соответствия пользователей;
     *    matchByLogin - осуществить связку с пользователем, который успешно аутентифицируется;
     *    refine - получет список пользователей, запрашивает пароль и осуществляет связку с тем пользователем, для которого введен корректный пароль;
     * @param ctx - контекст процедур со следующими полями:
     *    iteration - номер итерации процедуры;
     *    extAttrs - атрибуты пользователя, полученные от поставщива идентификации;
     *    sid - уникальный идентификатор внешней УЗ.
     * @param users - пользователи.
     * @return - одно из перечисленных решений
     */
  @Override public MatchResult match(MatchingContext ctx, List<MatchingUser> users){
    return matchError(ctx, new MatchingError("not_matched", "User not matched"));
  };

  /**
     * Возвращает обновляемые и удаляемые атрибуты.
     * @param extAttrs - атрибуты пользователя, полученные от поставщива идентификации.
     * @param user - внутренний пользователь.
     * @param justMatched - признак того, что связь внутренних УЗ с УЗ внешнего поставщика установлена впервые.
     * @return - кортеж с изменяемыми и удаляемыми атрибутами. Например: change(JsObj.empty(), Collections.<String>emptySet())
     */
  @Override public Tuple2<JsObj, Set<String>> update(JsObj extAttrs, MatchingUser user, Boolean justMatched, HttpFactory httpFactory){
    return change(JsObj.empty(), Collections.<String>emptySet());
  };
}

```

Отмена | Сохранить

Регистрация пользователя во внешнем поставщике

В форме ввода логина и пароля для авторизации через внешнего поставщика может отображаться ссылка на страницу регистрации во внешнем поставщике (Нет аккаунта? Зарегистрироваться). Для того чтобы ссылка не отображалась, функции `refine` и `matchByLogin` в процедуре привязки должны быть вызваны без указания параметров для регистрации.

- `refine(cxt, users)` вместо `refine(cxt, users, regAttrs);`
- `matchByLogin(cxt)` вместо `matchByLogin(cxt, regAttrs).`

Пример использования в процедуре:

```
package com.identityblitz.idp.federation.matching.dynamic;

import java.lang.*;
import java.util.*;
import java.text.*;
import java.time.*;
import java.math.*;
import java.security.*;
import javax.crypto.*;
import org.slf4j.LoggerFactory;
import org.slf4j.Logger;
import com.identityblitz.idp.federation.*;
import com.identityblitz.idp.federation.matching.*;
import com.identityblitz.idp.flow.common.api.*;
import com.identityblitz.idp.flow.common.model.*;
import com.identityblitz.idp.federation.matching.dynamic.*;
import java.util.function.Consumer;
import java.util.stream.Stream;
import java.util.stream.Collectors;
import org.codehaus.jackson.map.ObjectMapper;
import org.codehaus.jackson.type.TypeReference;
import com.identityblitz.idp.extensions.types.JsObject;
import com.identityblitz.idp.federation.matching.*;
import com.identityblitz.idp.flow.common.api.HttpFactory;

public class Esia_1Esia extends MatchingBlock {

  private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.
↪federation.matching.dynamic");

  @Override public MatchResult match(MatchingContext ctx, List<MatchingUser> users)
↪{
    if (ctx.iteration() == 1) {
      return find(ctx, MatchingFilter.empty().eq("uid", "00000").or().eq("uid",
↪"test@test.ru"));
    } else {
      //return refine(ctx, Collections.singletonList((users.get(0))));
      //return refine(ctx, users);
      return matchByLogin(ctx);
    }
  };

  @Override public Tuple2<JsObj, Set<String>> update(JsObj extAttrs, MatchingUser␣
↪user, Boolean justMatched, HttpFactory httpFactory){
    return change(JsObj.empty(), Collections.<String>emptySet());
  };
}
```

Вычисление имени внешней учетной записи

В процедуре связывания можно при каждом входе через внешнего поставщика вычислять имя внешней учетной записи и обновлять соответствующий параметр в базе данных. Для этого используется функция `updateFederatedAccountName`.

Пример использования в процедуре:

```
package com.identityblitz.idp.federation.matching.dynamic;

import java.lang.*;
import java.util.*;
import java.text.*;
import java.time.*;
import java.math.*;
import java.security.*;
import javax.crypto.*;
import org.slf4j.LoggerFactory;
import org.slf4j.Logger;
import com.identityblitz.idp.federation.*;
import com.identityblitz.idp.federation.matching.*;
import com.identityblitz.idp.flow.common.api.*;
import com.identityblitz.idp.flow.common.model.*;
import com.identityblitz.idp.federation.matching.dynamic.*;
import java.util.function.Consumer;
import java.util.stream.Stream;
import java.util.stream.Collectors;
import org.codehaus.jackson.map.ObjectMapper;
import org.codehaus.jackson.type.TypeReference;
import com.identityblitz.idp.extensions.types.JsObject;

import com.identityblitz.idp.federation.matching.*;
import com.identityblitz.idp.flow.common.api.HttpFactory;

public class Esia_1Esia extends MatchingBlock {

  private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.
↪federation.matching.dynamic");

  @Override public MatchResult match(MatchingContext ctx, List<MatchingUser> users)
↪{

    if (ctx.iteration() == 1) {
      //return matchError(ctx, new MatchingError("bad_err_code","bad_err_msg"));
      return tryToSearch(ctx);
    }

    if (users.isEmpty()) {
      return matchError(ctx, new MatchingError("error","error"));
    }

    if (users.size() == 1) {
      return matched(ctx, users.get(0));
    }

    return refine(ctx, users, ctx.extAttrs());
  };

  private MatchResult tryToSearch(MatchingContext ctx) {
    return find(ctx, filter().eq("uid", "test@test.ru"));
  }
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  @Override public Tuple2<JsObj, Set<String>> update(JsObj extAttrs, MatchingUser␣
↪user, Boolean justMatched, HttpFactory httpFactory){
    return change(JsObj.empty(), Collections.<String>emptySet());
  };

  @Override  public boolean isAllowMultiBind() {
    return true;
  }

  @Override public String updateFederatedAccountName(JsObj extAttrs){
    if (extAttrs.contains("firstName") && extAttrs.contains("lastName")){
      String name = extAttrs.asString("firstName") + " " + extAttrs.asString(
↪"lastName");
      if (extAttrs.contains("middleName")) {
        name = name + " " + extAttrs.asString("middleName");
      }
      return name;
    } else {
      // don't update federated account name
      return super.updateFederatedAccountName(extAttrs);
    }
  };
}
```

2.5 Дизайн и тексты интерфейса

2.5.1 Страница входа

Предупреждение: Администратор консоли управления должен самостоятельно проверять корректность помещаемых на страницу входа JS-скриптов и содержимое страниц входа на предмет возможных уязвимостей.

В разделе *«Внешний вид»* консоли управления администратор может настроить параметры отображения единой страницы входа. Если применяются приложения Blitz Identity Provider по регистрации пользователей и восстановлению пароля, то их внешний вид также будет соответствовать заданным настройкам внешнего вида единой страницы входа.

При входе в раздел *«Внешний вид»* отображается перечень настроенных шаблонов страницы входа. Каждый шаблон описывается:

- идентификатором;
- названием;
- перечнем приложений;
- описанием.

По умолчанию создан шаблон с идентификатором `default` – он используется для всех приложений, подключенных к Blitz Identity Provider, а также для страниц единого логаута.

Редактирование шаблона по умолчанию осуществляется с помощью специального конструктора (подробнее ниже).

Также имеется возможность:

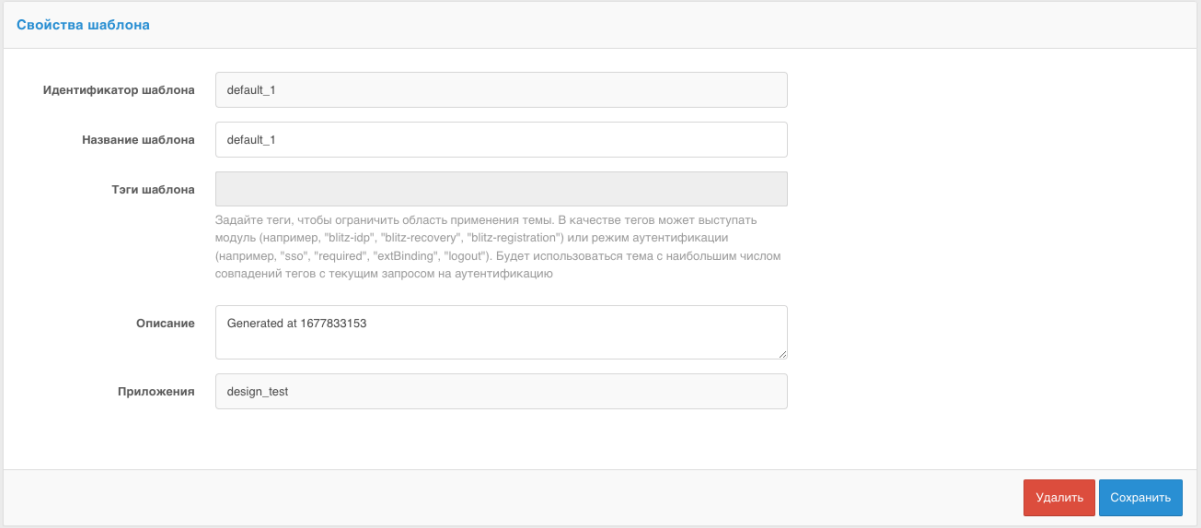
- создавать и изменять новые шаблоны с помощью конструктора и назначать их разным приложениям;

- создавать и изменять новые шаблоны в ручном режиме.

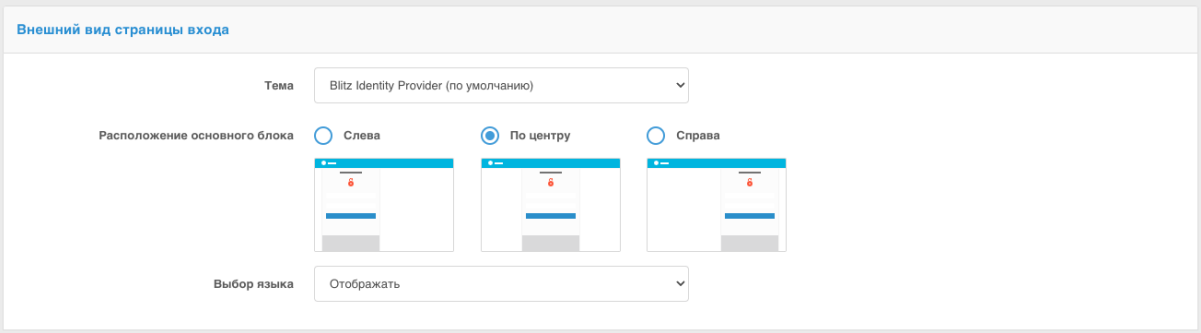
Редактирование шаблона по умолчанию

При открытии страницы редактирования шаблона по умолчанию отображается информация о самом шаблоне (идентификатор шаблона, название шаблона, описание и перечень приложений), а также интерфейс конструктора страницы входа.

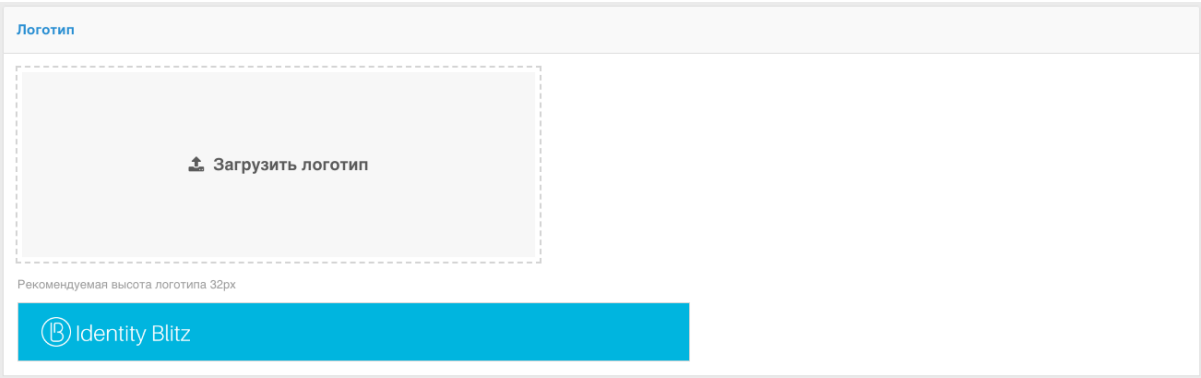
Настройка внешнего вида страницы входа (свойства шаблона):



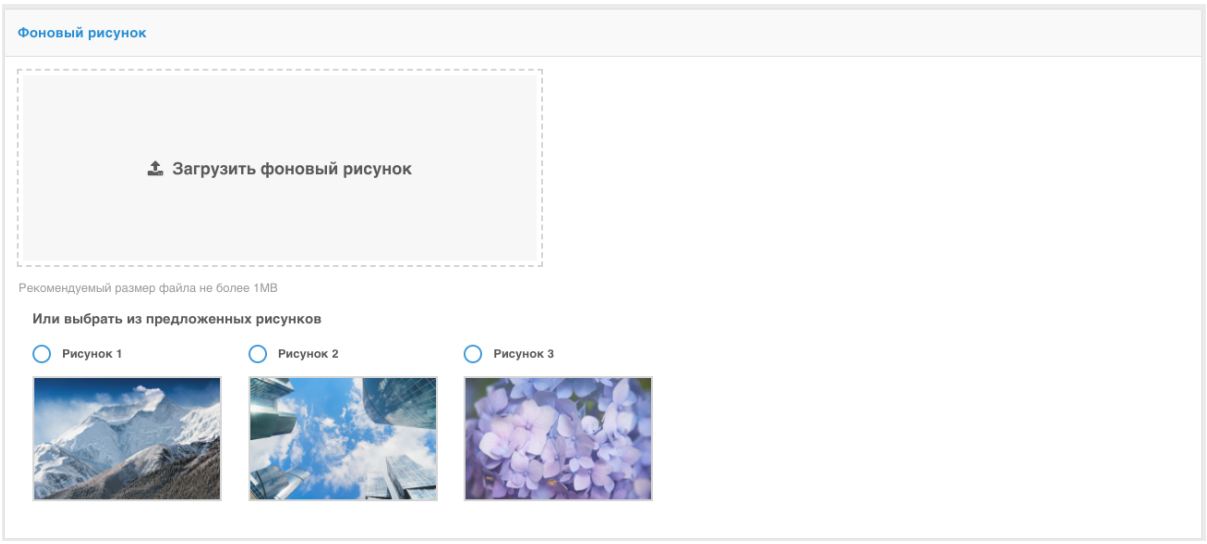
Настройка внешнего вида страницы входа (внешний вид страницы входа):



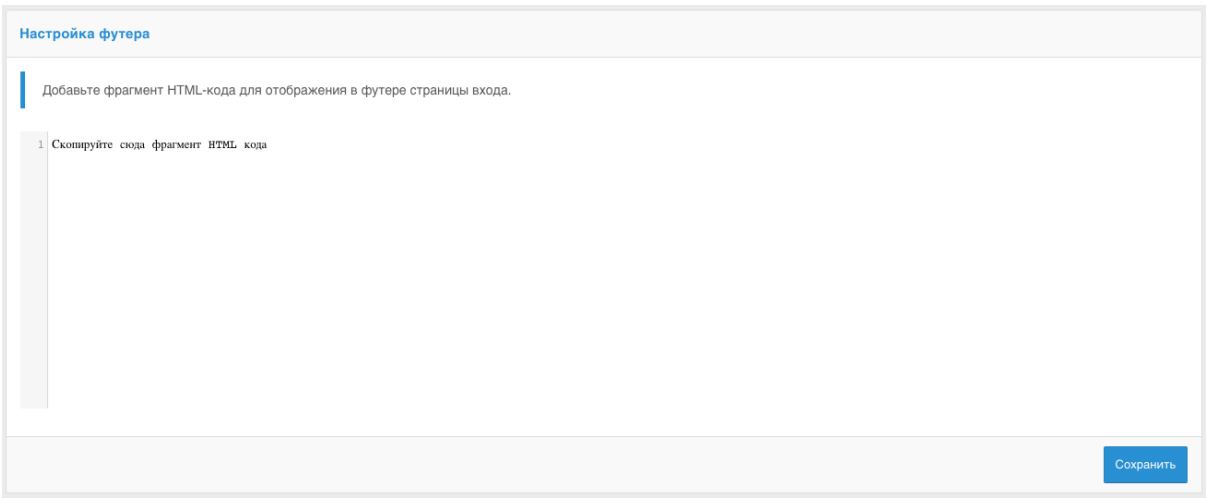
Настройка внешнего вида страницы входа (логотип):



Настройка внешнего вида страницы входа (фоновый рисунок):



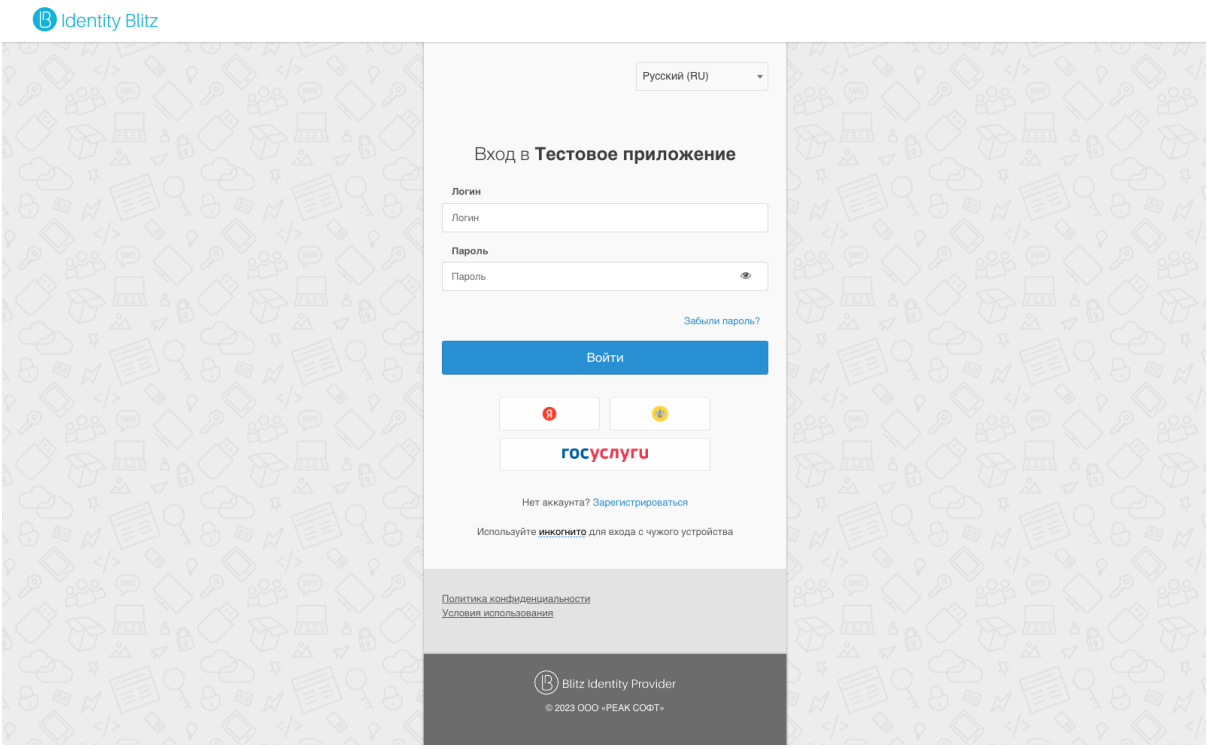
Настройка внешнего вида страницы входа (настройка футера):

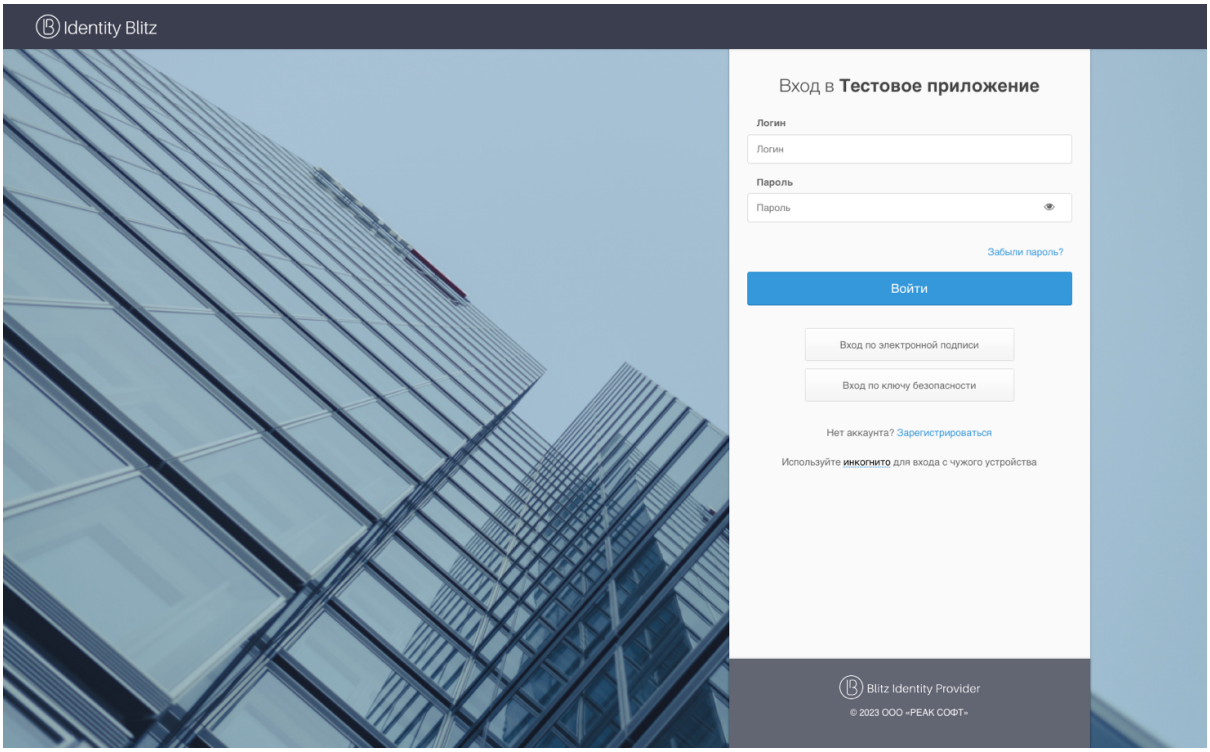


В стандартной поставке конструктор Blitz Identity Provider предоставляет следующие возможности:

- три цветовых темы оформления элементов интерфейса;
- возможность определить местоположения блока ввода сведений (идентификации и аутентификации, регистрации, восстановления пароля);
- возможность загрузки логотипа компании для отображения в заголовке страницы;
- выбор фонового рисунка (можно выбрать из 3 стандартных рисунков в каждой теме оформления, либо загрузить свой собственный фоновый рисунок);
- настройка содержания футера страницы входа.

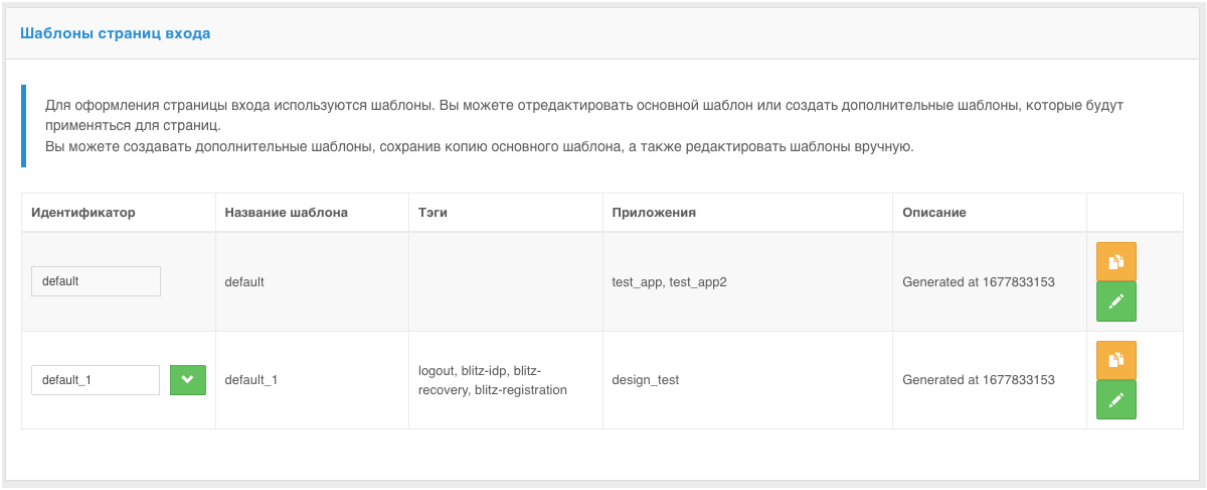
На рисунках ниже приведены некоторые примеры страниц входа, полученных в результате стандартной настройки.


Identity Blitz
Русский (RU)
Вход в Тестовое приложение
Логин
Логин
Пароль
Пароль
Забыли пароль?
Войти
госуслуги
Нет аккаунта? Зарегистрироваться
Используйте инкогнито для входа с чужого устройства
Политика конфиденциальности
Условия использования
Blitz Identity Provider
© 2023 ООО «РЕАК СОФТ»

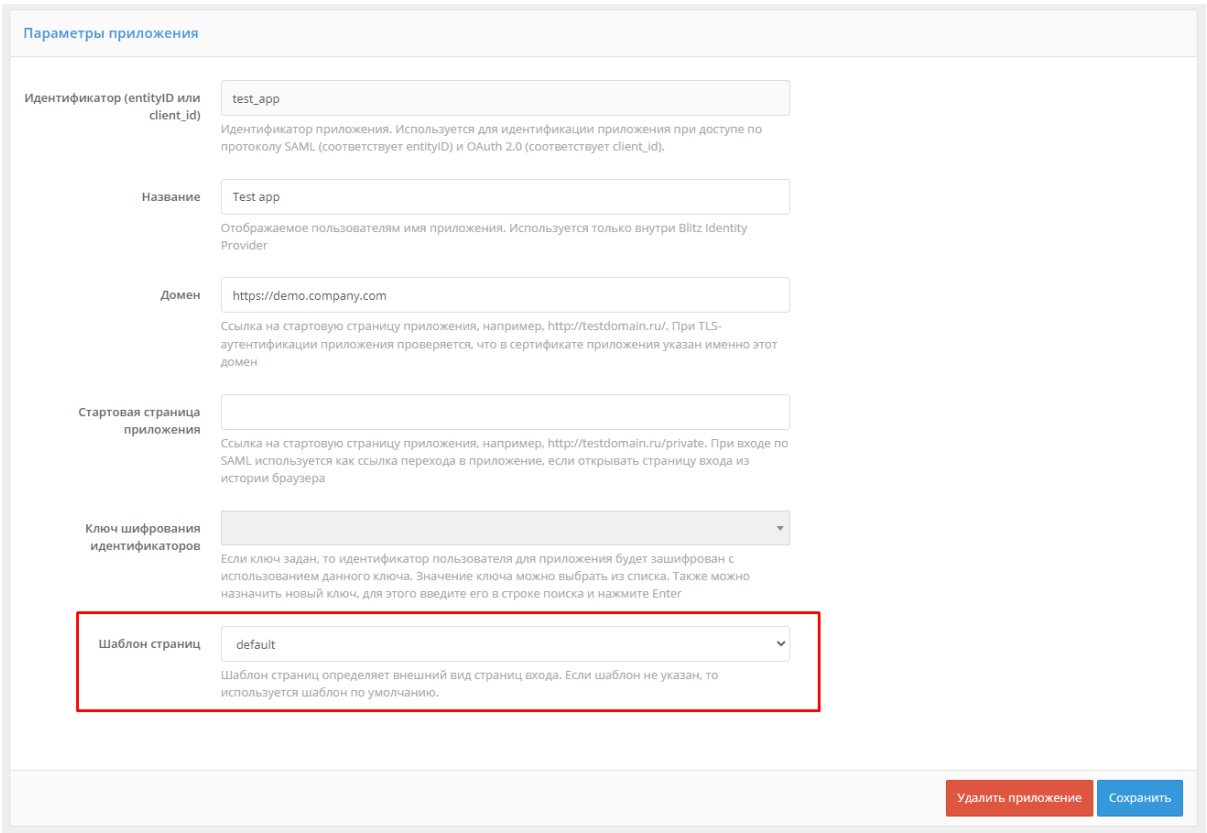

Identity Blitz
Вход в Тестовое приложение
Логин
Логин
Пароль
Пароль
Забыли пароль?
Войти
Вход по электронной подписи
Вход по ключу безопасности
Нет аккаунта? Зарегистрироваться
Используйте инкогнито для входа с чужого устройства
Blitz Identity Provider
© 2023 ООО «РЕАК СОФТ»

Создание и изменение новых шаблонов с помощью конструктора

Blitz Identity Provider позволяет настроить разный вид страниц входа для случая входа пользователя в различные подключенные приложения. Для этого необходимо создавать новые шаблоны входа – проще всего это сделать на базе существующего default-шаблона, нажав на кнопку *«Копировать»*. После этого будет создан новый шаблон, который можно редактировать с помощью конструктора.



Чтобы новый шаблон использовался при входе в некоторое приложение, необходимо в разделе *«Приложения»* перейти к редактированию нужного приложения и выбрать требуемый шаблон страниц.



Создание и изменение новых шаблонов в ручном режиме

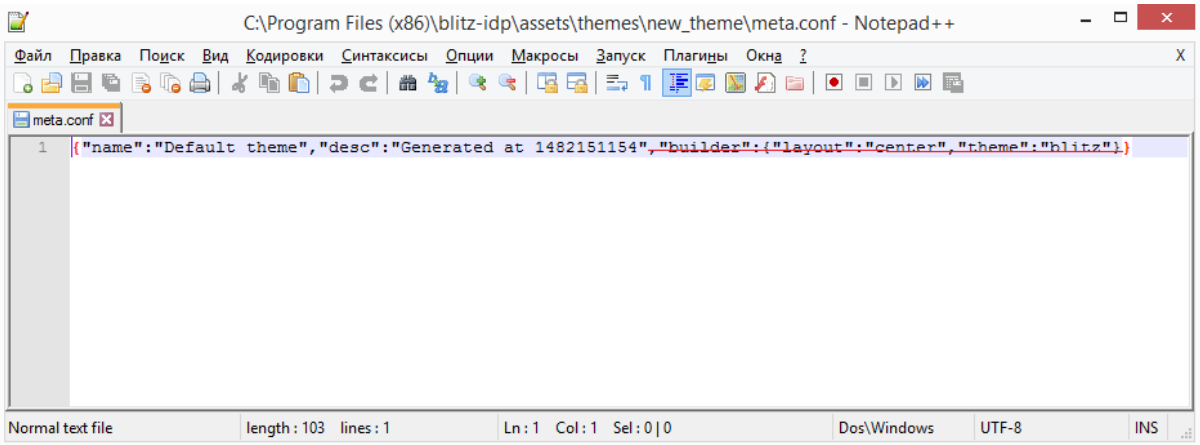
Можно настроить вид страницы входа под индивидуальные требования организации, т.е. нет необходимости ограничиваться только возможностями конструктора.

Каждый шаблон страницы входа представляет собой zip-архив. Все шаблоны размещены в директории:

```
\assets\themes
```

Самый простой способ перейти к ручному редактированию шаблона – выполнить следующие шаги:

- создать копию существующего шаблона (например, default-шаблона), нажав в консоли кнопку “Копировать”;
- перейти в соответствующую директорию с шаблонами;
- распаковать архив с только что созданным шаблоном;
- отредактировать файл `meta.conf`, содержащийся в архиве, удалив параметр `builder`;
- обратно заархивировать файлы шаблона, убедившись, что файл `meta.conf` находится в корневой директории.



После выполнения этих шагов появится возможность редактирования темы в ручном режиме. Помимо стандартных полей, описывающих саму тему, доступен блок *«Шаблон страниц»*. Он позволяет создать / изменить шаблон – текстовый файл, который компилируется с помощью шаблонизатора Twirl[63].

Шаблон должен иметь сигнатуру:

```
@(headers: Html, fBuilder: FormBuilder, scripts: Html, path: String)(implicit␣
↪request: RelyingPartyRequest[_], messages: Messages)
```

В качестве параметров при создании шаблона следует использовать:

- `headers` – HTML-код заголовка страницы, который необходимо расположить в теге `head`;
- `form` – HTML-код основной формы, который необходимо расположить в теге `body`;
- `scripts` – HTML-код с JavaScript, необходимый для корректной работы формы, который необходимо расположить в теге `body`;
- `pathAssets` – контекстный путь к ресурсам шаблона.

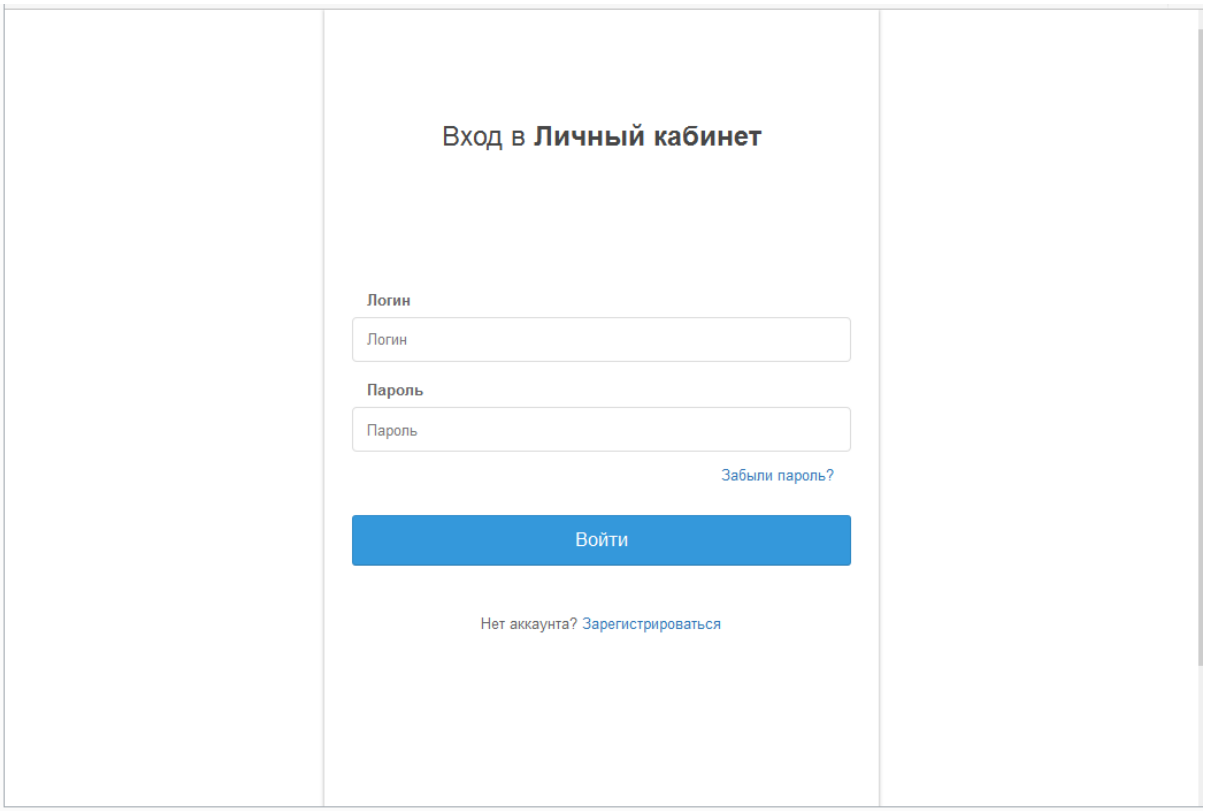
Функция `@fBuilder()` добавляет на страницу код основной формы аутентификации. Форма аутентификация (перечень и состав полей, расположение кнопок) не настраивается за исключением изменений, реализуемых средствами CSS. Иными словами, средствам CSS можно изменить цвет отдельных элементов или скрыть их – для этого следует найти соответствующий класс в CSS-файле темы и изменить его свойства.

[63] https://www.playframework.com/documentation/2.5.x/ScalaTemplates

Листинг простейшего шаблона приведен ниже:

```
@(headers: Html, fBuilder: FormBuilder, scripts: Html, path: String)(implicit␣
↪request: RelyingPartyRequest[_], messages: Messages)

<!DOCTYPE html>
<html>

<head>
    @headers
</head>

<body>
    <div id="main">
        <section id="content_wrapper">
                @fBuilder()
        </section>
        <div>
            <div>
                @Html(messages("author.copyright"))
            </div>
        </div>
    </div>
    @scripts
</body>

</html>
```

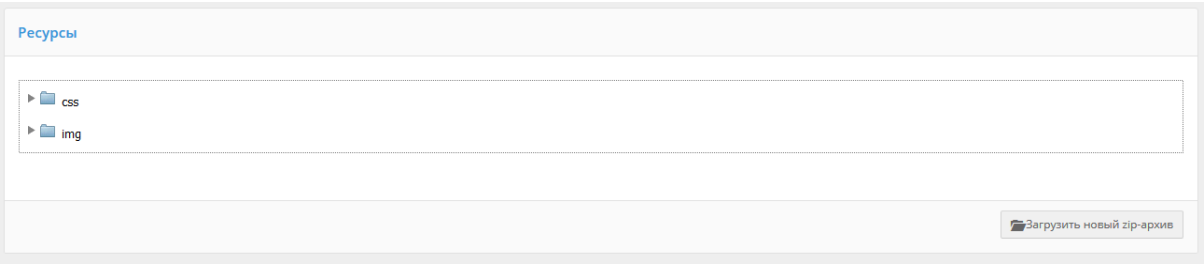
При использовании такого шаблона страница входа будет иметь вид, приведенный на рисунке ниже.



При формировании шаблона страницы входа имеется возможность использовать ресурсы – например, таблицы стилей или рисунки.

Для их загрузки следует использовать блок *«Ресурсы»* внешнего вида страницы, который позволяет загру-

зить необходимые файлы в zip-архиве. Чтобы соответствующие файлы были доступны, их следует размещать в директории архива с названием `assets`. Необходимые ресурсы также можно вручную включить в состав исходного zip-архива с шаблоном страницы.



Для добавления переключателя языка в body шаблона необходимо добавить следующий блок:

```
<div ...>
  <section class="language-section">
    <div class="language-selector">
      <select id="lang-selector"></select>
    </div>
  </section>
  @langSelector()
</div>
```

2.5.2 Личный кабинет

Blitz Identity Provider позволяет менять логотипы заголовка и футера в Личном кабинете, а также кастомизировать цветовую схему Личного кабинета с помощью CSS.

Логотип в заголовке

Для замены логотипа в заголовке Личного кабинета можно использовать один из следующих способов:

Способ 1

Замените файл логотипа `logo-ib_h30.png` в директории `.../assets/public/lib/blitz-common/` новым файлом с тем же названием.

Способ 2

1. Положите в директорию `.../assets/public/lib/blitz-common/` файл с кастомным логотипом (`mylogo.png` в примере ниже).
2. Откройте файл `.../assets/public/lib/blitz-profile/stylesheets/custom.min.css`.

   ```
   sudo vim /usr/share/identityblitz/blitz-config/assets/public/lib/blitz-profile/
   ↪stylesheets/custom.min.css
   ```

3. Укажите URL-путь к новому файлу.

   Внимание: В качестве пути к директории `.../assets/public/lib/blitz-common/` используйте `https://<domain>/blitz/assets/img/`.

```
:root{
  --navbar-branding-img: url(https://<domain>/blitz/login/assets/img/mylogo.
↪png);
  ...
}
```

Совет: Для логотипа также можно использовать файл на внешнем URL.

Логотип в футере

Для замены логотипа в футере Личного кабинета замените файл логотипа `logo-bip.png` в директории `.../assets/public/lib/blitz-common/` новым файлом с тем же названием.

Кастомизация цветовой схемы

Изменение цветовой схемы выполняется в файле:

`.../assets/public/lib/blitz-profile/stylesheets/custom.min.css`.

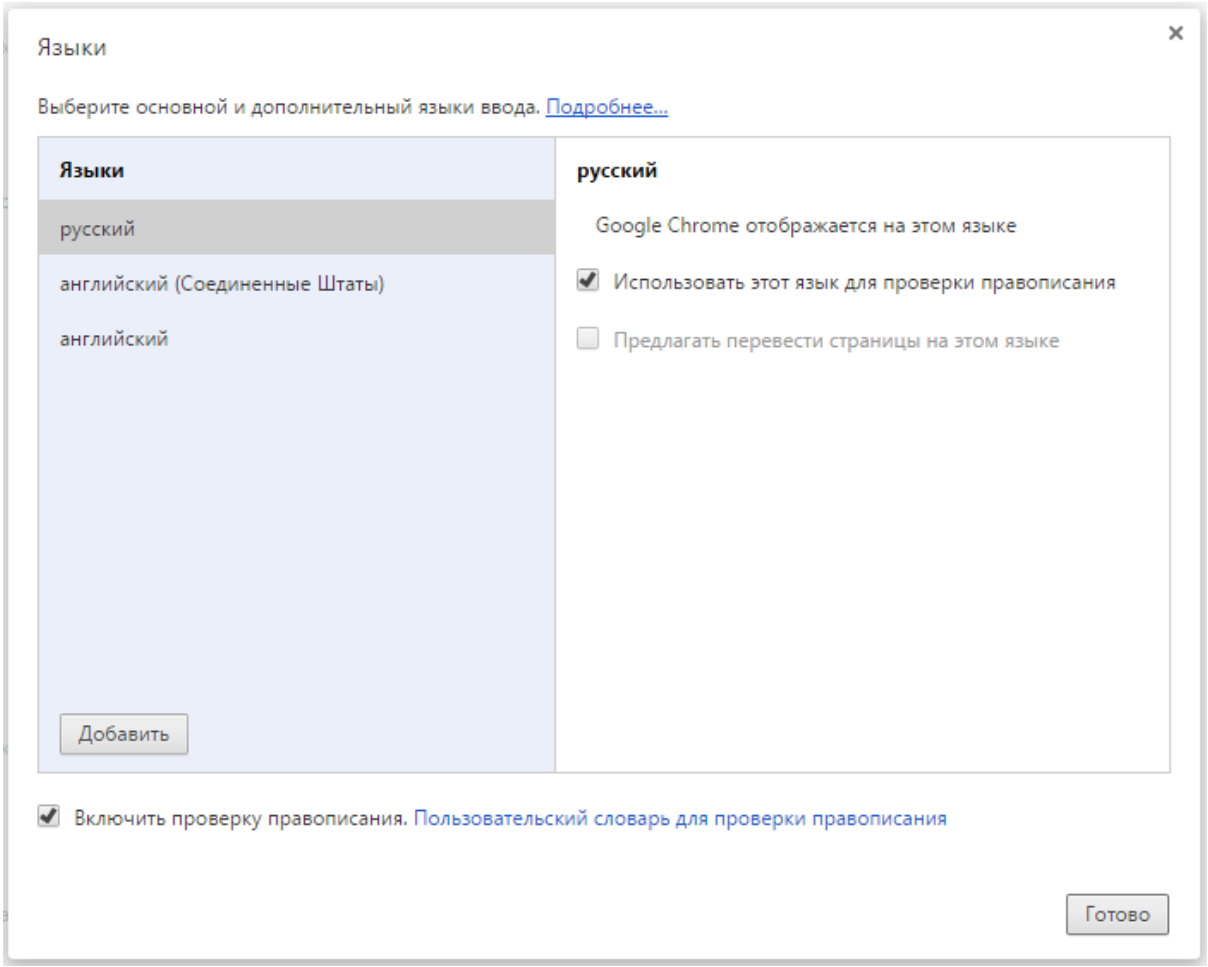
```
:root{
  --profile-color-accent:#00bde5;
  --profile-color-border-primary:#ddd;
  --profile-color-border:#ddd;
  --profile-color-button:#f1f1f1;
  --profile-color-href-hover:#1d6fa5;
  --profile-color-href:#3498db;
  --profile-color-outline:#ddd;
  --profile-color-primary:#3498db;
  --profile-color-text-button:#666;
  --profile-color-text-light:#fff;
  --profile-color-text-primary:#3498db;
}
```

2.5.3 Мультиязычность

Веб-интерфейс Blitz Identity Provider поддерживает мультиязычность. По умолчанию предусмотрено два языка – русский и английский.

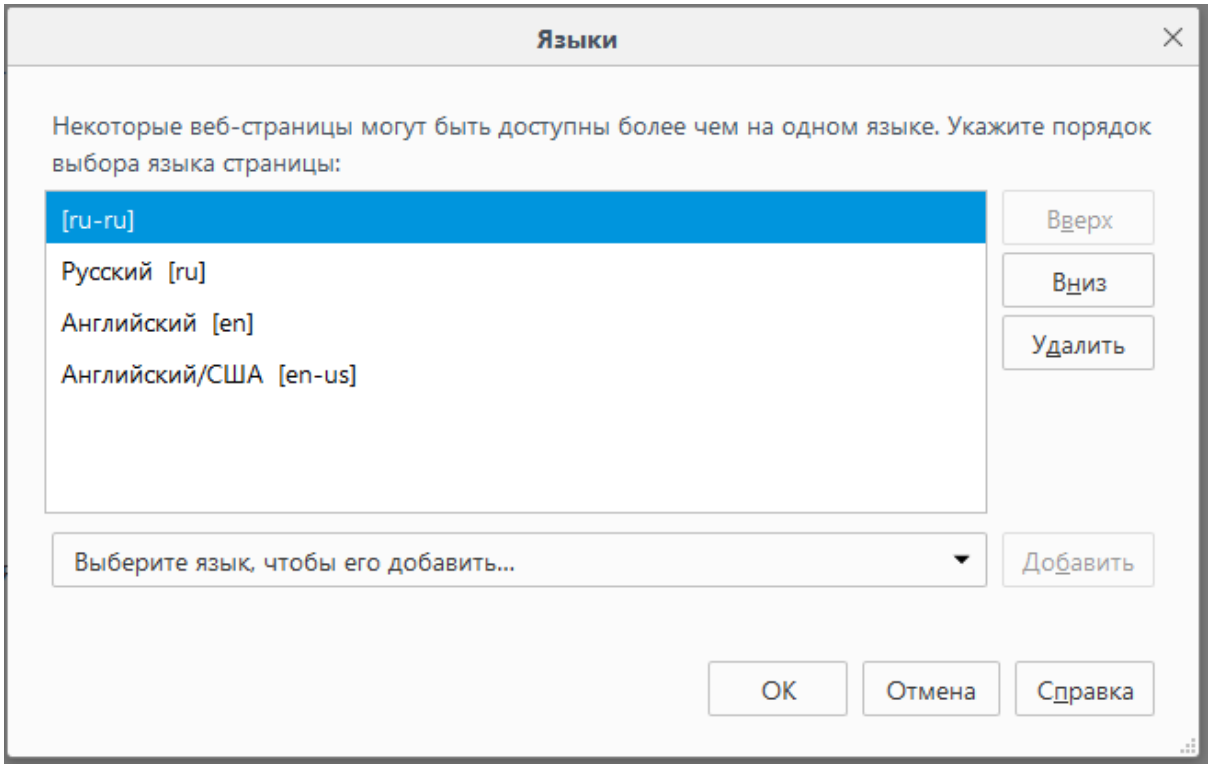
По умолчанию пользователю отображается интерфейс на том языке, который соответствует его системному языку в ОС и предпочтительному языку в браузере. В этом случае переключение языка осуществляется посредством изменения основного языка ввода (языка отображения веб-страниц) в используемом браузере. Например, для изменения языка в браузере Chrome нужно выполнить шаги:

- перейти к настройкам браузера (`chrome://settings/`);
- выбрать пункт Показать дополнительные настройки;
- нажать на кнопку Изменить языковые настройки;
- переместить нужный язык на первое место в списке.



Для изменения языка в браузере Firefox нужно выполнить шаги:

- перейти к настройкам браузера (`about:preferences`);
- выбрать раздел Содержимое настроек;
- в подразделе Языки нажать на кнопку Выбрать;
- переместить нужный язык на первое место в списке.



Дополнительно возможно провести настройку языка через конфигурационный файл `blitz.conf`. Для этого следует отредактировать раздел для настройки языка `blitz.prod.local.idp.lang` со следующими параметрами:

- `languages` – список доступных языков. Первый язык в списке считается языком по умолчанию;
- `portal-lang-cookie` – имя (`name`) и домен установки (`domain`) cookie с текущим языком портала (опциональный). Если портальная cookie задана, то смена языка в Blitz Identity Provider сохраняется в указанной cookie;
- `ignore-browser` – выключен или нет режим игнорирования языка браузера;
- `lang-variants` – *перечень идентификаторов специальных наборов строк для отдельных приложений* (страница 308).

Пример фрагмента конфигурационного файла:

```
"lang" : {
  "ignore-browser" : true,
  "languages" : [
    "ru",
    "en"
  ],
  "lang-variants": ["special1", "special2"],
  "portal-lang-cookie" : {
    "domain" : "domain.com",
    "name" : "blitzlng"
  }
}
```

Таким образом, например, если применение английского языка интерфейса не требуется, то его можно удалить из параметра `languages`.

2.5.4 Настройки текстов интерфейса

Текстовые сообщения веб-интерфейса

Blitz Identity Provider позволяет менять текстовые строки, используемые в интерфейсе системы. Для этого необходимо отредактировать файл `messages`, размещенный в директории `/custom_messages/`, добавив строку вида «параметр=значение», где параметр – идентификатор текстовой строки, а значение – необходимый текст.

Все текстовые строки, используемые Blitz Identity Provider по умолчанию, сохранены в архиве `messages.zip`, входящий в состав ПО.

Например, следующая строка отвечает за текст на форме регистрации, где размещена ссылка на условия использования:

```
reg.page.reg.action.agreement=Нажимая на кнопку &laquo;Зарегистрироваться&
↪raquo; вы соглашаетесь с <a href="{0}" target="_blank">условиями␣
↪использования</a>
```

Для корректного отображения файл должен быть сохранен в кодировке UTF-8.

При необходимости изменить английский язык следует добавить в указанную директорию файл `messages.en` и изменить в нем необходимые файлы.

При необходимости использовать в текстах символ `@` его следует ввести дважды.

Шаблоны писем и SMS-сообщений

Шаблоны писем представляют собой текстовые строки, сохраняемые аналогично обычным строкам в веб-интерфейсе. Их изменение производится аналогичным образом.

Используется унифицированный формат кодов сообщений, который имеет вид:

```
message.$[группа_сообщений].$[тип_сообщения].$[вариация].$[канал].$[часть]
```

Используются группы сообщения:

- `notif` - для информационных сообщений;
- `reg` - для взаимодействия с пользователем при регистрации;
- `recovery` - для взаимодействия с пользователем при восстановлении доступа;
- `auth` - для взаимодействия с пользователем при аутентификации;
- `profile` - для взаимодействия с пользователем в Личном кабинете;
- `api` - для взаимодействия с пользователем при использовании API.

Типы сообщений из различных групп:

notif

login_unknown_device

Информирование пользователя о входе с неизвестного устройства.

Параметры:

- `device` - код устройства;
- `device.msg` - название устройства, вычисленное через строку `msg(audit.device.$[device])`;

- `browser` - браузер пользователя;
- атрибуты из сессии пользователя;
- `ua.name` - имя устройства;
- `app.id` - идентификатор приложения;
- `app.name` - имя приложения;
- `ip` - IP-адрес;
- `ip.country` - страна;
- `ip.state` - регион;
- `ip.city` - город;
- `ip.lat` - широта;
- `ip.lng` - долгота;
- `ip.radius` - радиус окрестности;
- `device.type` - тип устройства;
- `device.mkey` - собранный ключ для сообщений, правило формирования: `s"$deviceType.$osName.$osVer"`;
- `os.name` - имя операционной системы;
- `os.ver` - версия операционной системы;
- `os.mkey` - собранный ключ для сообщений, правило формирования: `s"$osName.$osVer"`;
- `event.time` - время события (в `unixtime`).

В шаблоне сообщения можно использовать следующие функции форматирования:

- `$[<ATTR>&dic(<MSG_KEY_PREFIX>,<PARAM_SUBSTITUTION>)]` - получение значения из строки;
- `$[<ATTR>&formatUnixTime(dd MMMM YYYY r.,ru,GMT)]` - форматирование даты и времени, где `dd MMMM YYYY` г. - шаблон в формате `SimpleDateFormat`, `ru` - локаль (опционально), `GMT` - таймзона (опционально).

В шаблоне можно задавать условия на наличие параметров. Следующий пример позволяет отобразить слово `Город` и значение из параметра `ip.city` при его наличии, если же `ip.city` отсутствует, то не будет показано ничего:

```
$[ip.city+Город: ]$[ip.city-]
```

Совет: Для работы примера создайте и активируйте *процедуру входа, получающую геоданные пользователя* (страница 279).

link_social_network

Информирование пользователя о присоединении к социальной сети.

Параметры:

- `fp.humanReadableName` - название внешнего поставщика идентификации;
- атрибуты пользователя.

change_pwd

Информирование пользователя о смене пароля.

Параметры:

- атрибуты пользователя.

changed_pwd_to_object

Информирование о смене пароля в зависимой учетной записи.

Параметры:

- атрибуты зависимой учетной записи с префиксом `obj`.

access_recovery

Информирование пользователя о восстановлении пароля

Параметры:

- атрибуты пользователя.

access_recovery_by_object

Информирование о восстановлении пароля в зависимой учетной записи.

Параметры:

- атрибуты зависимой учетной записи с префиксом `obj`.

set_2factor_auth

Информирование пользователя о назначение второго фактора аутентификации.

Параметры:

- `method` - код метода аутентификации;
- `method.msg` - имя метода аутентификации, полученное через строку `msg(message.method.name.$[method]`;
- атрибуты пользователя.

granted_access_to

Информирование субъекта о предоставление доступа к объекту.

Параметры:

- `blitz_right` - код права доступа;
- атрибуты субъекта;
- атрибуты объекта с префиксом `obj`.

granted_access_on

Информирование объекта о предоставление доступа к нему.

Параметры:

- `blitz_right` - код права доступа;
- атрибуты субъекта;
- атрибуты объекта с префиксом `obj`.

revoked_access_to

Информирование субъекта об отзыве доступа к объекту.

Параметры:

- `blitz_right` - код права доступа;
- атрибуты субъекта;
- атрибуты объекта с префиксом `obj`.

revoked_access_on

Информирование объекта об отзыве доступа к нему.

Параметры:

- `blitz_right` - код права доступа;
- атрибуты субъекта;
- атрибуты объекта с префиксом `obj`.

on_registration

Информирование пользователя о регистрации его учетной записи.

Параметры:

- `_entryPoint_` - канал регистрации;
- `_appId_` - приложение;
- `_requesterId_` - приложение;
- атрибуты пользователя.

Пример строки:

```
message.notif.login_unknown_device.email.body=Уважаемый пользователь!<br><br>Мы␣
↪обнаружили, что вы вошли в систему с нового устройства $[event.time&
↪formatUnixTime(dd MMMM YYYY г.,ru,GMT)]:<br>$[device.mkey&dic(dics.devices,os.
↪ver)], браузер $[ua.name&dic(dics.browsers)]<br>Если вы не совершали это␣
↪действие, обратитесь к администратору.
```

reg

vrf_code

Отправка кода подтверждения контакта при регистрации.

Параметры:

- `code` - код подтверждения;
- `link` - ссылка для подтверждения (только для `email`);
- `req.ip` - IP-адрес;
- `req.userAgent` - `userAgent` пользователя;
- `cfg.domain` - домен;
- атрибуты пользователя из контекста регистрации с префиксом `attrs`.

set_pwd_link

Отправка ссылки на смену пароля при регистрации (только для канала `email`).

Параметры:

- `link` - ссылка на страницу смены пароля;
- `req.ip` - IP- адрес;
- `req.userAgent` - `userAgent` пользователя;
- `cfg.domain` - домен;
- атрибуты пользователя из контекста регистрации с префиксом `attrs`.

generated_pwd

Отправка назначенного при регистрации пароля (только для канала SMS).

Параметры:

- `pwd` - сгенерированный пароль;
- `req.ip` - IP-адрес;
- `req.userAgent` - `userAgent` пользователя;
- `cfg.domain` - домен атрибуты пользователя из контекста регистрации с префиксом `attrs`.

recovery

vrf_code

Отправка кода подтверждения контакта при восстановлении доступа.

Параметры:

- `code` - код подтверждения;
- `link` - ссылка для подтверждения (только для `email`).

auth

vrf_code

Отправка кода подтверждения мобильного номера (каналы: `SMS`/`push`).

Параметры:

- `code` - код подтверждения.

profile

vrf_code

Отправка кода подтверждения контакта при изменении его в Личном кабинете.

Параметры:

- `attr.msg` - наименование атрибута в форме профиля;
- `attr` - код атрибута;
- `link` - ссылка для подтверждения (только для `email`);
- `code` - код подтверждения.

api

vrf_code

Вариации:

- `$attr.$rpId` - отдельно для данного приложения и атрибута;
- `$attr` - отдельно для данного атрибута.

Отправка кода подтверждения контакта через API

Параметры:

- `code` - код подтверждения;
- `link` - ссылка для подтверждения (только для `email`);
- `attr.value` - новый контакт (e-mail или мобильный телефон);
- `attr` - код атрибута контакта.

Вариации позволяют помимо базового шаблона сообщения задать его варианты (например, отдельный шаблон в разрезе приложений). Наличие вариации проверяется по основному шаблону с текстом сообщения (часть `body`). Если вариация основного шаблона описана в системе, то все остальные шаблоны (`email.subject`, `email.from`, `push.title`) будут применяться с этой же вариацией. Если вариаций несколько, то они будут проверяться в некотором заданном порядке (обычно от большей детализации к меньшей). При отсутствии вариаций будет использован базовый шаблон. В большинстве случаев вариации отсутствуют.

Возможны следующие каналы:

- `sms` - отправка сообщений по SMS. Части для этого канала отсутствуют;
- `email` - отправка сообщений по электронной почте. Части для этого канала:
  - `subject` - тема;
  - `body` - основное содержание;
  - `from` - отправитель (необязательно);

- `push` - отправка push-уведомлений. Части для этого канала:
  - `title` - тема;
  - `body` - основное содержание.

Пример ключей для сообщений типа `login_unknown_device`:

- `message.notif.login_unknown_device.email.subject` - тема сообщения по email;
- `message.notif.login_unknown_device.email.body` - текст сообщения по email;
- `message.notif.login_unknown_device.email.from` - отправитель email сообщения;
- `message.notif.login_unknown_device.sms` – текст сообщения по SMS.

Имена устройств и браузеров

В Blitz Identity Provider можно настроить имена устройств (операционных систем) и браузеров с точностью до версий. Для этого в нужно создать в директории `custom_messages` в файле `messages` строки, имена которых соответствуют следующим паттернам:

- для браузеров – `dics.browsers.<name>`. Поддерживаются определение следующих браузеров для подстановки в `<name>`: `Firefox`, `Opera`, `Chrome`, `Safari`, `IE`, `Edge`, `Yandex`, `Sputnik`, `unknown`. В текст строки в качестве строки подстановки `{0}` передается версия браузера.
- для устройств (операционных систем) – `dics.devices.<typ>.<os>.<ver>`. В качестве `<typ>` можно указывать: `kindle`, `mobile`, `tablet`, `iphone`, `windowsPhone`, `pc`, `ipad`, `playStation`, `unknown`. В качестве `<os>` можно указывать: `Android`, `iOS`, `WindowsPhone`, `Windows`, `macOS`, `Linux`, `ChromeOS`, `unknown`. Если для `<os>` и `<ver>` не определена частная строка, то берется более общая строка. В текст строки в качестве строки подстановки `{0}` передается версия операционной системы.

Примеры строк:

```
dics.browsers.Firefox=Firefox Browser {0}
dics.browsers.Opera=Opera {0}
dics.browsers.Chrome=Google Chrome {0}
dics.browsers.Safari=Safari {0}
dics.browsers.IE=Internet Explorer
dics.browsers.Edge=Microsoft Edge {0}
dics.browsers.Yandex=Яндекс.Браузер {0}
dics.browsers.Sputnik=Спутник
dics.devices.mobile=Мобильное устройство
dics.devices.mobile.Android=Android
dics.devices.mobile.Android.10=Android 10
dics.devices.mobile.Android.9=Android 9
dics.devices.tablet=Планшет
dics.devices.iphone=iPhone
dics.devices.iphone.iOS.14=iPhone (iOS {0})
dics.devices.pc.macOS=macOS {0}
dics.devices.pc.macOS.13=macOS Ventura {0}
dics.devices.pc.macOS.12=macOS Monterey {0}
dics.devices.pc.macOS.11=macOS Big Sur {0}
dics.devices.pc.macOS.10.15=macOS Catalina {0}
dics.devices.pc.macOS.10.14=macOS Mojave {0}
dics.devices.pc.macOS.10.13=macOS High Sierra {0}
dics.devices.pc.macOS.10.12=macOS Sierra {0}
dics.devices.pc.Windows.8=Windows 8
dics.devices.pc.Windows.10=Windows 10
dics.devices.pc.Windows.11=Windows 11
```

Сообщения для разных приложений

Возможно изменение всех текстовых сообщений и шаблонов таким образом, чтобы использовались специфические тексты/шаблоны для разных приложений. Таким образом можно, например, брендировать письма, отправляемые при регистрации на разных сайтах, подключенных к одной установке Blitz Identity Provider, или давать ссылку на скачивание различных правил использования ресурса.

Для привязки набора шаблонов к конкретному приложению следует выполнить шаги:

1. Создать экземпляр файла с текстами, который будет использоваться исключительно для данного приложения. Для этого в директории `custom_messages/` создать текстовый файл `messages.ru-special1` (`messages.en-special1`) для данного приложения, где `special1` – последовательность из 5-8 символов (допускаются как цифры, так и буквы латинского алфавита).
2. Отредактировать файл `messages.ru-special1` (`messages.en-special1`), *добавив* (страница 301) в него специфические строки для данного приложения. Все остальные строки будут взяты из базы строк по умолчанию.
3. Отредактировать файл `blitz.conf` следующим образом:
   - в разделе `blitz.prod.local.idp.apps` файла найти идентификатор приложения, который должен использовать созданный файл шаблона;
   - добавить в настройки приложения параметр вида `"lang-variant" : "special1"`, где `special1` – использованная для маркировки шаблона последовательность символов.

   Пример:

```
"demo-application" : {
    "domain" : "http://testdomain.ru",
    "lang-variant" : "special1",
    "name" : "test",
    "oauth" : {
        "autoConsent" : false,
        "clientSecret" : "1234567890",
        "defaultScopes" : [],
        "enabled" : true,
        "redirectUriPrefixes" : [
            "http://localhost"
        ]
    },
    "theme" : "default"
}
```

4. Зарегистрировать в разделе `blitz.prod.local.idp.lang` в настройке `lang-variant` все используемые для маркировки различных приложений последовательности символов (`special1`, `special2`).

После этого при входе в приложение будет использоваться специально созданный файл сообщений.

Сообщения вспомогательных приложений (pipes)

В Blitz Identity Provider можно настроить сообщения вспомогательного приложения, выпускающего ключ безопасности (Passkey, WebAuthn, FIDO2) при входе пользователя. Можно настроить разные тексты сообщений в зависимости от устройств (операционных систем), используемых пользователем. Для этого в нужно создать в директории `custom_messages` в файле `messages` строки, имена которых соответствуют следующим паттернам:

- `pipes.conf.webAuthn.addKey.<message-path>.<device-type>.<os>`;
- `login.outside.flow.error.internal.webAuthn.addKey.<device-type>.<os>`.

В качестве `<message-path>` указывается имя строки (см. ниже пример). В качестве `<device-type>` указывается тип устройства: `mobile`, `tablet`, `iphone`, `pc`, `ipad`. В качестве `<os>` можно указывать:

Android, iOS, Windows, macOS, Linux, ChromeOS. Если для <device-type> и <os> не определена частная строка, то берется более общая строка.

Примеры строк:

```
pipes.conf.webAuthn.addKey.page.title.pc.macOS=Вход по Touch ID
pipes.conf.webAuthn.addKey.head.title.pc.macOS=Вход по Touch ID
pipes.conf.webAuthn.addKey.info.pc.macOS=Использовать Touch ID или пароль␣
↪компьютера Mac для входа в приложения?
pipes.conf.webAuthn.addKey.finishInfo.pc.macOS=Вход по Touch ID для учетной записи␣
↪настроен. Нажмите кнопку "Продолжить"
pipes.conf.webAuthn.addKey.name.pc.macOS=Touch ID на Mac
login.outside.flow.error.internal.webAuthn.addKey.pc.macOS=Произошла ошибка при␣
↪настройке входа по Touch ID на Mac

pipes.conf.webAuthn.addKey.page.title.pc.Windows=Вход через Windows Hello
pipes.conf.webAuthn.addKey.head.title.pc.Windows=Вход через Windows Hello
pipes.conf.webAuthn.addKey.info.pc.Windows=Использовать PIN-код компьютера,␣
↪распознавание лица или отпечатка пальца для входа в приложения?
pipes.conf.webAuthn.addKey.finishInfo.pc.Windows=Вход через Windows Hello для␣
↪учетной записи настроен. Нажмите кнопку "Продолжить"
pipes.conf.webAuthn.addKey.name.pc.Windows=Windows Hello
login.outside.flow.error.internal.webAuthn.addKey.pc.Windows=Произошла ошибка при␣
↪настройке входа через Windows Hello

pipes.conf.webAuthn.addKey.page.title.iphone.iOS=Вход по Face ID
pipes.conf.webAuthn.addKey.head.title.iphone.iOS=Вход по Face ID
pipes.conf.webAuthn.addKey.info.iphone.iOS=Использовать Face ID или Touch ID␣
↪телефона для входа в приложения?
pipes.conf.webAuthn.addKey.finishInfo.iphone.iOS=Вход через Face ID для учетной␣
↪записи настроен. Нажмите кнопку "Продолжить"
pipes.conf.webAuthn.addKey.name.iphone.iOS=Face ID на iPhone
login.outside.flow.error.internal.webAuthn.addKey.iphone.iOS=Произошла ошибка при␣
↪настройке входа через Face ID

pipes.conf.webAuthn.addKey.page.title.ipad.iOS=Вход по Touch ID
pipes.conf.webAuthn.addKey.head.title.ipad.iOS=Вход по Touch ID
pipes.conf.webAuthn.addKey.info.ipad.iOS=Использовать Touch ID планшета для входа␣
↪в приложения?
pipes.conf.webAuthn.addKey.finishInfo.ipad.iOS=Вход через Touch ID для учетной␣
↪записи настроен. Нажмите кнопку "Продолжить"
pipes.conf.webAuthn.addKey.name.ipad.iOS=Touch ID на iPad
login.outside.flow.error.internal.webAuthn.addKey.ipad.iOS=Произошла ошибка при␣
↪настройке входа через Touch ID

pipes.conf.webAuthn.addKey.page.title.mobile.Android=Вход по распознаванию лица␣
↪или отпечатка пальца
pipes.conf.webAuthn.addKey.head.title.mobile.Android=Вход по распознаванию лица␣
↪или отпечатка пальца
pipes.conf.webAuthn.addKey.info.mobile.Android=Использовать распознавание лица или␣
↪отпечатка пальца для входа в приложения?
pipes.conf.webAuthn.addKey.finishInfo.mobile.Android=Вход через распознавание лица␣
↪или отпечатка пальца для учетной записи настроен. Нажмите кнопку "Продолжить"
pipes.conf.webAuthn.addKey.name.mobile.Android=Smart Lock на Android
login.outside.flow.error.internal.webAuthn.addKey.mobile.Android=Произошла ошибка␣
↪при настройке входа через распознавание лица или отпечатка пальца

pipes.conf.webAuthn.addKey.page.title=Вход по ключу безопасности
pipes.conf.webAuthn.addKey.head.title=Вход по ключу безопасности
pipes.conf.webAuthn.addKey.info=Использовать ключ безопасности FIDO2 для входа в␣
↪приложения?
pipes.conf.webAuthn.addKey.finishInfo=Вход через ключ безопасности для учетной␣
```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪записи настроен. Нажмите кнопку "Продолжить"
pipes.conf.webAuthn.addKey.name=FIDO2
```

В Blitz Identity Provider можно настроить тексты для вспомогательного приложения, показывающего пользователю сообщение при входе в приложение. Для этого нужно определить в директории `custom_messages` в файле `messages` строки для настроенных в `blitz.prod.local.idp.built-in-pipes.info` приложений с их `{id}` вспомогательного приложения.

Пример строк:

- `pipes.info.head.title.{id}`: название вкладки
- `pipes.info.page.title.{id}`: заголовок вспомогательного приложения
- `pipes.info.message.{id}`: текст сообщения
- `pipes.info.read.{id}`: название кнопки (для вспомогательных приложений с типом «news»)
- `pipes.info.agree.{id}`: название первой кнопки (для вспомогательных приложений с типом «agreement»)
- `pipes.info.disagree.{id}`: название второй кнопки (для вспомогательных приложений с типом «agreement»)

В Blitz Identity Provider можно настроить тексты для вспомогательного приложения, запрашивающего у пользователя при входе выбор значения из списка и сохраняющего результат выбора в атрибут учетной записи. Для этого нужно определить в директории `custom_messages` в файле `messages` строки для настроенных в `blitz.prod.local.idp.built-in-pipes.choice` приложений с их `{id}` вспомогательного приложения.

Пример строк:

- `pipes.choice.head.title.{id}`: название вкладки
- `pipes.choice.page.title.{id}`: заголовок вспомогательного приложения
- `pipes.choice.info.{id}`: текст информации под заголовком
- `pipes.choice.button.{id}.{choiceId}`: текст на кнопке выбора
- `pipes.choice.skip`: текст на кнопке пропуска

В Blitz Identity Provider можно настроить тексты для вспомогательного приложения, запрашивающего у пользователя при входе в приложение ввод значения атрибута. Для этого нужно определить в директории `custom_messages` в файле `messages` строки, соответствующие следующему паттерну – `pipes.act.attr. <message-path>.common.<attr-name>`. В качестве `<message-path>` указывается имя строки (см. ниже пример). В качестве `<attr-name>` указывается имя атрибута.

Примеры строк (в случае заполнения атрибута `family_name`):

```
pipes.act.attr.page.title.common.family_name=Подтверждение фамилии
pipes.act.attr.head.title.common.family_name=Подтверждение фамилии
pipes.act.attr.info.confirm.common.family_name=В учетной записи указана ваша␣
↪фамилия?<br>Проверьте и нажмите кнопку <b>Подтвердить</b>.
pipes.act.attr.info.enter.common.family_name=В учетной записи не указана фамилия.
↪<br>Введите и нажмите кнопку <b>Подтвердить</b>.
pipes.act.attr.label.common.family_name=Фамилия
pipes.act.attr.msg.required.msg.common.surname=Введите фамилию
```

2.5.5 Логотипы кнопок входа через сервисы внешних поставщиков

В Blitz Identity Provider можно изменить логотипы, отображаемые на кнопках входа через внешние поставщики идентификации (социальные сети) на странице входа и кнопок привязок внешних поставщиков идентификации в личном кабинете.

Для настройки нужно создать в директории `custom_messages` в файле `messages` строки, имена которых соответствуют следующим паттернам:

- для страницы входа – `meth-logo.${type}.${name}`
- для личного кабинета – `social-icon.${type}.${name}`

В `${type}` указывается тип внешнего поставщика идентификации, в `${name}` – имя поставщика идентификации. Значения берутся из *настроек* (страница 143).

В значении строк указываются имена `<icon class>`, присваиваемые кнопкам.

Примеры строк:

```
social-icon.saml.demo-idp=saml-demo
social-icon.esia.esia_1=esia
meth-logo.saml.demo-idp=meth-saml-demo
meth-logo.esia=meth-esia
```

2.6 Настройки конфигурационных файлов

2.6.1 Полный список файлов

Настройки Blitz Identity Provider, кроме приложения `blitz-keeper`, расположены в каталоге `/usr/share/identityblitz/blitz-config`.

Список вложенных директорий и файлов:

- `apps/` – *настройки подключенных приложений* (страница 362);
- `assets/` – настройки пользовательского интерфейса. См.:
  - *Использование и обновление плагина* (страница 105),
  - *Внешние поставщики идентификации* (страница 143),
  - *Страница входа* (страница 290);
- `custom_messages/` – *строки пользовательского интерфейса* (страница 301);
- `devices/` – *вспомогательные каталоги для обработки загрузки HOTP и TOTP устройств* (страница 126);
- `/dynamic/idstore/` – *пользовательские процедуры для кастомизации логики операций с хранилищами данных* (страница 285);
- `flows/` – *процедуры входа* (страница 260);
- `saml/` – *настройки SAML* (страница 234);
- `simple/` – *настройки подключения приложений по протоколу Simple* (страница 224);
- `token_exchange/rules/` – *настройки правил обмена маркеров доступа* (страница 546);
- `blitz.conf` – *основной файл конфигурации* (страница 312);
- `boot.conf` – настройки путей к конфигурационным файлам;
- `console.conf` – *настройки консоли управления* (страница 363);
- `credentials` – *учетные записи администраторов консоли управления* (страница 366);

- `play.conf` – настройки серверов приложений. См.:
  - установка консоли управления `blitz-console` в *Общая инструкция по установке* (страница 18),
  - *Домен Blitz Identity Provider* (страница 331);
- `logback.xml` – настройки журналирования событий и ошибок.

Большинство настроек задается с использованием консоли управления. Для ряда настроек необходимо самостоятельное редактирование конфигурационных файлов. Такие настройки описаны далее в подразделах.

Конфигурационный файл приложения blitz-keeper расположен в `/etc/blitz-keeper`. Используются следующие конфигурационные файлы:

- `blitz-keeper.conf` – *настройки шлюза безопасности* (страница 546);
- `blitz-keeper-log4j.xml` – настройки журналирования событий и ошибок.

2.6.2 Настройки в файле blitz.conf

Основной конфигурационный файл `blitz.conf` состоит из следующих блоков настроек, имеющих следующее назначение:

- `blitz.prod.local.idp.apps` – настройки подключенных приложений;
- `blitz.prod.local.idp.apps-source` – расположение настроек подключенных приложений;
- `blitz.prod.local.idp.audit` – настройки регистрации событий безопасности;
- `blitz.prod.local.idp.captcha` – настройки взаимодействия с сервисом CAPTCHA;
- `blitz.prod.local.idp.events` – настройки отправки событий в очередь;
- `blitz.prod.local.idp.federation` – настройки внешних поставщиков идентификации;
- `blitz.prod.local.idp.flexible-flows` – настройки процедур входа;
- `blitz.prod.local.idp.id-attrs` – настройки атрибутов;
- `blitz.prod.local.idp.id-stores` – настройки хранения атрибутов в хранилище учетных записей;
- `blitz.prod.local.idp.internal-store` – настройки подключения к СУБД;
- `blitz.prod.local.idp.keystore` – настройки доступа к хранилищу ключей;
- `blitz.prod.local.idp.lang` – языковые настройки Blitz Identity Provider;
- `blitz.prod.local.idp.license` – лицензионный ключ Blitz Idenity Provider;
- `blitz.prod.local.idp.logger` – настройки логгеров;
- `blitz.prod.local.idp.login` – настройки методов аутентификации;
- `blitz.prod.local.idp.logout` – настройки процесса логаута;
- `blitz.prod.local.idp.messages` – настройки файлов сообщений;
- `blitz.prod.local.idp.messaging` – настройки вызова сервисов информирования;
- `blitz.prod.local.idp.net` – настройки сети;
- `blitz.prod.local.idp.notifier` – настройки уведомлений о событиях;
- `blitz.prod.local.idp.oauth` – настройки разрешений (scope);
- `blitz.prod.local.idp.password-policy` – настройки парольной политики;
- `blitz.prod.local.idp.play` – настройки сервера приложений Blitz Identity Provider;

- `blitz.prod.local.idp.provisioning` – настройки сервисов регистрации пользователей и восстановления забытого пароля;
- `blitz.prod.local.idp.realms` – настройки шифрования идентификаторов приложений (домены приватности);
- `blitz.prod.local.idp.rights` – настройки прав доступа;
- `blitz.prod.local.idp.saml` – настройки SAML;
- `blitz.prod.local.idp.stores` – настройки основной СУБД;
- `blitz.prod.local.idp.tasks` – настройки механизма обработки задач;
- `blitz.prod.local.idp.user-profile` – настройки личного кабинета;
- `blitz.prod.local.idp.webAuthn` – настройки ключей безопасности;
- `home` – путь к каталогу установки Blitz Identity Provider на сервере приложений.

Далее приведено описание настроек, недоступных из консоли управления, и проводимых посредством редактирования конфигурационного файла `blitz.conf`.

Логины и пароли

Количество проверок пароля

Можно установить ограничение на количество одновременных парольных аутентификаций с одинаковым логином пользователя за период времени. По умолчанию установлен режим, что Blitz Identity Provider разрешает пройти не более 3 аутентификаций на один и тот же логин в течение 600 мс. Чтобы скорректировать стандартные настройки, необходимо в конфигурационном файле `blitz.conf` добавить в раздел `blitz.prod.local.idp.login.methods.password` следующий блок:

```
"throughput": {
  "limit": 3,
  "window": 600
}
```

Смена пароля при входе

Если Blitz Identity Provider подключен к хранилищу учетных записей, в которое разрешена запись (хранилище не в режиме «Только для чтения»), то при входе пользователя с учетной записью из этого хранилища, если парольная политика потребует от пользователя смены пароля, то пользователю будет показан экран изменения пароля (с просьбой ввести старый и новый пароль). Иногда отображение экрана смены пароля при входе нежелательно. Отключить экран можно с помощью задания в конфигурационном файле `blitz.conf` в разделе `blitz.prod.local.idp.login.methods.password` следующего блока настроек:

```
"changePasswordMode": {
  "type": "except_for",
  "idStores":["ldap1","ldap2"]
}
```

В настройке `idStores` нужно перечислить идентификаторы тех хранилищ учетных записей, для которых пользователю не должна предлагаться смена пароля при входе.

Системные имена полей логина и пароля

По умолчанию Blitz Identity Provider на странице ввода логина и пароля называет поля ввода логина и пароля идентификаторами `login` и `password`. При внедрении Blitz Identity Provider при миграции с существующей системы входа, в которой использовались другие названия полей, может существовать требование, что нужно сохранить в Blitz Identity Provider прежние используемые названия полей. Это может быть полезно, так как некоторые браузеры, сохранившие логины и пароли пользователей, и использующие их для автоподстановки, смогут продолжать осуществлять автоподстановку сохраненных значений и при переключении системы входа на использование Blitz Identity Provider, при условии сохранения домена страницы входа и названия полей на странице входа.

Для установки требуемых названий полей ввода логина и пароля необходимо в блок настроек `blitz.prod.local.idp.password` добавить следующие настройки:

- `loginInputName` – идентификатор поля ввода логина на странице входа;
- `passwordInputName` – идентификатор поля ввода пароля на странице входа.

Пример настроек:

```
"password" : {
    …
    "loginInputName" : "j_username",
    "passwordInputName" : "j_password",
    …
}
```

Атрибуты

Внешний валидатор атрибута

Если возможностей, предоставляемых *правилами преобразования* (страница 75) входных значений с помощью регулярных выражений недостаточно для реализации требуемой бизнес-логики проверки допустимости значения атрибута, то для атрибута можно запрограммировать и настроить использование внешнего валидатора.

Для этого нужно создать программу с внешним валидатором и собрать ее в JAR-файл.

Созданный JAR-файл нужно скопировать на серверы с приложениями Blitz Identity Provider. Адрес размещения JAR-файлов прописать в Java-опцию `extensionsDir`.

Пример:

```
export JAVA_OPTS="${JAVA_OPTS} -DextensionsDir=/usr/share/identityblitz/extensions"
```

В блоке настроек атрибутов `blitz.prod.local.idp.id-attrs.attrsMeta` в блок описания атрибута, для которого нужно включить проверку через внешний валидатор, необходимо добавить в блоке `source` блок `validators`:

- в настройке `className` прописать адрес Java-класса, реализующего имплементацию интерфейса `AttributeValidator` из Blitz JDK;
- в блоке `conf` прописать настройки, передающиеся в валидатор.

Пример настроек:

```
"id-attrs" : {
    "attrsMeta" : [
        {
            {
                "class" : "verified-mobile",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                "format" : "string",
                "name" : "phone_number",
                "realmed" : false,
                "required" : false,
                "searchable" : true,
                "source" : {
                    "validators" : [
                        {
                            "className" : "validator.MobileValidator",
                            "conf" : {
                                "conf1" : "value1"
                            }
                        }
                    ],
                    "type" : "idStore"
                },
                "unique" : false
            },
            …
        }
    ]
}
```

Транслятор атрибута

С атрибутом можно ассоциировать транслятор, описывающий правила преобразования атрибута при чтении из LDAP-каталога и при записи в LDAP-каталог. В блоке настроек хранилища атрибута в разделе настроек соответствия атрибутов `blitz.prod.local.idp.id-stores.list.mappingRules` в блоке описания атрибута, для которого нужно включить транслятор, необходимо добавить блок `translator` с настройкой `className`, в которой указать имя Java-класса, реализующего алгоритм трансляции. Java-класс должен реализовывать имплементацию интерфейса LdapAttributeTranslator из Blitz JDK.

Для некоторых атрибутов из LDAP-каталога Active Directory Blitz Identity Provider предоставляет встроенные Java-классы:

- При необходимости настроить транслятор для атрибута `objectGUID` так, чтобы этот атрибут представлялся не в байтовом виде, а в форме строки GUID, используйте Java-класс `com.identityblitz.idp.store.ldap.core.translator.ObjectGUIDTranslator`.

  Пример настройки:

```
"id-stores" : {
    "list" : [
        {
            …
            "mappingRules" : [
                …
                {
                    "name" : "objectGUID",
                    "storeAttr" : "objectGUID",
                    "translator" : {
                        "className" :
                            "com.identityblitz.idp.store.ldap.core.translator.
↪ObjectGUIDTranslator"
                    }
                }
            ],
        },
        …
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    ]
}
```

- При необходимости настроить транслятор для атрибута `objectSID` для его конвертации в строковый вид, используйте Java-класс `com.identityblitz.idp.store.ldap.core.translator.ObjectSIDTranslator`. Для конвертированного атрибута возможен поиск, при этом операция `LIKE` не поддерживается. Также нельзя его изменять и задавать при создании.

  Пример настройки:

```
"id-stores" : {
    "list" : [
        {
            …
            "mappingRules" : [
                …
                {
                    "name": "objectSID",
                    "storeAttr": "objectSID",
                    "translator": {
                        "className": "com.identityblitz.idp.store.ldap.core.
↪translator.ObjectSIDTranslator"
                    }
                }
            ],
        },
        …
    ]
}
```

При использовании самостоятельно разработанного транслятора необходимо создать программу с внешним транслятором и собрать ее в JAR-файл.

Созданный JAR-файл нужно скопировать на серверы с приложениями Blitz Identity Provider. Адрес размещения JAR-файлов прописать в Java-опцию `extensionsDir`.

Пример:

```
export JAVA_OPTS="${JAVA_OPTS} -DextensionsDir=/usr/share/identityblitz/extensions"
```

Электронная подпись

Вызов внешнего сервиса проверки ЭП

Для интеграции с внешним сервисом проверки электронной подписи должна быть разработана специальная библиотека проверки подписи. Система будет производить проверку электронной подписи через эту систему после прописывания данной библиотеки в конфигурационном файле, в разделе `blitz.prod.local.idp.login.methods.x509`, следующим образом:

```
"x509-verifier" : {
    "javaClass" : "<Java-класс реализации коннектора>",
    "pathToJar" : "/usr/…/check-signature-1.0.0.jar",
    "signatureValidationServiceUrl" : "<адрес сервиса >"
}
```

Вызов плагина ЭП

Для задания нестандартных настроек вызова плагина электронной подписи при запросе входа пользователем по электронной подписи необходимо в конфигурационном файле в разделе `blitz.prod.local.idp.login.methods.x509` необходимо создать блок настроек `plugin` с переопределенными настройками вызова плагина:

```
"plugin" : {
    "allModulesEnabled" : false,
    "capi" : {
        "providers" : [
            {
                "name" : "Crypto-Pro GOST R 34.10-2001 Cryptographic Service␣
↪Provider",
                "pinMode" : 1
            },
            {
                "name" : "Crypto-Pro GOST R 34.10-2012 Strong Cryptographic␣
↪Service Provider",
                "pinMode" : 1
            },
            {
                "name" : "Infotecs Cryptographic Service Provider",
                "pinMode" : 1
            },
            {
                "name" : "Signal-COM GOST R 34.10-2012 (512) Cryptographic Provider
↪",
                "pinMode" : 1
            },
            {
                "name" : "Signal-COM CPGOST Cryptographic Provider",
                "pinMode" : 1
            },
            {
                "name" : "Signal-COM GOST R 34.10-2012 (256) Cryptographic Provider
↪",
                "pinMode" : 1
            }
        ],
        "stores" : []
    },
    "modules" : [
        "capi",
        "Aladdin R.D. Unified JaCarta",
        "ISBC ESMART",
        "Rutoken",
        "SafeNet"
    ]
}
```

В блоке конфигурации можно убрать лишние модули из `modules` и `providers`, чтобы ограничить перечень доступных средств электронной подписи. Также для используемых провайдеров можно настроить режим ввода пин-кода согласно документации на плагин[64].

Если необходимо, чтобы отображались только ключи подписи из реестра ОС Windows, доступные через MS CAPI, то блок настроек должен иметь следующий вид:

[64] https://identityblitz.ru/products/smart-card-plugin/documentation/?ref=p_pl#document-9

```
"plugin": {
    "allModulesEnabled": false,
    "capi": {
        "stores": [
            {
                "name": "My"
            }
        ]
    },
    "modules": []
}
```

CAPTCHA

Для отображения сервиса CAPTCHA при входе по логину и паролю необходимо внести изменения в конфигурационный файл, а также загрузить необходимые файлы (CSS и JS).

Изменения конфигурационного файла должны быть произведены:

- в блоке настроек `blitz.prod.local.idp.captcha`. Пример записи настройки приведен ниже:

```
"captcha" : {
    "exampleCaptcha": {
        "operations": [
            {
                "call": {
                    "headers": [
                        "accept:application/json",
                        "Authorization:Bearer ${cfg.bearerToken}"
                    ],
                    "method": "post",
                    "url": "https://captcha.example.com/captcha/1.0.0/check?
↪uniqueFormHash=${ste.uniqueFormHash}&code=${ocp.code}&options[system]=${cfg.
↪system}&options[token]=${cfg.token}"
                },
                "check": {
                    "errRegExp": {},
                    "okRegExp": {
                        "error": "0"
                    }
                },
                "name": "check",
                "newState": {
                    "uniqueFormHash": "${rsp.result.uniqueFormHash-}"
                }
            },
            {
                "call": {
                    "headers": [
                        "accept:application/json",
                        "Authorization:Bearer ${cfg.bearerToken}"
                    ],
                    "method": "get",
                    "url": "https://captcha.example.com/captcha/1.0.0/create?type=$
↪{cfg.type}&options[system]=${cfg.system}&options[token]=${cfg.token}"
                },
                "name": "create",
                "newState": {
                    "uniqueFormHash": "${rsp.result.uniqueFormHash-}"
                }
```

(continues on next page)

(продолжение с предыдущей страницы)

```
            },
            {
                "call": {
                    "headers": [
                        "accept:application/json",
                        "Authorization:Bearer ${cfg.bearerToken}"
                    ],
                    "method": "post",
                    "url": "https://captcha.example.com/captcha/1.0.0/refresh?
↪uniqueFormHash=${ste.uniqueFormHash}&type=${cfg.type}&options[system]=${cfg.
↪system}&options[token]=${cfg.token}"
                },
                "name": "refresh"
            }
        ],
        "plainParams": {
            "type": "arithmetic"
        },
        "secureParams": {
            "bearerToken": "<access_token>",
            "system": "<system_id>",
            "token": "<system_token>"
        }
    }
}
```

В этом блоке содержатся параметры вызова трех методов сервиса CAPTCHA (`create`, `check`, `refresh`), а также секретные параметры – маркер доступа (`bearerToken`), идентификатор системы (`system`), а также токен системы (`token`).

- в блоке настроек входа по логину и паролю `blitz.prod.local.idp.password`. Внутри этого блока следует добавить блок `captcha` и настроить согласно примеру:

```
"captcha" : {
  "enabled": true,
  "initJs": "require(['https://demo.reaxoft.ru/themes/default/assets/js/
↪passwordCaptcha.js', 'captcha-conf'], function(captcha, conf){ captcha(conf,
↪'https://demo.reaxoft.ru/themes/default/assets/css/passwordCaptcha.css');});",
  "mode": {
    "type": "always_on"
  },
  "name": "exampleCaptcha"
}
```

В этом блоке следует настроить следующие параметры:

- `enabled` – признак того, включена CAPTCHA или нет (`true/false`);
- `initJs` – содержит ссылки на JS-скрипт и CSS-стили, загружаемые на странице входа и необходимые для отображения/вызова CAPTCHA на странице входа;
- `mode` – режим отображения CAPTCHA, предусмотрены следующие режимы:
  - `always_on` – CAPTCHA отображается всегда;
  - `on_header` – CAPTCHA отображается, если в запросе есть заголовок, указанный в параметре `name`, и значением, указанном в параметре `value`.
  - `by_brute_force_protection` – CAPTCHA отображается, если Blitz Identity Provider обнаружил подбор пароля к конкретной учетной записи или массовый подбор пароля ко всем учетным записям.

При использовании режима `by_brute_force_protection` требуется дополнительно создать в блоке `blitz.prod.local.idp.password` блок настроек `bruteForceProtection` со следуюшими настройками:

- `disabled` – выключена или нет защита (true/false);
- `captcha` – использовать ли тест CAPTCHA при срабатывании защиты (true/false);
- `delay` – время задержки входа в секундах (применяется, если выключено использование CAPTCHA);
- блок `system` в настройке `thresholds` – если необходима защита на уровне системы (защита от перебора на разные логины). Задаются настройки:
  - `minAttemptsToActivate` – минимальное кол-во прошедших входов для включения механизма защиты на основе статистики системы (по умолчанию 100 входов);
  - `timeWindowInMin` – временное окно сбора статистики по соотношению успешных и неуспешных входов в минутах, должно быть четным (по умолчанию 100 минут);
  - `failedAttemptsPercent`, настройка `turnOff` – порог выключения автоматической защиты, в процентах;
  - `failedAttemptsPercent`, настройка `turnOn` – порог включения автоматической защиты, в процентах.
  - `forced` – включить принудительно защиту для всех (true/false).
- блок `system` в настройке `thresholds` – если необходима защита на уровне отдельных пользователей (защита от подбора пароля на конкретного пользователя). Задаются настройки:
  - `ttlInSec` – период, за который накапливается счетчик неуспешных входов по пользователю в секундах (по умолчанию 3600 секунд);
  - `failedAttempts`, настройка `turnOn` – количество ошибочных входов за период, после которого для учетной записи включится защита.

Пример настроек блока `bruteForceProtection` (включена только защита на уровне пользователя):

```
"bruteForceProtection" : {
    "delay" : 0,
    "captcha" : true,
    "disabled" : false,
    "thresholds" : {
        "user" : {
            "failedAttempts" : {
                "turnOn" : 5
            },
            "ttlInSec" : 3600
        }
    }
}
```

Пример настроек `bruteForceProtection` (включена защита на уровне пользователя и на уровне системы):

```
"bruteForceProtection" : {
    "disabled": false,
    "delay" : 0,
    "captcha" : true,
    "thresholds" : {
        "system" : {
            "minAttemptsToActivate": 1000,
            "timeWindowInMin": 180,
            "failedAttemptsPercent" : {
                "turnOff" : 20,
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                "turnOn" : 30
            },
            "forced" : false
        },
        "user" : {
            "ttlInSec": 3600,
            "failedAttempts" : {
                "turnOn" : 5
            }
        }
    }
}
```

В случае использования в качестве CAPTCHA сервиса Google reCAPTCHA v3[65] необходимо:

- задать следующие настройки в `blitz.prod.local.idp.captcha`:

```
"captcha" : {
  "reCAPTCHAv3" : {
    "operations" : [
      {
        "call" : {
          "headers" : [],
          "method" : "post",
          "url" : "https://www.google.com/recaptcha/api/siteverify?secret=${cfg.
↪secret}&response=${ocp.response}"
        },
        "check" : {
          "errRegExp" : {},
          "okRegExp" : {
            "score" : "1\\.0|0\\.(5|6|7|8|9)",
            "success" : "true"
          }
        },
        "name" : "verify"
      }
    ],
    "plainParams" : {
      "sitekey" : "SITE_KEY"
    },
    "secureParams" : {
      "secret" : "SITE_SECRET"
    }
  }
}
```

Вместо `SITE_KEY` и `SITE_SECRET` нужно заполнить значения, полученные при регистрации Google reCAPTCHA v3 на сайте `https://g.co/recaptcha/v3`. Также нужно скорректировать значение в параметре `score` – установить требуемый порог успешного прохождения проверки (в примере выставлен порог не ниже 0,5).

- задать следующие настройки в `blitz.prod.local.idp.password.captcha`:

```
"captcha" : {
  "mode" : {
    "_name" : "X-Captcha-Check",
    "_value" : "true",
    "_type" : "on_header",
    "type" : "always_on"
```

(continues on next page)

[65] https://developers.google.com/recaptcha/docs/v3

(продолжение с предыдущей страницы)

```
  },
  "enabled" : true,
  "initJs" : "require(['/blitz/assets/blitz-common/javascripts/recaptcha_v3.js',
↪'captcha-conf'], function(captcha, conf){ captcha(conf);});",
  "name" : "reCAPTCHAv3"
}
```

Для добавления CAPTCHA на страницу подтверждения привязки учетной записи пользователя к учетной записи из внешнего поставщика идентификации необходимо задать следующие настройки в `blitz.prod.local.idp.externalIdps.captcha`:

```
"captcha" : {
  "mode" : {
    "_name" : "X-Captcha-Check",
    "_value" : "true",
    "_type" : "on_header",
    "type" : "always_on"
  },
  "enabled" : true,
  "initJs" : "require(['/blitz/assets/blitz-common/javascripts/recaptcha_v3.js',
↪'captcha-conf'], function(captcha, conf){ captcha(conf);});",
  "name" : "reCAPTCHAv3"
}
```

Для добавления CAPTCHA на страницу регистрации пользователей необходимо задать следующие настройки в `blitz.prod.local.idp.provisioning.registration.captcha`:

```
"captcha" : {
  "mode" : {
    "_name" : "X-Captcha-Check",
    "_value" : "true",
    "_type" : "on_header",
    "type" : "always_on"
  },
  "enabled" : true,
  "initJs" : "require(['/blitz/assets/blitz-common/javascripts/recaptcha_v3.js',
↪'captcha-conf'], function(captcha, conf){ captcha(conf);});",
  "name" : "reCAPTCHAv3"
}
```

Для добавления CAPTCHA на страницу восстановления пароля необходимо задать следующие настройки в `blitz.prod.local.idp.provisioning.recovery.captcha`:

```
"captcha" : {
  "mode" : {
    "_name" : "X-Captcha-Check",
    "_value" : "true",
    "_type" : "on_header",
    "type" : "always_on"
  },
  "enabled" : true,
  "initJs" : "require(['/blitz/assets/blitz-common/javascripts/recaptcha_v3.js',
↪'captcha-conf'], function(captcha, conf){ captcha(conf);});",
  "name" : "reCAPTCHAv3"
}
```

Сервер очередей

Отправка событий в сервер очередей

В сервер очередей могут быть отправлены следующие события:

- регистрация пользователя (`USER_REGISTERED`);
- смена пароля (`USER_PASSWORD_SET`);
- смена признака аннулирования сессий (`USER_CRID_CHANGED`);
- изменения атрибутов пользователя (`USER_ATTRIBUTE_CHANGED`);
- очистка атрибутов пользователя (`USER_ATTRIBUTE_REMOVED`);
- удаление пользователя (`USER_REMOVED`);
- привязка внешней учетной записи (`FEDERATION_POINT_BOUND`);
- отвязка внешней учетной записи (`FEDERATION_POINT_UNBOUND`);
- отзыв выданного приложению разрешения (`scopes`) (`SCOPES_REVOKED`);
- создание группы (`GROUP_CREATED`);
- изменение атрибутов группы (`GROUP_UPDATED`);
- удаление группы (`GROUP_REMOVED`);
- включение пользователя в группу (`GROUP_MEMBER_ADDED`);
- исключение пользователя из группы (`GROUP_MEMBER_REMOVED`).

Для отправки событий в очередь следует создать блок `blitz.prod.local.idp.events` следующего содержания (на примере регистрации пользователя и смены пароля):

```
"events" : {
    "drivers" : {
        "rabbit_driver" : {
            "properties" : {},
            "server" : {
                "host" : "<RMQ_HOST>",
                "port" : 5672
            },
            "type" : "RMQ",
            "user" : {
                "password" : "<RMQ_PASS>",
                "username" : "<RMQ_USERNAME>"
            }
        }
    },
    "routes" : {
        "USER_PASSWORD_SET" : [
            "password_sync"
        ],
        "USER_REGISTERED" : [
            "registration"
        ]
    },
    "targets" : [
        {
            "discardList" : "PSWD_SYNC_DISCARD",
            "driver" : {
                "ext" : {
                    "exchange_name" : "users",
                    "routing_key" : "pwd_sync"
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                },
                "id" : "rabbit_driver"
            },
            "encCertificate" : "rmqkey",
            "name" : "password_sync",
            "redelivery" : 3
        },
        {
            "discardList" : "REG_DISCARD",
            "driver" : {
                "ext" : {
                    "exchange_name" : "users",
                    "routing_key" : "registration"
                },
                "id" : "rabbit_driver"
            },
            "encCertificate" : "rmqkey",
            "name" : "registration",
            "redelivery" : 3
        }
    ]
}
```

В данных настройках следует задать:

- `RMQ_HOST` – домен сервера очередей RabbitMQ;
- `RMQ_USERNAME` – имя пользователя для работы с сервером очередей;
- `RMQ_PASS` – пароль для работы с сервером очередей.

Кроме того, для шифрования паролей, отправляемых в очередь (только для сообщений `USER_REGISTERED` и `USER_PASSWORD_SET`), в параметре `encCertificate` следует указать псевдоним ключа электронной подписи (в стандартном хранилище ключей `BlitzIdPKeystore.jks`), которым следует шифровать пароли в сообщениях.

Сервер очередей как брокер сообщений

В Blitz Identity Provider для обработки асинхронных задач применяется встроенный брокер сообщений, использующий для отслеживания задач базу данных.

При большой интенсивности запросов к Blitz Identity Provider может быть целесообразным использование сервера очередей RabbitMQ в качестве брокера сообщений. Для этого нужно в консоли RabbitMQ (обычно, `http://hostname:15672/`) выполнить следующие настройки:

- создать `queue` с именем `blitz-tasks` (в меню *«Queues»* консоли);
- создать `exchange` с именем `blitz-tasks-exh` (в меню *«Exchanges»* консоли) и настроить `binding` на очередь `blitz-tasks` с `routing_key` с именем `blitz-tasks`;
- создать пользователя `blitz` (в меню *«Admin»* консоли) и назначить ему права на созданную очередь.

После настройки RabbitMQ необходимо скорректировать настройки в `blitz.conf` – в блоке `blitz.prod.local.idp.tasks` установить `broker-type` в значение `rmq` и задать настройки подключения к RabbitMQ в блоке `broker-rmq`:

- в параметре `exchange` задать имя `blitz-tasks-exh`;
- в параметре `queue` в блоке `executionRules` и в параметре `name` в блоке `queues` задать имя `blitz-tasks`;
- в параметре `username` в блоке `user` задать имя пользователя (`blitz`);

- в параметре `password` в блоке `user` задать пароль пользователя в открытом виде – после запуска Blitz Identity Provider пароль будет зашифрован;
- в параметрах `host` и `port` блока `server` указать адрес и порт подключения к RabbitMQ;
- при необходимости скорректировать остальные параметры, определяющие размер пула соединений (`poolSize`), количества каналов (`channelSize`), время ожидания отклика от сервера очередей (`ackTimeout`);
- при необходимости скорректировать настройки брокера обработки задачи, определяющие количество попыток (`maxAttempts`) повторной обработки задач в случае ошибки, время между попытками (`redeliveryDelayInSec`), размер обрабатываемой пачки сообщений (`dequeueBatchSize`), период проверки очереди (`dequeuePeriodInSec`), количество обработчиков (`executorPoolSize`):

Пример конфига приведен ниже:

```
"tasks" : {
    "broker-type" : "rmq",
    "broker-rmq" : {
        "consumer" : {
            "poolSize" : 2
         },
         "exchange" : "blitz-task-exh",
         "publisher" : {
             "ackTimeout" : 15,
              "channelsSize" : 8,
              "poolSize" : 2
         },
         "server" : {
             "host" : "RMQ_HOST",
             "port" : 5672
         },
         "user" : {
             "password" : "CHANGE_ME",
             "username" : "blitz"
         }
    },
    "executionRules" : [
        {
            "maxAttempts" : 2,
            "queue" : "blitz-tasks",
            "redeliveryDelayInSec" : 60
        }
    ],
    "queues" : [
        {
            "dequeueBatchSize" : 10,
            "dequeuePeriodInSec" : 30,
            "executorPoolSize" : 5,
            "name" : "blitz-tasks"
        }
    ]
}
```

Хранилища и базы

Хранение объектов в Couchbase Server

Можно переназначить внутренние хранилища (`buckets`) Blitz Identity Provider в СУБД Couchbase Server, используемые для хранения данных. Предусмотрена возможность для следующих наборов данных указать необходимость использования иных хранилищ (`buckets`), чем стандартно используемые.

Для настройки иных хранилищ (`buckets`) нужно в блоке `blitz.prod.local.idp.internal-store-cb` добавить настройки:

- `buckets` – перечисление используемых хранилищ (`buckets`), в случае если отличаются от стандартных;
- `bucketsMapping` – переопределение стандартных размещений наборов данных на размещение в других хранилищах.

Пример настройки в конфигурационном файле представлен ниже. В результате набор данных `acl` размещается в хранилище `users`, а `clt` и `iat` – в `apps`. По умолчанию все три набора данных записывались в хранилище `oauth`.

```
"internal-store-cb" : {
  …
  "buckets" : {
    ["users", "oauth", "audit", "builtin_idstore", "ctxs"]
  },
  "bucketsMapping" : {
    "acl" : "users",
    "clt" : "apps",
    "iat" : "apps"
  },
  …
}
```

Считывание конфигурации кластера Couchbase Server

В случае недоступности одной из нод кластера Couchbase Server у пользователей могут возникать ошибки при входе в Blitz Identity Provider. В этом случае необходимо откалибровать значение интервала считывания глобальной конфигурации кластера. Если соединение с нодой прервется, конфигурация будет вовремя пересчитана и Blitz Identity Provider станет обращаться только к рабочим нодам. Выполните следующие действия:

1. Откройте файл конфигурации `/usr/share/identityblitz/blitz-config/blitz.conf`.
2. В секции `blitz.prod.local.idp.internal-store-cb.ioConf` задайте значение параметра `configPollInterval` в миллисекундах.

```
"internal-store-cb" : {
    "ioConf" : {
        "configPollInterval" : 2500
    },
    ...,
}
```

Время хранения объектов

Можно настроить для данных аудита ограничение по сроку хранения записей в базе данных (по умолчанию записи хранятся бессрочно). Для этого в блоке `blitz.prod.local.idp.internal-store-cb` нужно добавить настройку `ttlMapping` с указанием `doc_type` записи (`aud`) и времени хранения в секундах.

Пример настройки (время хранения аудита ограничено до 90 суток):

```
"internal-store-cb": {
  …
  "ttlMapping": {
    "aud": 7776000
  },
  …
}
```

Можно настроить для данных об устройствах ограничение по сроку хранения записей в базе данных. Для этого нужно в блоке `blitz.prod.local.idp.login` добавить настройки:

- `uaActiveTtlInSec` – время хранения записи об устройстве (в секундах), с которым связана долгосрочная сессия пользователя или которое пользователь при входе отметил в качестве доверенного. Если настройка не задана, то информация о таком устройстве хранится в течение года с последнего входа с этого устройства;
- `uaInactiveTtlInSec` – время хранения записи об остальных устройствах (в секундах). Если настройка не задана, то информация о таком устройстве хранится в течение 5 суток.

Пример настроек:

```
"login": {
  …
  "uaActiveTtlInSec": 2678400,
  "uaInactiveTtlInSec": 432000,
  …
}
```

Расширенные настройки подключения к PostgreSQL

Для расширенного управления пулом соединений с PostgreSQL или иной базой данных, поддерживающей JDBC, выполните следующие действия:

1. Откройте файл конфигурации `/usr/share/identityblitz/blitz-config/blitz.conf`.
2. В секции `blitz.prod.local.idp.internal-store-jdbc.pool` задайте следующие опциональные настройки:

Параметр	По умолчанию	Описание
testOnBorrow	true	Проверка состояния соединения перед отправкой данных. В случае ошибки соединение удаляется и выбирается следующее из пула.
testOnCreate	false	Проверка состояния соединения после создания в пуле.
testOnReturn	false	Проверка состояния соединения после возвращения в пул.
testWhileIdle	false	Проверка состояния соединения в состоянии idle. В случае ошибки соединение удаляется из пула.
timeBetweenEvictionRunsMilli	-1	Интервал в мс между запуском проверки соединения в состоянии idle, влияет на testWhileIdle.
validationQuery	-	SQL-запрос выполняемый при проверке состояния соединения из пула, при пустом значении используется isValid().

```
"internal-store-jdbc" : {
     "pool" : {
         "maxIdleConn" : 10,
         "maxTotalConn" : 20,
         "maxWaitConnMs" : 30000,
         "minIdleConn" : 7,
         "testOnBorrow" : false,
         "testOnCreate" : false,
         "testOnReturn" : false,
         "testWhileIdle" : true,
         "timeBetweenEvictionRunsMillis" : 30000,
         "validationQuery" : ""
     }
}
```

3. Перезапустите сервисы.

```
sudo systemctl restart blitz-idp blitz-console blitz-recovery blitz-
↪registration
```

Расширенные настройки подключения к LDAP

В консоли управления можно создать настройки подключения к хранилищам атрибутов, работающим по LDAP-протоколу. При этом через консоль управления можно задать настройки пула коннектов к LDAP. Blitz Identity Provider будет использовать общие настройки пула коннектов для установки подключений каждым приложением, использующим подключение к хранилищам. Это может привести к созданию большого числа коннектов к LDAP.

Через конфигурационный файл blitz.conf можно настроить параметры начального и максимального числа коннектов в разрезе различных приложений Blitz Identity Provider (например, для консоли управления задать меньшие значения коннектов в пуле, чем для сервиса аутентификации). Для этого в блоке blitz.prod.local.id-stores в настройках соответствующего хранилища наряду с настройками initialConnections и maxConnections можно создать настройки вида initialConnections#BLITZ_APP и maxConnections#BLITZ_APP, где в качестве BLITZ_APP указывается имя соответствующего приложения (blitz-console, blitz-idp, blitz-registration, blitz-recovery).

Пример настройки, когда для консоли управления задается меньший размер пула коннектов, чем для остальных приложений:

```
"id-stores" : {
  "list" : [
    {
      "type" : "LDAP",
      …
      "initialConnections" : 10,
      "initialConnections#blitz-console" : 1,
      "maxConnections" : 20,
      "maxConnections#blitz-console" : 1
    }
  ]
}
```

При выполнении запросов в LDAP хранилище атрибутов Blitz Identity Provider берет имеющееся соединение с LDAP-каталогом из пула соединений. После выполнения запроса Blitz Identity Provider не закрывает соединение, а возвращает его обратно в пул соединений для возможности повторного использования. Такой порядок взаимодействия с LDAP обеспечивает высокую производительность, но требует длительное время поддерживать соединения с LDAP-каталогом открытыми. Настройки межсетевых экранов или самих LDAP каталогов могут препятствовать длительному сохранению открытых соединений приложений Blitz Identity Provider с LDAP-каталогом.

TCP-соединения Blitz Identity Provider с LDAP-каталогом могут быть закрыты без согласованного разрыва соединения, так что в LDAP каталоге соединение будет закрыто, а Blitz Identity Provider об этом уведомлен не будет. При попытке использования такого соединения из пула может возникнуть длительный таймаут, прежде чем Blitz Identity Provider расценит соединение как закрытое и исключит его из пула соединений. Чтобы такая ситуация не влияла на пользователей, в Blitz Identity Provider предусмотрен алгоритм периодической проверки действительности открытых LDAP соединений. С периодом `healthCheckInterval` (в миллисекундах) выполняется проверка состояния соединения, а время таймаута при отсутствии ответа LDAP-каталога на запрос задается параметров `connectionTimeout` (в миллисекундах). Сам описанный режим оптимального взаимодействия с пулом соединений по умолчанию включен (настройка `useSyncMode` в значении `false`). В случае нестабильной работы соединений с LDAP-каталогом рекомендуется попробовать включить синхронный режим взаимодействия с каталогом (установить `useSyncMode` в значении `true`).

Примеры рекомендуемых настроек приведены ниже:

```
"id-stores" : {
  "list" : [
    {
      "type" : "LDAP",
      …
      "connectionTimeout" : 3000,
      "healthCheckInterval" : 300000,
      "useSyncMode" : false
    }
  ]
}
```

В случае подключения к Blitz Identity Provider одновременно нескольких хранилищ атрибутов может возникнуть такая ситуация, что при идентификации и аутентификации пользователя по логину и паролю в нескольких хранилищах может обнаружиться несколько учетных записей, возможно принадлежащих разным людям, с совпадающими логинами. Необходимо избегать такой ситуации при внедрении Blitz Identity Provider, и по умолчанию при выявлении такой ситуации Blitz Identity Provider при выявленных дублях будет выдавать пользователю ошибку входа, указывающую на наличие некорректной ситуации с учетной записью пользователя. Тем не менее, в ряде случае может возникнуть ситуация, когда при внедрении намеренно допускают, что по одному логину может быть найдено несколько учетных записей разных пользователей в разных хранилищах. В этом случае можно указать в блоке настроек `blitz.prod.local.idp.login` режим `firstSucceded` в настройке `authStrategy`. В этом случае все найденные учетные

записи будут проверены, и к какой из них первой подойдет пароль пользователя, с этой учетной записью и будет выполнен вход.

Пример настройки:

```
"login" : {
  "authStrategy" : {
    "mode" : "firstSucceeded"
  },
  …
}
```

База геоданных

Можно подключить к Blitz Identity Provider базу данных в формате mmdb[66] с геоданными. В этом случае Blitz Identity Provider при регистрации событий безопасности, а также при запоминании устройств и браузеров пользователя дополнительно к сохранению IP адреса будет записывать соответствующие IP-адресу данным о стране, регионе и городе, а также широту, долготу и радиусу точности, полученные из базы геоданных.

Сохраненные геоданные будут показываться администратору в консоли управления. Также можно включить отображение геоданных пользователю в «Личном кабинете» и включить их в тексты уведомлений, отправляемых по SMS или email.

Для подключения базы данных с геоданными необходимо выложить на серверах с Blitz Identity Provider файл формата mmdb с базой данных, а также создать блок настроек `blitz.prod.local.idp.geoIp` со следующими настройки в блоке `driver`:

- `type` – тип базы с геоданными. Поддерживается только тип `geoIp2-db`;
- `path` – путь на сервере к файлу с базой геоданных в формате `mmdb`.

Пример настроек:

```
"geoIp": {
    "driver": {
        "type": "geoIp2-db",
        "path": "geoIp/GeoIP2-City.mmdb"
    }
}
```

Использование нескольких СУБД

В Blitz Identity Provider можно настроить одновременное использование СУБД Couchbase Sever и СУБД PostgreSQL для хранения разных типов объектов. Для этого необходимо в блоке настроек `blitz.prod.local.idp.stores` задать следующие настройки:

- `default_type` – используемая по умолчанию СУБД. Возможные значения: `cb` – Couchbase Server, `jdbc` – PostgreSQL или иная реляционная СУБД с поддержкой JDBC;
- `list-of-types` – идентификаторы классов объектов Blitz Identity Provider и используемые для размещения соответствующих им объектов СУБД (`cb` или `jdbc`). Включать в настройку нужно только те классы объектов, которые размещаются в СУБД, отличной от указанной в `default_type`. Доступны следующие классы объектов:
  - `user-store` – атрибуты учетных записей;
  - `access-token-store` – маркеры безопасности;
  - `refresh-token-store` – маркеры обновления;

[66] https://www.maxmind.com/en/geoip2-databases

  - `id-ext-store` – привязки внешних поставщиков идентификации;
  - `device-code-store` – коды подтверждения для OAuth 2.0 Device Authorization Grant;
  - `access-list-store` – выданные пользователем разрешения приложениям;
  - `blitz-action-store` –коды подтверждения контактов (sms, email);
  - `oath-token-store` – привязки HOTP и TOTP генераторов разовых паролей;
  - `oath-load-proc-store` – история загрузок описаний аппаратных HOTP и TOTP генераторов разовых паролей;
  - `confirmation-request-store` – запросы разовых паролей;
  - `reg-context-store` – контекст регистрации пользователей;
  - `reg-context-storef` – контекст регистрации пользователей;
  - `id-store-maker` – встроенное хранилище идентификаторов пользователей;
  - `rcv-ctx-store` – контекст восстановления паролей пользователей;
  - `db-client-store` – динамические клиенты;
  - `db-client-storef` – динамические клиенты;
  - `initial-token-store` – IAT маркеры;
  - `user-agent-store` – устройства (браузеры) пользователей;
  - `web-authn-key-store` – ключи безопасности;
  - `audit-store` – события безопасности;
  - `task-store` – асинхронные задачи.

- `utils` – перечень модулей с утилитами, необходимых для используемого типа СУБД: `modules.CouchbaseModule` – для Couchbase Server, `modules.JDBCModule` – для PostgreSQL.

Пример настроек совместного использования двух СУБД:

```
"stores" : {
  "default-type" : "jdbc",
  "list-of-types" : {
    "access-token-store" : "cb",
    "refresh-token-store" : "cb",
    "user-agent-store" : "cb"
   },
  "utils" : [
    "modules.CouchbaseModule",
    "modules.JDBCModule"
  ]
}
```

Домен Blitz Identity Provider

Изменение домена Blitz Identity Provider осуществляется путем редактирования в блоке настроек `blitz.prod.local.idp.net` конфигурационного файла настройки `domain`.

Пример настройки:

```
"net" : {
"domain" : "demo.identityblitz.com"
}
```

При необходимости *изменить* (страница 298) в `blitz.prod.local.idp.lang` в блоке `portal-lang-cookie` значение настройки `domain`.

Пример фрагмента конфигурационного файла:

```
"lang" : {
  …
  "portal-lang-cookie" : {
    "domain" : "identityblitz.com",
    …
  }
}
```

При необходимости можно изменить путь до приложений (по умолчанию приложения доступны с использованием пути `/blitz`). Отредактировать путь можно в конфигурационном файле `play.conf`. Нужно изменить параметр `context` в блоке `play.http`:

```
"http" : {
"context" : "/blitz",
...
}
```

Изменить домен и путь Blitz Identity Provider в файлах `/blitz-config/saml/conf/relying-party.xml`, `/blitz-config/saml/metadata/idp-metadata.xml`.

Пример изменения настроек в `relying-party.xml`:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<ns18:RelyingPartyGroup …>
  <ns18:AnonymousRelyingParty
    provider="https://demo.identityblitz.com/blitz/saml"
    defaultSigningCredentialRef="IdPCredential"/>
  <ns18:DefaultRelyingParty
    provider="https://demo.identityblitz.com/blitz/saml"
    defaultSigningCredentialRef="IdPCredential">
    …
  </ns18:DefaultRelyingParty>
  …
</ns18:RelyingPartyGroup>
```

Пример изменения настроек в `idp-metadata.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<EntityDescriptor … entityID="https://demo.identityblitz.com/blitz/saml">
  <IDPSSODescriptor …>
    …
    <ArtifactResolutionService
      Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML1/SOAP/
↪ArtifactResolution"
      index="1"/>
    <ArtifactResolutionService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML2/SOAP/
↪ArtifactResolution"
      index="2"/>
    <SingleLogoutService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML2/Redirect/
↪SLO"
      ResponseLocation="https://demo.identityblitz.com/blitz/saml/profile/SAML2/
```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪Redirect/SLO"/>
    <SingleLogoutService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Plain-Redirect"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML2/Redirect/
↪Plain/SLO"
      ResponseLocation=
        "https://demo.identityblitz.com/blitz/saml/profile/SAML2/Redirect/Plain/SLO
↪"/>
    <SingleLogoutService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML2/SOAP/SLO" /
↪>
    …
    <SingleSignOnService
      Binding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"
      Location="https://demo.identityblitz.com/blitz/saml/profile/Shibboleth/SSO"/>
    <SingleSignOnService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML2/POST/SSO"/>
    <SingleSignOnService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-SimpleSign"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML2/POST-
↪SimpleSign/SSO"/>
    <SingleSignOnService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML2/Redirect/
↪SSO"/>
    <SingleSignOnService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Plain-Redirect"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML2/Redirect/
↪Plain/SSO"/>
  </IDPSSODescriptor>
  <AttributeAuthorityDescriptor …>
    …
    <AttributeService
      Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML1/SOAP/
↪AttributeQuery"/>
    <AttributeService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML2/SOAP/
↪AttributeQuery"/>
    …
    </AttributeAuthorityDescriptor>
</EntityDescriptor>
```

Пользователи

Блокирование неактивного пользователя

Blitz Identity Provider отслеживает время последней активности пользователя. Предусмотрена возможность выполнять блокирование учетных записей пользователей, которые долгое время неактивны. Для активации этой возможности необходимо запустить в cron выполнение скрипта `lockinactive.sh`. Скрипт находится директории `/usr/share/identityblitz/blitz-console/bin` на сервере с приложением blitz-console. Рекомендуется выполнять скрипт раз в день во время минимальной активности в системе. Перед запуском скрипта необходимо отредактировать его в текстовом редакторе – установить:

- `inactive_period` – требуемый период неактивности (в днях), после которого должна быть произведена блокировка учетной записи;

- `range_size` – диапазон охвата учетных записей (в днях), под блокировку попадут учетные записи, последняя активность по которым была в период с `(текущая дата – inactive_period – range_size)` до `(текущая дата – inactive_period)`.

Blitz Identity Provider позволяет также осуществлять автоматическое блокирование учетной записи в момент попытки входа, если до этого учетная запись была длительно неактивна. Для включения данной возможности нужно добавить блок настроек `blitz.prod.local.idp.lock` с значением в блоке `inactivity` настройки `limit` в секундах, определяющей максимально разрешенный период неактивности, по прошествии которого при попытке входа учетная запись будет заблокирована по неактивности. В настройке `checkInterval` можно задать минимальный период в секундах, не чаще которого при входе учетной записи будет проверяться срок неактивности.

Пример настройки:

```
"lock" : {
    "inactivity" : {
        "checkInterval" : 86400,
        "limit" : 31536000
    }
}
```

В настройках сервиса восстановления пароля можно включить режим, при котором будет разрешена разблокировка учетной записи, заблокированной по неактивности, в случае успешного прохождения *восстановления забытого пароля* (страница 199).

Запрет на использование ID удаленного пользователя

Blitz Identity Provider отслеживает использованные ранее идентификаторы пользователей, чтобы их нельзя было использовать повторно после удаления учетной записи пользователя в течение установленного периода времени. Для этого в блок `blitz.prod.local.idp.provisioning` нужно добавить раздел remove следующего содержания, указав нужное число дней (`days`), в течение которых идентификатор пользователя нельзя будет использовать при повторной регистрации:

```
"provisioning" : {
  …
  "remove": {
    "mode": "keepRemovedId",
    "days": 365
  }
}
```

ЕСИА

Контейнер ключей для ЕСИА и Цифрового профиля

Запросы на аутентификацию через внешний поставщики идентификации ЕСИА и ЦП ЕСИА должны быть подписаны электронной подписью с использованием контейнера ключей с алгоритмами ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012. Для использования в Blitz Identity Provider необходим файловый контейнер ключа электронной подписи в одном из следующих форматов:

- КриптоПро CSP (версия 4.0 или выше),
- КриптоПро JCP 2.0.

Для получения контейнера ключей необходимо обратиться в аккредитованный удостоверяющий центр.

Внимание: В случае если контейнер ключей выпускается удостоверяющим центром на физический носитель, то важно указать, что контейнер ключей должен быть выпущен с экспортируемым закрытым ключом, чтобы его можно было использовать в Blitz Identity Provider.

После получения в аккредитованном удостоверяющем центре контейнера ключей его необходимо импортировать в хранилище ключей, используемое Blitz Identity Provider. Для этого можно или обратиться в техническую поддержку Blitz Identity Provider или воспользоваться одним из описанных далее в подразделах способов конвертации.

КриптоПро CSP на Windows

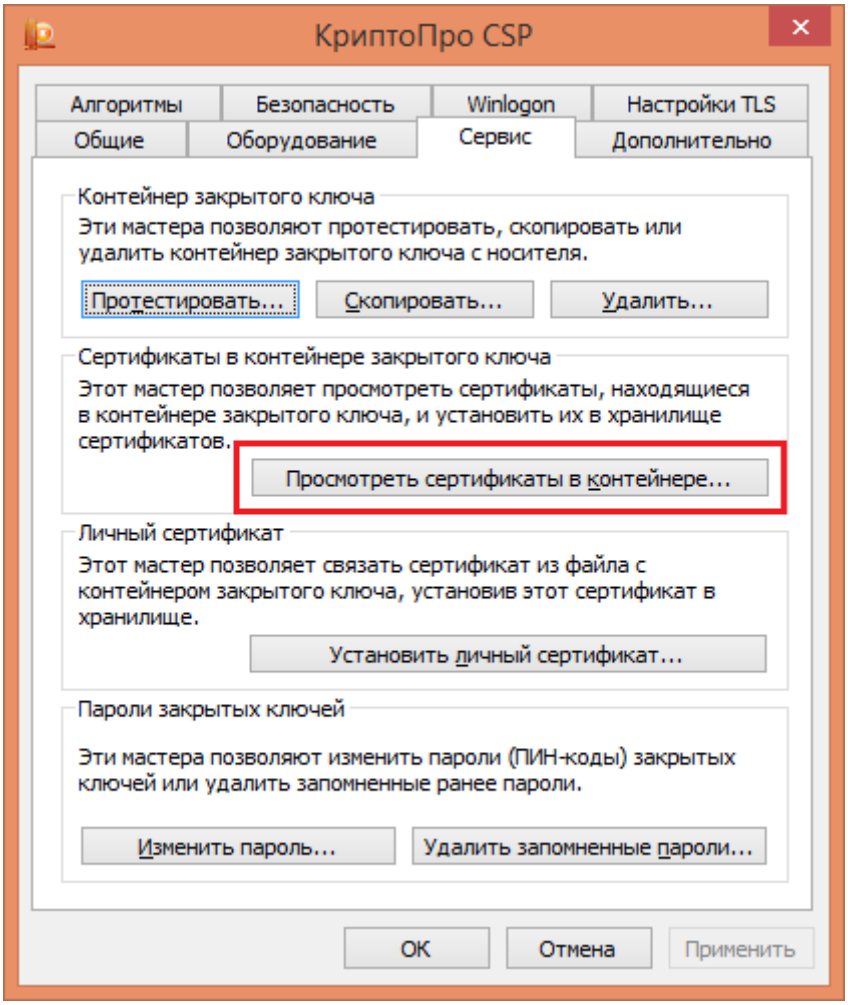
Для конвертации контейнера КриптоПро CSP в Windows понадобятся:

- ПК под управлением Windows с настроенным браузером Internet Explorer 11.
- Платная утилита P12FromGostCSP.
- Установленный на ПК криптопровайдер КриптоПро CSP (версия 4.0 и новее).

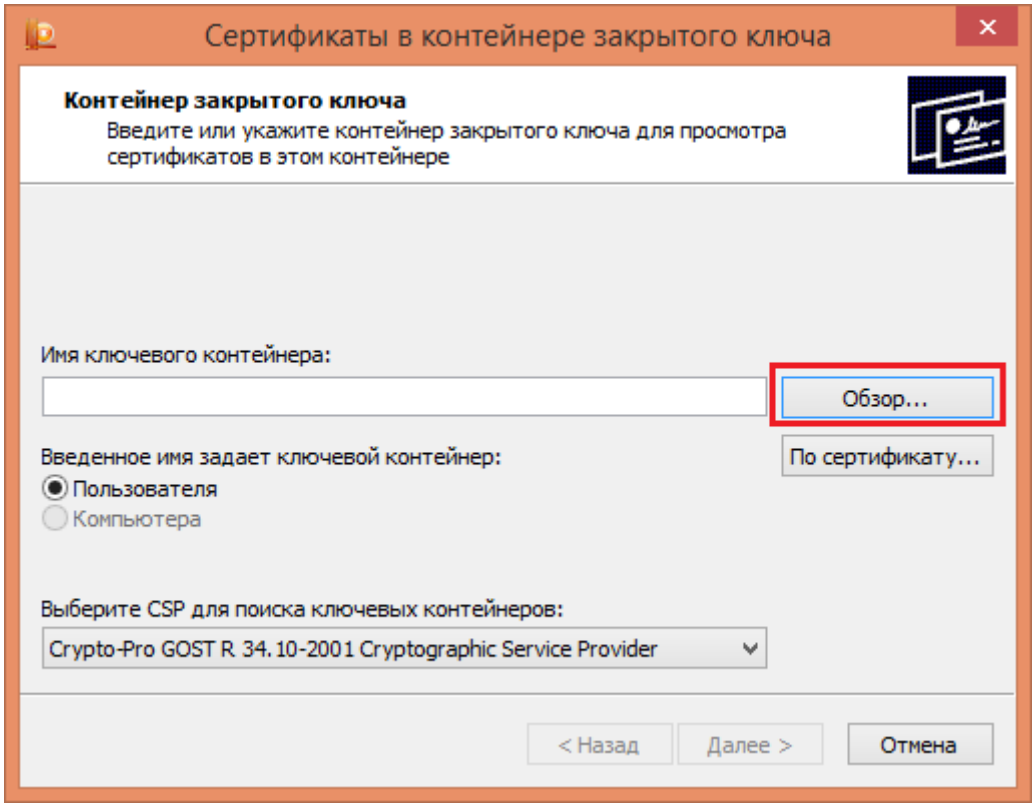
Инструкция по конвертации контейнера:

1. Установить на ПК под управлением Windows криптопровайдер КриптоПро CSP[67] (версия 4.0 и новее). Установить закрытый ключ электронной подписи в реестр Windows. Для этого выполнить шаги:
    - запустить КриптоПро CSP;
    - выбрать вкладку Сервис и нажать на кнопку Посмотреть сертификаты в контейнере:

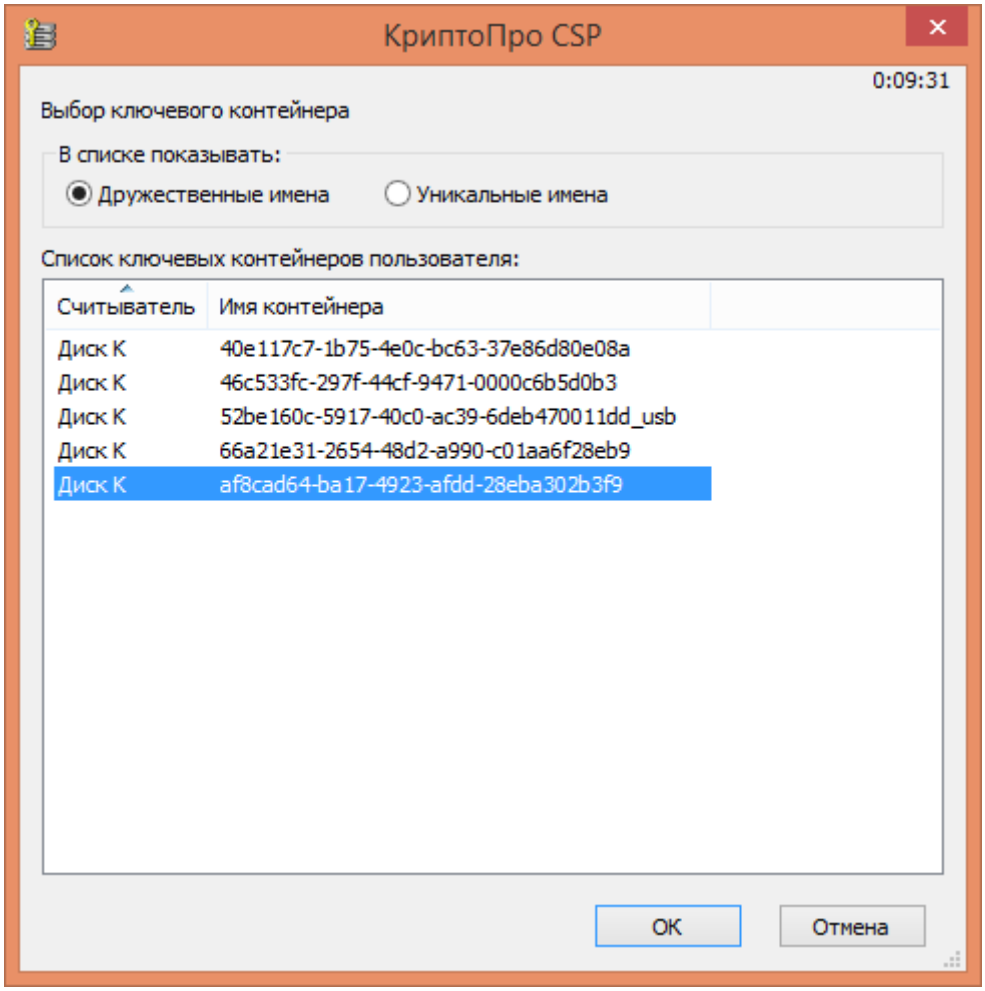
[67] https://cryptopro.ru



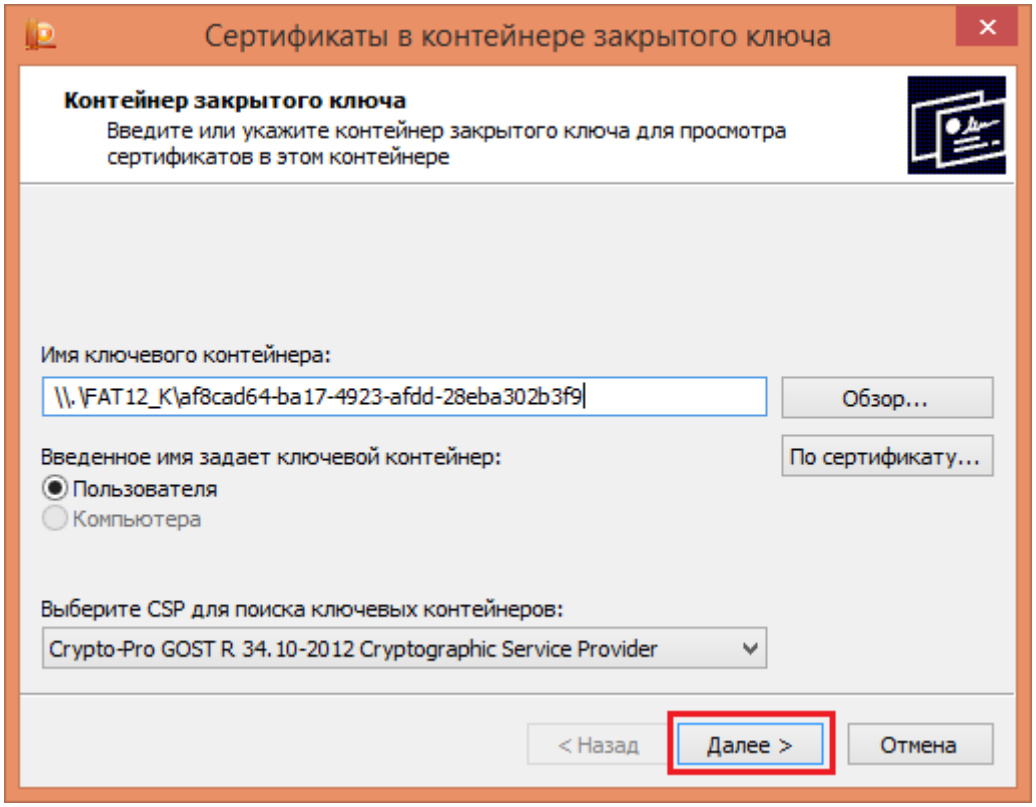
- нажать на кнопку Обзор:



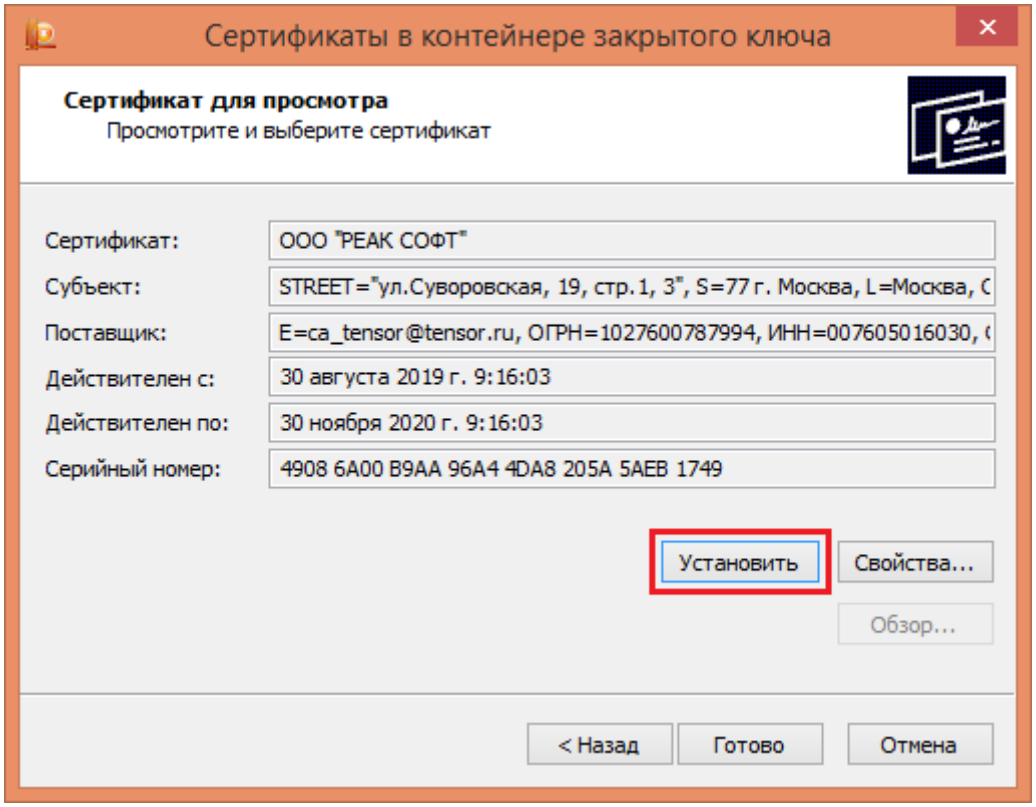
- выбрать контейнер с ключом ГОСТ Р 34.10-2012:



- в открывшемся окне убедиться, что выбран нужный ключевой контейнер и нажать далее:



- в окне со свойствами сертификата нажать на кнопку Установить:

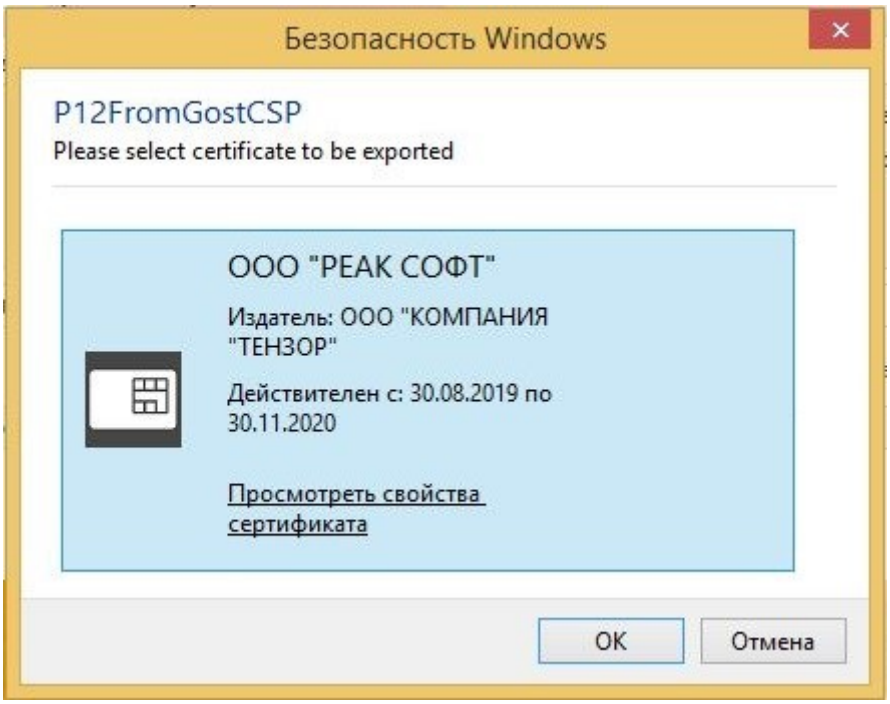


- сообщение Сертификат был установлен в хранилище «Личные» текущего пользователя свидетельствует об успешном сохранении.

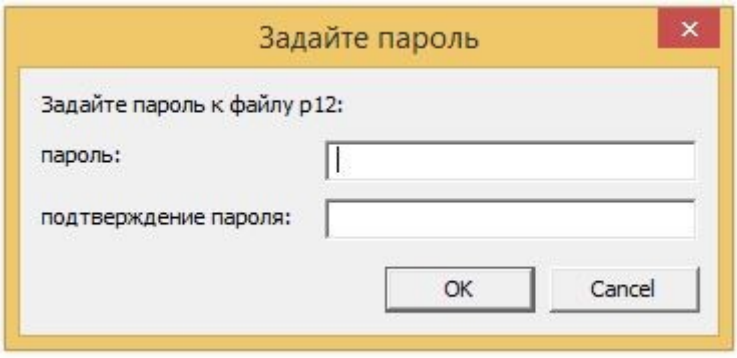
2. Экспортировать ключ КриптоПро из хранилища Windows в PKCS#12. Для этого приобрести и запустить утилиту P12FromGostCSP[68]. Нужна специальная версия утилиты – рекомендуется проконсультироваться с технической поддержкой Blitz Identity Provider.

   После запуска утилиты выполнить шаги:

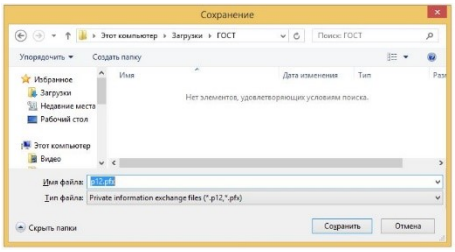
   - выбрать сертификат:



   - задать пароль от создаваемого контейнера PKCS#12:



   - указать файл для сохранения PKCS#12. В качестве расширения обязательно указать `.pfx`:



3. Установить на ПК окружение Java (JRE) версии 8.

[68] http://soft.lissi.ru/ls_product/utils/p12fromcsp/

4. Импортировать ключ из PKCS#12 в BKS-хранилище с помощью бесплатной утилиты gost-keytool[69] и актуальной версии библиотеки bcprov-jdk15on[70].

```
java -cp "gost-keytool.jar;bcprov-jdk15on-1.70.jar" ru.reaxoft.gost.Keytool␣
↪import_pkcs12 --srckeystore gost.pfx --srcstorepass 12345678 --srcalias csp_
↪exported --srckeypass 12345678 --destkeystore blitz-keystore.bks --
↪deststoretype BKS --deststorepass pass --destalias gost2012 --destkeypass␣
↪pass
```

В этой операции используются параметры:

- `srckeystore` – путь к файлу PKCS#12;
- `srcstorepass` – пароль от контейнера PKCS#12 (был задан на шаге 2);
- `srcalias` – имя ключа в контейнере PKCS#12. Нужно указать значение `csp_exported`;
- `srckeypass` – пароль к ключу в контейнере PKCS#12. Нужно указать то же самое значение, что и для параметра `srcstorepass`;
- `destkeystore` – путь к файлу BKS `blitz-keystore.bks`, взятому с сервера Blitz Identity Provider;
- `deststoretype` – тип хранилища. Нужно указать значение `BKS`;
- `deststorepass` – пароль к хранилищу BKS;
- `destalias` – имя (`alias`) ключа в BKS, например, `gost2012`;
- `destkeypass` – пароль к ключу в BKS. Нужно указать то же самое значение, что и для параметра `deststorepass`.

5. Заменить файл `blitz-keystore.bks` на сервере Blitz Identity Provider;
6. Перезапустить приложения Blitz Identity Provider.

КриптоПро CSP на Linux

Для конвертации контейнера КриптоПро CSP в Linux понадобятся:

- ПК или сервер под управлением одного из следующих Linux: Astra Linux 1.7, РЕД ОС 7.3, Альт. В случае использование сервера рекомендуется использовать иной сервер, чем те, на которых развернут Blitz Identity Provider.
- Бесплатная утилита CryptoPro PFX Decoder by li0ard.

Важно: Перед началом конвертации нужно сохранить контейнер ключей КриптоПро CSP в формат `pfx` (файл `sourcekey.pfx`) и сертификат открытого ключа в формат `cer` (`sourcecrt.cer`).

Инструкция по конвертации контейнера:

1. Переключить ОС на использование ГОСТ в библиотеке OpenSSL согласно инструкции для используемой ОС: Astra Linux[71], РЕД ОС 7.3[72], Альт[73]. Инструкция далее приводится на примере РЕД ОС 7.3 с включенной настройкой для использования ГОСТ в OpenSSL.
2. Включить настройку ГОСТ в OpenSSL:

[69] https://identityblitz.ru/wp-content/uploads/2019/11/gost-keytool.zip
[70] https://repo1.maven.org/maven2/org/bouncycastle/bcprov-jdk15on/
[71] https://wiki.astralinux.ru/pages/viewpage.action?pageId=27362269
[72] https://redos.red-soft.ru/base/manual/safe-redos/gost-in-openssl/
[73] https://www.altlinux.org/ГОСТ_в_OpenSSL

```
openssl-switch-config gost
```

3. Настроить зависимости (делается однократно перед первой конвертацией ключа на сервере):

   Установка модуля `asn1`:

```
pip3 install asn1
```

   Установка модуля `pyderasn`:

```
[fetch|wget] http://www.pyderasn.cypherpunks.ru/download/pyderasn-9.3.tar.zst
[fetch|wget] http://www.pyderasn.cypherpunks.ru/download/pyderasn-9.3.tar.zst.
↪asc
gpg --verify pyderasn-9.3.tar.zst.asc pyderasn-9.3.tar.zst
zstd -d < pyderasn-9.3.tar.zst | tar xf -
cd pyderasn-9.3
python setup.py install
```

   Установка `pygost`:

```
[fetch|wget] http://www.pygost.cypherpunks.ru/pygost-5.12.tar.zst
[fetch|wget] http://www.pygost.cypherpunks.ru/pygost-5.12.tar.zst.asc
gpg --verify pygost-5.12.tar.zst.asc pygost-5.12.tar.zst
zstd -d < pygost-5.12.tar.zst | tar xf -
cd pygost-5.12
python setup.py install
```

4. Загрузить утилиту `CryptoPro PFX Decoder by li0ard`:

```
[fetch|wget] https://raw.githubusercontent.com/li0ard/cpfx/pyderasn/cpfx.py
[fetch|wget] https://raw.githubusercontent.com/li0ard/cpfx/pyderasn/schemas.py
```

5. Конвертировать контейнер ключа из формата `pfx` в формат `PEM` с помощью `CryptoPro PFX Decoder by li0ard`:

```
python cpfx.py sourcekey.pfx
```

   Ответ утилиты будет содержать информацию о том, в какой файл был сохранен преобразованный контейнер. Например, `exported_4192476d-3e66-4963-8684-bd95d6be7967.pem`.

6. Конвертировать сертификат открытого ключа из формата `DER` в формат `PEM` с помощью OpenSSL:

```
openssl x509 -in sourcecrt.cer -inform der -out sourcecrt.pem
```

7. Собрать новый pfx-контейнер из преобразованных ключа и сертификата:

```
openssl pkcs12 -export -in sourcecrt.pem -inkey 4192476d-3e66-4963-8684-
↪bd95d6be7967.pem -out correct.pfx -name gost2012
```

8. Установить окружение Java (JRE) версии 8.

9. Импортировать ключ из `PKCS#12` в BKS-хранилище с помощью бесплатной утилиты gost-keytool[74] и актуальной версии библиотеки bcprov-jdk15on[75].

[74] https://identityblitz.ru/wp-content/uploads/2019/11/gost-keytool.zip
[75] https://repo1.maven.org/maven2/org/bouncycastle/bcprov-jdk15on/

```
java -cp gost-keytool.jar:bcprov-jdk15on-1.70.jar ru.reaxoft.gost.Keytool␣
↪import_pkcs12 --srckeystore correct.pfx --srcstorepass 1234 --srckeypass␣
↪1234 --destkeystore blitz-keystore.bks --deststoretype BKS --deststorepass␣
↪pass --destalias gost2012 --destkeypass pass --srcalias gost2012
```

В этой операции используются параметры:

- `srckeystore` – путь к сконвертированному pfx-контейнеру (`PKCS#12`);
- `srcstorepass` – пароль от контейнера `PKCS#12`;
- `srcalias` – имя ключа в контейнере `PKCS#12`. Нужно указать значение `gost2012`;
- `srckeypass` – пароль к ключу в контейнере `PKCS#12`. Нужно указать то же самое значение, что и для параметра `srcstorepass`;
- `destkeystore` – путь к файлу BKS `blitz-keystore.bks`, взятому с сервера Blitz Identity Provider;
- `deststoretype` – тип хранилища. Нужно указать значение `BKS`;
- `deststorepass` – пароль к хранилищу BKS;
- `destalias` – имя (alias) ключа в BKS, например, `gost2012`;
- `destkeypass` – пароль к ключу в BKS.

10. Заменить файл `blitz-keystore.bks` на сервере Blitz Identity Provider;
11. Перезапустить приложения Blitz Identity Provider.

КриптоПро JCP

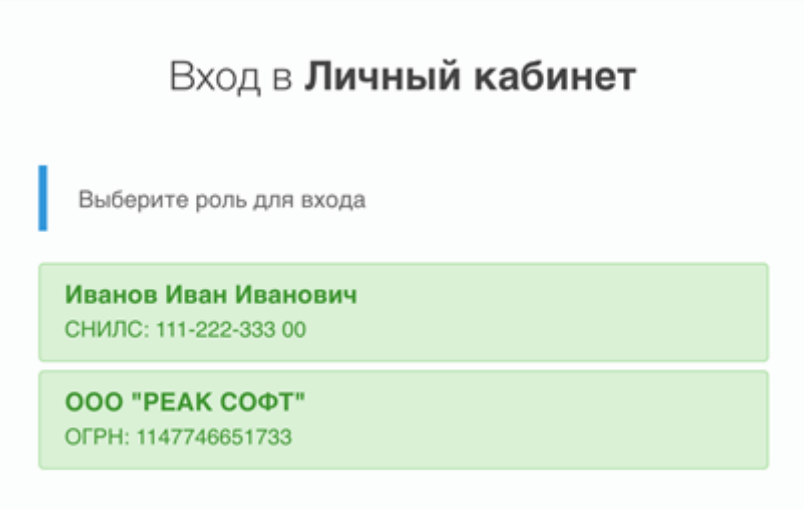
Для импорта контейнера в формате КриптоПРО JCP необходимо установить данный криптопровайдер[76] и следовать инструкциям из официальной документации.

Вход через ЕСИА в режиме выбора организации

Когда для входа в Blitz Identity Provider сконфигурирован *внешний поставщик идентификации ЕСИА* (страница 154), то к обычному режиму входа пользователя можно сконфигурировать следующие дополнительные возможности:

- Отображение пользователю экрана выбора режима входа и организации, если вошедший через ЕСИА пользователь имеет в ЕСИА роли сотрудника индивидуального предпринимателя, юридического лица или органа государственной власти.

[76] https://cryptopro.ru/products/csp/jcp



- Получение из ЕСИА сведений о выбранной при входе организации, автоматическое создание на основе этих сведений в LDAP-хранилище группы пользователей с атрибутами, соответствующими организации (если соответствующая организации группа не найдена в момент входа), добавление пользователя в созданную (или найденную) группу пользователей.
- Обновление атрибутов группы пользователя значениями атрибутов организации из ЕСИА в момент входа, если атрибуты в ЕСИА изменились.
- Возможность добавления в маркер доступа и маркер обновления сведений о выбранной в момент входа пользователя роли в ЕСИА (физическое лицо, индивидуальный предприниматель, должностное лицо юридического лица, должностное лицо органа государственной власти).

Для настройки режимов входа необходимо предварительно настроить в Blitz Identity Provider использование *групп доступа* (страница 213) и вход через *ЕСИА* (страница 154). После этого необходимо в конфигурационном файле в секции `blitz.prod.local.idp.federation` в блоке `esia` создать дополнительный блок настроек `org` следующего вида:

```
"federation" : {
  "points" : {
    "esia" : [
      {
        …
        "org" : {
          "embeds" : [
            "documents.elements-1",
            "addresses.elements-1",
            "contacts.elements-1"
          ],
          "group" : {
            "id" : "${org.oid}",
            "mapping" : [
              {
                "attr" : "org_ogrn",
                "master" : true,
                "value" : "${org.ogrn}"
              },
              {
                "attr" : "org_inn",
                "master" : true,
                "value" : "${org.inn}"
              },
              {
                "attr" : "org_fullname",
                "master" : true,
```

(continues on next page)

(продолжение с предыдущей страницы)

```
      "value" : "${org.fullName-}"
    },
    {
        "attr" : "org_shortname",
        "master" : true,
        "value" : "${org.shortName-}"
    },
    {
        "attr" : "org_type",
        "master" : true,
        "value" : "${org.type-}"
    },
    {
        "attr" : "org_oktmo",
        "master" : true,
        "value" : "${org.oktmo-}"
    },
    {
        "attr" : "org_leg",
        "master" : true,
        "value" : "${org.leg-}"
    },
    {
        "attr" : "org_kpp",
        "master" : true,
        "value" : "${org.kpp-}"
    },
    {
        "attr" : "org_phone",
        "master" : true,
        "value" : "${org.phone-}"
    },
    {
        "attr" : "org_email",
        "master" : true,
        "value" : "${org.email-}"
    },
    {
        "attr" : "org_fax",
        "master" : true,
        "value" : "${org.fax-}"
    },
    {
        "attr" : "org_agencytype",
        "master" : true,
        "value" : "${org.agencyType-}"
    },
    {
        "attr" : "org_agencyterrange",
        "master" : true,
        "value" : "${org.agencyTerRange-}"
    },
    {
        "attr" : "org_address_post",
        "master" : true,
        "value" : "${org.postAddress-}"
    },
    {
        "attr" : "org_address_leg",
        "master" : true,
        "value" : "${org.legalAddress-}"
```

(continues on next page)

(продолжение с предыдущей страницы)

```
            }
          ],
          "matchingRules" : [
            [
              {
                "attr" : "id",
                "value" : "${org.oid}"
              }
            ]
          ],
          "profile" : "orgs"
        },
        "scopes" : [
          "http://esia.gosuslugi.ru/org_addrs",
          "http://esia.gosuslugi.ru/org_leg",
          "http://esia.gosuslugi.ru/org_oktmo",
          "http://esia.gosuslugi.ru/org_inn",
          "http://esia.gosuslugi.ru/org_type",
          "http://esia.gosuslugi.ru/org_kpp",
          "http://esia.gosuslugi.ru/org_ctts",
          "http://esia.gosuslugi.ru/org_agencyterrange",
          "http://esia.gosuslugi.ru/org_ogrn",
          "http://esia.gosuslugi.ru/org_shortname",
          "http://esia.gosuslugi.ru/org_fullname",
          "http://esia.gosuslugi.ru/org_agencytype",
          "http://esia.gosuslugi.ru/org_grps"
        ]
      },
      …
    ]
  }
}
```

В добавленном блоке нужно скорректировать:

- набор получаемых из ЕСИА сведений об организации и их маппинг на атрибуты группы пользователей (блок `group.mapping`), признаком `master` отметить те атрибуты, которые должны перезаписываться в группе пользователей при каждом обновлении из ЕСИА, полученном в момент входа;
- набор запрашиваемых в ЕСИА разрешений (настройка `scopes`).

Если необходимо передавать в маркер идентификации и маркер доступа сведения о текущей выбранной организации и о роли пользователя в ЕСИА, то необходимо настроить соответствие необходимых атрибутов ЕСИА сессионным утверждениям в Blitz Identity Provider. Это выполняется с помощью настройки `claims` в блоке настроек ЕСИА:

```
"federation" : {
  "points" : {
    "esia" : [
      {
        …
        "claims" : [
          {
            "name" : "org_id",
            "value" : "org.oid"
          },
          {
            "name" : "global_role",
            "value" : "globalRole"
          },
          {
```

(continues on next page)

(продолжение с предыдущей страницы)

```
            "name" : "org_shortname",
            "value" : "org.shortName"
          },
          {
            "name" : "org_fullname",
            "value" : "org.fullName"
          },
          {
            "name" : "org_type",
            "value" : "org.type"
          },
          {
            "name" : "org_ogrn",
            "value" : "org.ogrn"
          },
          {
            "name" : "org_inn",
            "value" : "org.inn"
          },
          {
            "name" : "org_oktmo",
            "value" : "org.oktmo"
          },
          {
          "name" : "org_groups",
          "value" : "org.groupNames"
          },
        }
      },
      …
    ]
  }
}
```

Добавление параметров в вызов ЕСИА

Blitz Identity Provider позволяет добавлять в вызов сервиса аутентификации ЕСИА следующие параметры:

- `obj_type` – тип пользователя;
- `roles` – тип роли пользователя.

Для того чтобы включить передачу параметров, выполните следующие действия:

1. Откройте файл конфигурации `/usr/share/identityblitz/blitz-config/blitz.conf`.
2. В секции `blitz.prod.local.idp.federation.points.esia.authExtParams` (настройки ЕСИА) или `blitz.prod.local.idp.federation.points.esiadp.authExtParams` (цифрового профиля ЕСИА) укажите параметры для добавления в вызов сервиса аутентификации. Параметры необходимо перечислить в виде `<имя параметра>: <строковое значение параметра>`. Например, для передачи параметров `obj_type=P+L&roles=E+C`, нужно добавить в файл конфигурации соответствующего поставщика идентификации раздел вида:

```
"authExtParams": {
    "obj_type": "P L",
    "roles": "E C"
},
```

Сертификаты поставщиков WebAuthn, Passkey, FIDO2, U2F

См.также:

- *Вход с помощью WebAuthn, Passkey, FIDO2* (страница 97)
- *Подтверждение входа с помощью WebAuthn, Passkey, FIDO2, U2F* (страница 98)

Blitz Identity Provider позволяет переопределить перечень промежуточных и корневых сертификатов поставщиков ключей безопасности (WebAuthn, Passkey, FIDO2, U2F). Для этого нужно в блоке настроек `blitz.prod.local.idp.webAuthn.trustedStores` указать настройки, содержащие тип (`type`), файловый путь (`path`) и пароль (`password`) доступа к контейнеру ключей, который необходимо использовать для проверки подписи аттестационных объектов, формируемых при регистрации ключей безопасности. Стандартный контейнер ключей автоматически обновляется при установке новых версий Blitz Identity Provider и содержит актуальные корневые и промежуточные сертификаты TPM модулей, FIDO, а также сертификаты Apple и Google, необходимые для проверки подписи аттестационных объектов. При необходимости ограничить ключи безопасности до устройств определенных производителей нужно удалить из контейнера ключей лишние корневые и промежуточные сертификаты.

Пример настроек:

```
"webAuthn" : {
    …
    "trustedStores" : [
        {
            "password" : "*****",
            "path" : "webAuthn-trusted-ca.jks",
            "type" : "jKS"
        }
    ],
    …
}
```

OIDC, SAML и внешние поставщики идентификации

Сервис OIDC Discovery

Blitz Identity Provider автоматически публикует сервис OIDC Discovery[77] в соответствии с заданными в Blitz Identity Provider настройками. В составе сервиса можно прописать адрес документации на OIDC сервис. Чтобы задать свой адрес документации, необходимо в блоке настроек `blitz.prod.local.idp.oauth` прописать настройку `serviceDocumentationUrl` со значением адреса ссылки на документацию.

Адреса вызовов внешних поставщиков

При внедрении Blitz Identity Provider может возникнуть потребность настроить вызовы с серверов Blitz Identity Provider обработчиков внешних поставщиков идентификации не напрямую, а через прокси сервер. В этом случае есть необходимость изменить стандартные адреса обработчиков внешних поставщиков идентификации на адреса, зарегистрированные на прокси сервере. Чтобы скорректировать адреса обработчиков, необходимо изменить значения настроек `authUri`, `tokenUri`, `dataUri` в соответствующих блоках настроек внешних поставщиков идентификации в `blitz.prod.local.idp.federation`.

Пример настроек для входа через внешний поставщик Google:

```
"federation" : {
  "points" : {
```

(continues on next page)

[77] https://tools.ietf.org/html/rfc8414

(продолжение с предыдущей страницы)

```
    "google" : [
      {
        …
        "authUri" : "https://accounts.google.com/o/oauth2/auth",
        "tokenUri" : "https://accounts.google.com/o/oauth2/token",
        "dataUri" : "https://www.googleapis.com/oauth2/v1/userinfo?alt=json",
        …
      },
      …
    ]
  }
}
```

Внешний SAML-поставщик

Blitz Identity Provider позволяет настроить вход через внешний поставщик идентификации, работающий по протоколу SAML 2.0.

Для этого необходимо в блоке настроек `blitz.prod.local.idp.federations` создать внешний поставщик `saml` со следующими настройками:

- `name` – системное имя внешнего поставщика идентификации;
- `humanReadableName` – описание внешнего поставщика идентификации;
- `clientId` – имя поставщика услуг (`EntityId`), присвоенное Blitz Identity Provider при регистрации во внешнем SAML поставщике идентификации;
- `signAuthnReq` – определяет, должен ли Blitz Identity Provider подписывать SAML запрос, отправляемый внешнему поставщику идентификации;
- `checkAssertionSign` – определяет, необходимо ли проверять подпись SAML утверждений, полученных от внешнего поставщика идентификации (для ПРОД сред обязательно необходимо включать проверку подписи);
- блок `credentials` с настройками доступа к ключевому контейнеру, используемому для подписывания запросов к поставщику идентификации SAML. Настраивается опционально, в случае если для взаимодействия с внешним поставщиком идентификации требуется использовать отдельный контейнер ключей. Если настройка не задана, то ключи будут браться из основного keystore, настроенного в блоке `blitz.prod.local.idp.keystore` (при этом в качестве `alias` будет использоваться имя поставщика идентификации из настройки name).

  Задаются настройки:

  - `alias` – имя ключа в контейнере;
  - `keystore` – блок настроек, содержащий тип контейнера (`type`), который может быть JCEKS или BKS, а также путь к контейнеру (`path`) и пароль к контейнеру (`password`);
- `idpMetaPath` – путь к файлу, в котором хранятся метаданные внешнего поставщика идентификации (XML-файл с метаданными IDP);
- блок настроек `userMatching` – задает правила сопоставления учетных записей:
  - в настройке `type` – признак, что используется базовая (значение `builder`) настройка связывания учетных записей;
  - в настройке `mapping` – правила сопоставления учетных записей из внешнего SAML-поставщика идентификации учетным записям в Blitz Identity Provider;
  - в настройке `matchingRules` – правила переноса SAML-утверждений из внешнего поставщика идентификации в атрибуты учетной записи в Blitz Identity Provider;

- `requireLogInToBind` – признак *«Предлагать пользователю ввести логин и пароль для привязки, если учетная запись не была идентифицирована»*;
- `strictMatching` – признак *«Требовать ввод пароля, если учетная запись была идентифицирована»*;
- `uniqueMatching` – признак *«Для привязки должна быть найдена только одна учетная запись по заданным правилам соответствия»*.

Пример настроек внешнего поставщика идентификации:

```
"federation" : {
  "points" : {
    "saml" : [
      {
       "name" : "demo-idp",
        "humanReadableName" : "External SAML IDP",
        "clientId" : "login.company.com",
        "signAuthnReq" : true,
        "checkAssertionSign" : true,
        "_credentials" : {
            "alias" : "demo-idp",
            "keyStore" : {
              "password" : "*****",
              "path" : "demo-idp-key.jks",
              "type" : "JCEKS"
            }
        },
        "idpMetaPath" : "demo-idp-metadata.xml",
        "userMatching" : {
          "type" : "builder",
          "mapping" : [
            {
              "attr" : "urn:saml:mail",
              "master" : false,
              "value" : "${email}"
            }
          ],
          "matchingRules" : [
            [
              {
                "attr" : "urn:saml:mail",
                "value" : "${email}"
              }
            ]
          ],
          "requireLogInToBind" : false,
          "strictMatching" : false,
          "uniqueMatching" : false
        }
      }
    ],
    …
  }
}
```

После создания настроек внешнего поставщика необходимо включить его в списке доступных внешних поставщиков идентификации. Для этого в блок настроек `blitz.prod.local.idp.login` в перечень методов аутентификации (`methods`) в список внешних поставщиков входа `externalIdps` добавить внешний поставщик с `fedPoint`, соответствующий настроенному.

Пример настройки для включения внешнего поставщика идентификации с типом `saml` и именем `demo-idp`:

```
"login" : {
  …
  "methods" : {
    …
    "externalIdps" : {
      "idps" : [
        …
        {
          "fedPoint" : "saml:demo-idp"
        },
        …
      ],
      …
    },
    …
  },
  …
}
```

Настроить логотип (страница 311) для кнопки входа через внешний поставщик входа.

Внешний поставщик СУДИС

Blitz Identity Provider позволяет настроить вход через внешний поставщик идентификации СУДИС.

Для этого необходимо в блоке настроек `blitz.prod.local.idp.federations` создать внешний поставщик `sudis` со следующими настройками:

- `name` – системное имя внешнего поставщика идентификации;
- `humanReadableName` – описание внешнего поставщика идентификации;
- `clientId` – имя поставщика услуг (`EntityId`), присвоенное Blitz Identity Provider при регистрации в СУДИС;
- блок `credentials` с настройками доступа к ключевому контейнеру, используемому для подписывания и шифрования запросов к СУДИС. Должно использоваться ПО CMSServer, дистрибутив которого предоставляет поставщик СУДИС.

  Задаются настройки:

  - `endpoint` – адрес сервера CMSServer;
  - `fingerprint` – SHA1 хэш сертификата IDP СУДИС. Указывается в формате `{SHA1}значение`;
  - `recipientKey` – идентификатор ключа СУДИС (значение `cn` из `subject` сертификата СУДИС);
  - `senderKey` – идентификатор ключа Blitz Identity Provider, используемого для подписания запросов к СУДИС и расшифровывания ответов (значение `cn` из `subject` сертификата Blitz Identity Provider, зарегистрированного в СУДИС);
- `idpEntityId` – `EntityID` СУДИС;
- `sidAttrName` – имя атрибута, используемого в качестве идентификатора учетной записи в СУДИС;
- `ssoServiceLocation` – адрес обработчика запроса запросов на аутентификацию в СУДИС;
- `sloServiceResponseLocation` – адрес ответа на запросы логаута, инициированные в СУДИР из СУДИС;
- `sloServiceLocation` – адрес обработчика логаута в СУДИС при инициировании логаута из СУДИР в СУДИС;
- блок настроек `userMatching` – задает правила сопоставления учетных записей:

- в настройке `type` – признак, что используется базовая (значение `builder`) настройка связывания учетных записей;
- в настройке `mapping` – правила сопоставления учетных записей из внешнего SAML-поставщика идентификации учетным записям в Blitz Identity Provider;
- в настройке `matchingRules` – правила переноса SAML-утверждений из внешнего поставщика идентификации в атрибуты учетной записи в Blitz Identity Provider;
- `requireLogInToBind` – признак *«Предлагать пользователю ввести логин и пароль для привязки, если учетная запись не была идентифицирована»*;
- `strictMatching` – признак *«Требовать ввод пароля, если учетная запись была идентифицирована»*;
- `uniqueMatching` – признак *«Для привязки должна быть найдена только одна учетная запись по заданным правилам соответствия»*.

Пример настроек внешнего поставщика идентификации:

```
"federation" : {
  "points" : {
    "sudis" : [
    {
        "clientId" : "<ENTITY_ID BLITZ>",
        "credentials" : {
            "endpoint" : "http://<CMS_HOST>:<CMS_PORT>",
            "fingerprint" : "{SHA1}<FINGERPRINT SUDIS CERT>",
            "recipientKey" : "<SUDIS_KEY SUBJECT.CN>",
            "senderKey" : "<BLITZ_KEY SUBJECT.CN>"
        },
        "humanReadableName" : "SUDIS",
        "idpEntityId" : "http://…",
        "name" : "sudis",
        "sidAttrName" : "oid",
        "ssoServiceLocation" : "http://…/idp/profile/SAML2/POSTGOST/SSO",
        "sloServiceLocation" : "http://…/idp/Logout",
        "sloServiceResponseLocation" : "http://…/idp/profile/SAML2/Redirect/SLO",
        "userMatching" : {
            …
        }
    }
],
```

После создания настроек внешнего поставщика необходимо включить его в списке доступных внешних поставщиков идентификации. Для этого в блок настроек `blitz.prod.local.idp.login` в перечень методов аутентификации (`methods`) в список внешних поставщиков входа `externalIdps` добавить внешний поставщик с `fedPoint`, соответствующий настроенному.

Пример настройки для включения внешнего поставщика идентификации с типом `sudis` и именем `sudis`:

```
"login" : {
  …
  "methods" : {
    …
    "externalIdps" : {
      "idps" : [
        …
        {
          "fedPoint" : "sudis:sudis"
        },
        …
      ],
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        …
      },
      …
    },
    …
}
```

Режим регистрации незавершенных попыток входа

В Blitz Identity Provider все события фиксируются по факту окончания вызывавшего их процесса. Для большинства событий это нормально, так как процессы краткосрочные.

Среди всех регистрируемых событий есть важные события, связанные с входом пользователей. Если вход произошел успешно, то в самом конце процесса входа регистрируется событие безопасности, в котором указывается, кто, куда и когда вошел, какие методы аутентификации были задействованы, IP-адрес, UserAgent и много других деталей.

В зависимости от сделанных при внедрении настроек процесс входа может быть устроен сложно. Не всегда будет достаточно только ввести логин и пароль, и нужно будет дополнительно пройти подтверждение входа или в процессе входа пользователь будет взаимодействовать со вспомогательными приложениями (pipes), например, актуализировать контакт, настраивать Passkey или отвечать на вопрос, доверяет ли он устройству/браузеру. Если пользователь в какой-то момент этого процесса перестанет продолжать вход, то процесс входа не завершится, и как следствие, событие аудита о таком незавершенном входе не создастся. В зависимости от того, в какой момент это случится, это может быть проблемой безопасности. Например, если пользователь просто открыл страницу входа и не стал вводить логин и пароль, то фиксация такого события в журнале безопасности не представляет особого интереса. А вот если пользователь ввел правильные логин и пароль, но попал на экран подтверждения входа, который не стал проходить, то такое событие безопасности было бы хорошо зафиксировать. Возможно, злоумышленник перебирал пароль и смог успешно его подобрать, но не смог пройти проверку второго фактора аутентификации. Событие безопасности позволило бы узнать о такой ситуации, если бы оно было записано и анализировалось.

Для включения регистрации событий неуспешных (незавершенных) входов необходимо в блоке настроек `blitz.prod.local.idp.login` добавить параметры:

- `postponeEnabled` – значение `true`, если механизм включен;
- `postponeTtl` – время в секундах, после истечения которого регистрируется отложенное событие аудита, если вход не был завершен.

В случае если для обработки задач используется RabbitMQ, то для основной очереди задач необходимо сделать дополнительную очередь с названием `<название основной очереди>-postpone` и задать для нее следующие аргументы:

```
x-dead-letter-exchange = <используемый exchange>
x-dead-letter-routing-key = <основная очередь>
```

Так же для созданной очереди необходимо настроить `binding` на используемый `exchange`.

Передача событий безопасности в файл или Kafka

В Blitz Identity Provider можно настроить регистрацию событий безопасности в один или несколько приемников. Настройка задается в блоке настроек `blitz.prod.local.idp.audit`.

Необходимо задать следующие настройки:

- `emitters` – определяет список приемников записей аудита. По каждому приемнику заполняется блок настроек:
  - `type` – тип приемника. Возможные значения:
    - `audit-store` – запись производится в СУБД;
    - `log` – запись производится в логгер с именем `AUDIT`.
  - `enabled` – необязательная настройка – определяет, включен или нет приемник;
  - `include` – необязательная настройка – перечисляются типы событий безопасности (см. таблицу ниже), по которым осуществляется запись в приемник. Если настройка не указана, то пишутся все события безопасности;
  - `exclude` – необязательная настройка – перечисляются типы событий безопасности (см. таблицу ниже), которые не должны записываться в приемник. Если настройка не указана, то никакие события не исключаются. Если настройка указана вместе с `include`, то сначала список событий определяется настройкой `include`, а потом из него исключаются события, указанные в `exclude`. Рекомендуется не использовать совместно обе настройки `include` и `exclude`, а применять только что-то одно;
  - `logger` – необязательная настройка – указывается только для приемника с типом `log`. Позволяет определить имя логгера. Если настройка не задана, то запись производится в логгер с именем `AUDIT`;
  - `name` – необязательная настройка – указывается для приемников с типами `log` и `kafka`. Указывает имя приемника, так как для этих типов приемников можно настроить несколько приемников. Если настройка не задана, то используются `log` и `kafka` в качестве имен приемников;
  - `bootstrapServers` – обязательная настройка для приемника с типом `kafka` – указывается список адресов для первоначального подключения к кластеру Kafka;
  - `topic` – обязательная настройка для приемника с типом `kafka` – название топика Kafka, в который должно отправляться событие;
  - `securityProtocol` – необязательная настройка для приемника с типом `kafka` – в случае использования подключения по SASL может не указываться. При подключении по SSL в настройке должно быть указано значение `SSL`. Если в Kafka не настроен TLS, укажите значение `SASL_PLAINTEXT`;
  - `sasl` – необязательный блок настроек для приемника с типом `kafka` – задает параметры подключения при использовании SASL-аутентификации для подключения к Kafka:
    - `jaasConfig` – строка подключения, в которой можно использовать параметры подстановки из `secureParams`;
    - `mechanism` – значение `PLAIN`;
    - `secureParams` – блок с параметрами, которые будут зашифрованы в конфигурационном файле при запуске сервера.

  Пример блока:

```
"sasl": {
    "jaasConfig": "org.apache.kafka.common.security.plain.PlainLoginModule␣
↪required username=\"alice\" password=\"${pswd}\";",
    "mechanism": "PLAIN",
    "secureParams": {
        "pswd": "Содержимое зашифруется при запуске",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    }
}
```

- `ssl` – необязательный блок настроек для приемника с типом `kafka` – задает параметры SSL для подключения к Kafka:
  - `enabledProtocols` – строки со списком включенных протоколов;
  - `keyStore` – блок настроек с параметрами доступа к ключевому контейнеру Blitz Identity Provider. Содержит настройки `type`, `path`, `password`;
  - `trustedStore` – блок настроек с параметрами доступа к контейнеру с доверенными сертификатами. Содержит настройки `type`, `path`, `password`;
  - `keyPassword` – необязательная настройка – пароль для доступа к ключу.

Пример блока:

```
"securityProtocol" : "SSL",
"ssl" : {
    "enabledProtocols" : ["TLSv1.2,TLSv1.3"],
    "keyStore" : {
        "password" : "CHANGE-ME",
        "path" : "/etc/blitz-config/bip-d1app01-1.jks",
        "type" : "JKS"
    },
    "trustedStore" : {
        "password" : "CHANGE-ME",
        "path" : "/etc/blitz-config/ca.jks",
        "type" : "JKS"
    },
    "keyPassword": "CHANGE-ME"
},
```

- `tuning` – необязательный блок настроек для приемника с типом `kafka` – задает опциональные настройки `producer` для взаимодействия с Kafka. Имена параметров необходимо указывать с точкой как в документации Kafka[78].

Пример блока:

```
"tuning": {
    "client.id": "BlitzKafka"
}
```

- `emitAtLeastOneOf` – необязательная настройка – указывается список приемников, достаточно записи событий в любой из которых, чтобы операция считалась успешной;
- `emitToAllOf` – необязательная настройка – указывается список приемников, по которым обязательно должно быть получено подтверждение успешной записи события, чтобы операция считалась успешной. Если настройки `emitAtLeastOneOf` и `emitToAllOf` не заданы, то обязательно подтверждение от всех настроенных приемников;
- `emitTimeoutInSec` – необязательная настройка – определяет максимальное время отклика от приемника в ответ на запроса записи события. Если настройка не задана, то ожидание 60 секунд.

Пример настроек записи аудита одновременно в лог, в СУБД и в Kafka:

```
"audit": {
    "emitters": [
```

(continues on next page)

[78] https://kafka.apache.org/documentation/#producerconfigs

(продолжение с предыдущей страницы)

```
        {
            "type": "log",
            "name": "users-log",
            "enabled": true,
            "logger": "AUDIT",
            "exclude": ["admin_added", "admin_pswd_changed", "admin_removed",
↪"admin_roles_changed",
                        "config_changed"]
        },
        {
            "type": "log",
            "name": "admins-log",
            "enabled": true,
            "logger": "AUDITADMIN",
            "include": ["admin_added", "admin_pswd_changed", "admin_removed",
↪"admin_roles_changed",
                        "config_changed"]
        },
        {
            "type": "audit-store",
            "enabled": true
        },
        {
            "type" : "kafka",
            "enabled": true,
            "name" : "kafka",
            "include": ["login"],
            "bootstrapServers" : ["infra-kfk01:9443"],
            "topic" : "blitz_audit",
            "securityProtocol" : "SSL",
            "ssl" : {
                "enabledProtocols" : ["TLSv1.2,TLSv1.3"],
                "keyStore" : {
                    "password" : "CHANGE-ME",
                    "path" : "/etc/blitz-config/bip-app01.jks",
                    "type" : "JKS"
                },
                "trustedStore" : {
                    "password" : "CHANGE-ME",
                    "path" : "/etc/blitz-config/ca.jks",
                    "type" : "JKS"
                }
            },
        }
    ],
    "emitAtLeastOneOf": ["users-log","admins-log","kafka"],
    "emitToAllOf": ["audit-store"],
    "emitTimeoutInSec": 30
}
```

При регистрации аудита в лог можно настроить логгер с помощью файла конфигурации `logback.xml` (см. подробнее[79]). Пример настройки логгера `AUDIT` в файле конфигурации `logback.xml`:

```
…
    <appender name="AUDIT" class="ch.qos.logback.core.rolling.RollingFileAppender">
        <file>${dir.logs}/audit-${app.name}.log</file>
        <encoder>
            <pattern>%date - [%level] -[%file:%line] - %message%n%xException{20}</
↪pattern>
```

(continues on next page)

[79] https://logback.qos.ch/documentation.html

(продолжение с предыдущей страницы)

```
        </encoder>
        <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
            <fileNamePattern>${dir.logs}/archive/audit-${app.name}.%d{yyyy-MM-dd}.
↪log.gz</fileNamePattern>
            <maxHistory>90</maxHistory>
            <totalSizeCap>5GB</totalSizeCap>
        </rollingPolicy>
    </appender>

    <logger name="AUDIT" additivity="false">
        <appender-ref ref="AUDIT" />
    </logger>
…
```

Пример записи в логе:

```
2023-11-20 13:29:47,170 - [INFO] -[LoggerEventEmitterDriver.scala:37] - {"ip_st":
↪"Tashkent","ip":"213.230.116.179","authnDone":"true","process_id":"b80ca03e-4718-
↪44ff-9456-7d4255610eaa","ip_ctr":"Узбекистан","type":"login","object_id":"BIP-
↪123456","protocol":"oAuth","subject_id":"BIP-123456","auth_methods":"cls:password
↪","session_id":"f8d85ba2-a26a-447f-b82e-944b9218abb8","timestamp":1700476187069,
↪"ch_platform_version":"\"14.1.0\"","ch_platform":"\"macOS\"","ip_ct":"Tashkent",
↪"id_store":"ldap01","ip_lng":"69.2494","ip_rad":"5","ch_ua":"\"Google Chrome\";
↪v=\"119\", \"Chromium\";v=\"119\", \"Not?A_Brand\";v=\"24\"","user_agent":
↪"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like␣
↪Gecko) Chrome/119.0.0.0 Safari/537.36","lp_id":"test-system","id":
↪"6056828858453673-600312119","ip_lat":"41.3171","client_auth_method":"redirectUri
↪"}
```

Возможные типы событий безопасности:

- `admin_added` – добавлен администратор
- `admin_pswd_changed` – изменен пароль администратора
- `admin_removed` – удален администратор
- `admin_roles_changed` – изменены роли администратора
- `app_password_changed` – задан пароль для приложения
- `attribute_changed` – добавлен, изменен или удален атрибут
- `attribute_confirmed` – атрибут подтвержден
- `auth` – выполнена аутентификация (при OAuth 2.0 Resource Owner Password Credentials)
- `auth_failed` – ошибка аутентификации
- `auth_req` – запрос на аутентификацию
- `authz_granted` – выдано OAuth-разрешение
- `authz_rejected` – отказано в выдаче OAuth-разрешения
- `authz_revoked` – отозвано OAuth-разрешение
- `bind_ext_account` – учетная запись привязана к внешней
- `config_changed` – изменены настройки конфигурации
- `duo_put` – мобильное приложение Duo Mobile привязано
- `duo_remove` – мобильное приложение Duo Mobile отвязано
- `grant_right` – назначение прав доступа
- `group_attr_changed` – у группы пользователей изменен или удален атрибут

- `group_registered` – группа пользователей создана
- `group_removed` – группа пользователей удалена
- `hotp_attached` – привязан HOTP-генератор
- `hotp_detached` – отвязан HOTP-генератор
- `internal_user_deleted` – учетная запись удалена
- `locked_methods_changed` – изменен список заблокированных методов аутентификации
- `login` – выполнен вход
- `login_failed` – ошибка входа
- `login_stopped` – неуспешный вход
- `logout` – выполнен выход
- `logout_req` – запрос на выход
- `member_added` – пользователь включен в группу пользователей
- `member_removed` – пользователь исключен из группы пользователей
- `need_password_change` – установлен признак необходимости смены пароля
- `recovery` – доступ к учетной записи восстановлен
- `recovery_fail` – восстановление доступа не выполнено
- `recovery_req` – выполнен запрос на восстановление доступа
- `registration` – учетная запись зарегистрирована
- `registration_req` – выполнен запрос на регистрацию
- `required_factor_changed` – изменен режим аутентификации пользователя
- `reset_user_password` – пароль установлен администратором
- `reset_user_sessions` – выход с устройств (сброс сессий)
- `revoke_right` – отзыв прав доступа
- `send_email_code` – код подтверждения отправлен на email
- `send_push_code` – код подтверждения отправлен в Push
- `send_sms_code` – код подтверждения отправлен по SMS
- `token_exchange_failed` – отказано в обмене маркера доступа
- `token_exchanged` – произведен обмен маркера доступа
- `token_granted` – выдан маркер доступа
- `totp_attached` – привязан TOTP-генератор
- `totp_detached` – отвязан TOTP-генератор
- `unbind_ext_account` – учетная запись отвязана от внешней
- `user_locked` – учетная запись заблокирована
- `user_password_changed` – изменен пароль пользователя
- `user_sec_qsn_changed` – изменен контрольный вопрос
- `user_sec_qsn_removed` – удален контрольный вопрос
- `user_unlocked` – учетная запись разблокирована
- `web_authn_reg_key` – добавлен ключ безопасности
- `web_authn_revoke_key` – удален ключ безопасности

Набор атрибутов записи может отличаться в зависимости от типа события безопасности и особенностей процесса входа. Назначения атрибутов в записи аудита приведены в таблице:

Назначение атрибутов в записи аудита

Атрибут	Назначение и возможные значения
`id`	Идентификатор записи о событии безопасности
`type`	Тип события безопасности
`alt_pswd_cause`	Причина, по которой пользователя просили сменить пароль. Возможные значения: • `password_expired` – пароль просрочен • `password_reset` – пароль нужно сменить при первом входе • `password_policy_violated` – пароль не соответствует парольной политике
`attr_name`	Имя установленного, удаленного или измененного атрибута
`auth_methods`	Содержит список пройденных пользователем методов аутентификации. Возможные значения: • `password` – парольная аутентификация • `spnego` – вход с помощью сеанса ОС • `x.509` – вход с помощью средства электронной подписи • `qrCode` – вход по QR-коду • `tls` – вход с помощью HTTP-заголовков прокси-сервера • `webAuthn` – вход или подтверждение входа с помощью ключей безопасности • `css` – автоматический вход по результатам регистрации пользователя или восстановления пароля • `sms` – одноразовый пароль по SMS • `email` – одноразовый пароль по email • `hotp` – второй фактор аутентификации с помощью аппаратного брелока • `totp` – второй фактор аутентификации с помощью программного TOTP-генератора кодов подтверждения • `externalIdps:<type>:<name>` – вход с помощью внешнего поставщика идентификации (соцсети и пр.) • `userApp` – вторичная аутентификация в мобильном приложении • `outside_%NAME%` – внешний метод входа с именем %NAME% Наличие перед методом префикса `cls:` означает, что вход был выполнен с помощью долгосрочной сессии, а ранее при первичном входе использовались те методы входа, что перечислены после `cls:`
`auth_soft_id`	Приложение-аутентификатор (при входе по QR-коду)
`authnDone`	Проводилась ли аутентификация при этом входе
`captcha_passed`	Признак, что при входе спрашивалась CAPTCHA

continues on next page

Таблица 1 – продолжение с предыдущей страницы

Атрибут	Назначение и возможные значения
client_auth_method	Способ аутентификации вызвавшего Blitz Identity Provider приложения: • `internal` – для событий, вызванных внутренними приложениями Blitz Identity Provider • `x.509` – для событий, вызванных SAML-приложениями, при условии, что SAML-запрос пришел подписанным • `Basic` – для приложений, вызывающих REST-сервисы, использующие Basic-авторизацию • `redirectUri` – для приложений, которые идентифицировали себя в URL (например, указали свой client_id в URL-параметре), но чья аутентификация не проводилась (достоверно не известно, что это действительно вызывает Blitz Identity Provider именно это приложение) • `Bearer` – использование access_token для аутентификации мобильным приложением с динамическими client_id/client_secret
`dcId`	Динамический `client_id`
`device`	ID устройства
deviceFingerprint	Отпечаток устройства
`dTyp`	Тип устройства (при динамической регистрации)
`email`	Адрес электронной почты
`entry_point`	Тип интерфейса, использованного для регистрации пользователя: • `WEB` – при регистрации из веб-приложения Blitz Identity Provider, • `REST` – при регистрации через REST-сервисы Blitz Identity Provider.
`error`	Ошибка (при неуспешных событиях)
`ext_account_id`	Идентификатор внешней учетной записи
`ext_account_name`	Имя внешнего поставщика идентификации
`ext_account_type`	Тип внешнего поставщика идентификации
`failed_method`	Указывает, какой метод аутентификации не смог пройти пользователь
`group_id`	Идентификатор группы пользователей
`group_profile`	Идентификатор профиля использования групп пользователей
`id_store`	Хранилище учетной записи
`ip`	IP адрес пользователя
`ip_ctr`	Страна по IP адресу
`ip_st`	Регион по IP адресу
`ip_ct`	Город по IP адресу
`ip_lat`	Широта по IP адресу
`ip_lng`	Долгота по IP адресу
`ip_rad`	Окрестность по IP адресу
`lp_id`	Идентификатор приложения (`EntityId` для SAML или `client_id` для OIDC), вызвавшего Blitz Identity Provider.

continues on next page

Таблица 1 – продолжение с предыдущей страницы

Атрибут	Назначение и возможные значения
mobile	Номер мобильного телефона
module	Идентификатор измененного блока настроек
new_attr_value	Новое значение установленного или измененного атрибута
new_factor	Новое значение признака, указывающего на необходимость проверки второго фактора аутентификации
new_roles	Роли, добавленные учетной записи администратора
oauth_scopes	Список разрешений, которые выдал или отозвал пользователь
object_id	Идентификатор объекта операции (пользователь, по которому выполнялась операция)
old_attr_value	Прежнее значение удаленного или измененного атрибута
old_factor	Прежнее значение признака, указывающего на необходимость проверки второго фактора аутентификации
old_roles	Роли, отозванные у учетной записи администратора
origin_app	Идентификатор приложения, инициировавшего регистрацию пользователя или восстановление пароля
process_id	Идентификатор процесса
protocol	Протокол взаимодействия приложения с Blitz Identity Provider. Возможные значения: • SAML – для SAML и WS-Federation • OAuth – для OpenID Connect и OAuth 2.0 • simple – для прокси-аутентификации– • internal – для входа в Личный кабинет (_blitz_profile)
pswd_changed	Признак, что рекомендовалась смена пароля
pswd_tmp_locked	Признак, что была временная блокировка
recovery_contact	Указанный при восстановлении контакт (email или номер мобильного телефона)
recovery_type	Тип восстановления пароля: email или mobile
right_name	Название права доступа
roles	Роли учетной записи администратора
session_id	Уникальный идентификатор сессии пользователя. Позволяет коррелировать все события пользователя, выполненные им в рамках общей пользовательской сессии
subject_id	Идентификатор субъекта операции (пользователь, который вызвал операцию)
tags	Метка назначенного или отозванного права доступа
timestamp	Дата и время события. Например, 2022-11-04T17:49:58.384+0300
tried_old_pswd	Признак того, что была попытка входа с паролем из сохраненной истории паролей (предыдущим паролем)
used_login	Логин, использованный при входе

continues on next page

Таблица 1 – продолжение с предыдущей страницы

Атрибут	Назначение и возможные значения
user_agent	Данные о пользовательском устройстве (UserAgent)
wa_key_id	Идентификатор ключа безопасности
wa_key_name	Имя ключа безопасности
withDelay	Включалась задержка при входе

Хранение настроек приложений в отдельных файлах

По умолчанию настройки подключенных приложений хранятся внутри основного конфигурационного файла blitz.conf в блоке настроек blitz.prod.local.idp.apps. Если планируется подключать к Blitz Identity Provider большое число приложений (сотни приложений), то более предпочтительным может быть настроить хранение настроек приложений в отдельных конфигурационных файлах. Для этого нужно:

1. В каталоге настроек /usr/share/identityblitz/blitz-config создать корневой каталог для хранения настроек приложений (каталог настроек приложений). По умолчанию используется каталог /usr/share/identityblitz/blitz-config/apps.
2. Внутри каталога настроек приложений создать для каждого приложения свой каталог, соблюдая следующие правила:
   - имя каталога должно быть создано из идентификатора приложения (appId);
   - если в идентификаторе приложения был символ /, то его надо заменить на #;
   - если в идентификаторе приложения был символ :, то его надо заменить на %.

   Примечание: Например, для приложения с идентификатором https://example.com должен быть создан каталог с именем https%##example.com.

   Важно: Обязательно должны быть созданы каталоги для служебных приложений _blitz_console, _blitz_idp, _blitz_reg, _blitz_recovery, _blitz_profile.

3. Внутри каждого каталога приложения должен быть создан файл с именем app.conf, содержащий конфигурацию приложения из исходного blitz.conf. Раздел должен называться app, а не значение appId, как было в blitz.conf. Внутри каталога приложения также будет создан скрытый каталог .snapshot, куда будут копироваться предыдущие конфигурации приложений в случае изменения настроек приложения через консоль или API.

   Пример конфигурационного файла app.conf:

```
################################################################################
↪############
# version: 822
# modified: 2023-08-20 21:17:27 MSK
# author: admin
# ip: 127.0.0.1
# user agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.
↪36 …
################################################################################
↪############
{
    "app": {
        "domain": "https://company.com",
        "name": "test app",
        "oauth": {
            …
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        },
        ...
    }
}
```

4. После миграции всех существующих настроек приложений из `blitz.conf` в отдельные файлы настроек задать в `blitz.conf` в блоке `blitz.prod.local.idp.apps-source` режим чтения настроек приложений из отдельных файлов:

```
"apps-source": {
    "type": "filesystem",
    "dir": "apps"
 }
```

5. Перезапустить приложения Blitz Identity Provider и проверить корректность работы входа в приложения. Если все работает корректно, то при желании можно удалить настройки приложений из блока `blitz.prod.local.idp.apps`.

Продолжительность SSO-сессии

На продолжительность SSO-сессии пользователя может влиять срок действия cookie `blc` на стороне Blitz Identity Provider. По умолчанию срок действия cookie `blc` составляет 10800 секунд. Если *максимальная продолжительность сессии* (страница 91) превышает данное значение, у пользователя может быть запрошен повторный вход, как только срок действия cookie истечет, даже при активной SSO-сессии.

Для настройки срока действия cookie добавьте в блок `blitz.prod.local.idp.login` файла конфигурации `blitz.conf` параметр `lstateTtlInSec` со значением, равным или превышающим максимальную продолжительность сессии.

```
"lstateTtlInSec" : 20200
```

2.6.3 Настройки консоли управления

Консоль управления настраивается с помощью файлов `console.conf` и `credentials`. Далее в подразделах описаны возможные настройки.

Вход в консоль через SSO

В консоль управления Blitz Identity Provider можно настроить вход через поставщика идентификации OIDC. В качестве такого поставщика может выступить как текущая установка Blitz Identity Provider, так и отдельная его установка или даже стороннее ПО, если оно совместимо с OIDC.

Поддерживаются следующие режимы входа в консоль управления:

- *стандартный режим* (страница 71) по логину/паролю учетных записей, заведенных в разделе Администраторы;
- режим входа через SSO;
- гибридный режим входа, когда администратор может войти как по логину/паролю в стандартном режиме, так и через SSO.

При использовании режима SSO учетные записи администраторов не требуется заводить в разделе Администраторы.

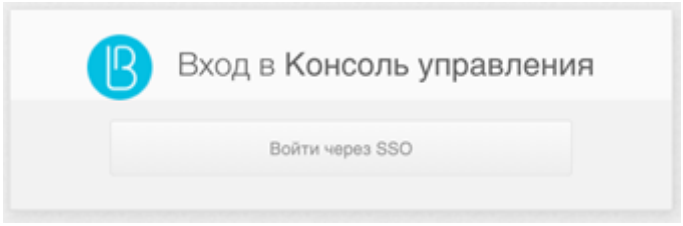
Для настройки режима входа в консоль управления с помощью SSO выполните следующие действия:

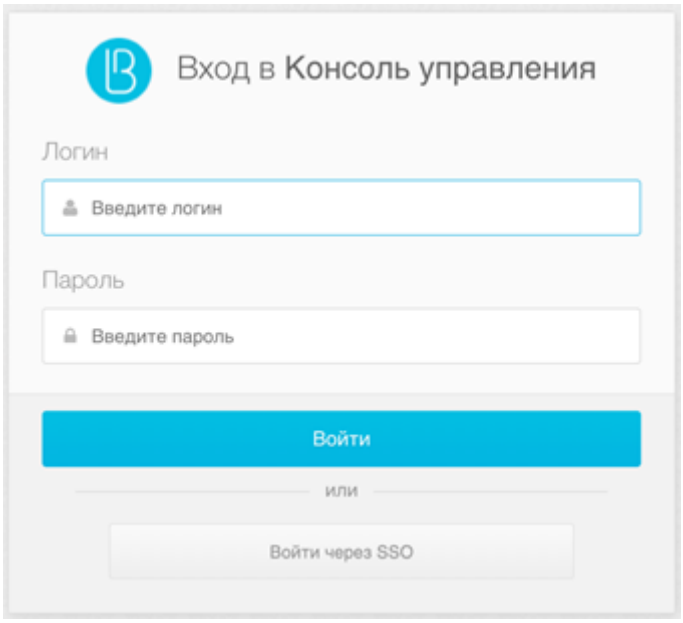
- В настройках поставщика SSO, через которого будет осуществляться вход в консоль, зарегистрируйте консоль управления как приложение. В разрешенных префиксах возврата (`redirect_uri`) укажите домен установки Blitz Identity Provider. По итогам регистрации вы получите `client_id` и `client_secret` приложения для консоли управления;
- в конфигурационном файле `console.conf` создайте блок настроек `login` следующего содержания:

```
{
  "login" : {
    "fp" : {
      "authUri" : "https://idp-host.com/blitz/oauth/ae",
      "clientId" : "blitz-console",
      "clientSecret" : "client_secret_value",
      "logoutUrl" : "https://idp host.com/blitz/login/logout?post_logout_redirect_
↪uri=https://idp host.com/blitz/console",
      "scopes" : [
        "openid"
      ],
      "subjectClaim" : "sub",
      "roleClaim" : "roles",
      "tokenUri" : "https://idp-host.com/blitz/oauth/te"
      },
    "mode" : "sso"
  }
}
```

Необходимо уточнить параметры:

- В параметрах `authUri` и `tokenUri` нужно указать адреса `Authorization Endpoint` и `Token Endpoint` обработчиков внешнего поставщика идентификации.
- В параметрах `clientId` и `clientSecret` указать значения `client_id` и `client_secret`, присвоенный зарегистрированному во внешнем поставщике идентификации приложению, соответствующему консоли управления.
- В параметре `logoutUrl` прописать ссылку, на которую должен перенаправляться пользователь при выходе из консоли управления, чтобы был произведен единый выход через внешний поставщик идентификации.
- В параметре `scopes` прописать список разрешений, который должны быть запрошены (минимально необходимо только разрешение `openid`).
- В `subjectClaim` указать имя атрибута из маркера идентификации (`id_token`), используемого в качестве идентификатора учетной записи. Именно с таким идентификатором будет осуществлен вход администратора при режиме входа `sso`.
- В `roleClaim` указать имя атрибута из маркера идентификации (`id_token`), в котором передается роль (строка) или список ролей (массив строк) администратора. Именно с такими идентификаторами ролей будет осуществлен вход администратора при режиме входа `sso`.
- В параметре `mode` нужно указать требуемый режим страницы входа: `sso` – вход только через внешний поставщик идентификации (см. рисунок ниже); `internal` – вход только по логину и паролю из настроек консоли управления. Если параметр не задан, то доступны оба варианта на выбор пользователя. При входе в режиме Войти через SSO не требуется предварительно создавать администратору учетные записи в меню Администраторы.





Чтобы не показывался промежуточный экран входа, в котором пользователь нажимает кнопку Войти через SSO, можно вызывать консоль управления с помощью ссылки следующего вида: `https://hostname:port/blitz/console?mode=SSO`.

Ограничение сессий

По политике безопасности может требоваться, чтобы пользователь или администратор одновременно не мог быть залогинен с нескольких устройств. Для удовлетворения такой политики безопасности при доступе администратора в консоль управления необходимо в конфигурационном файле `console.conf` добавить блок `session`:

```
"session" : {
  "mode" : "exclusive",
  "check-interval" : 10
}
```

При наличии такой настройки в случае, если будет зафиксирован вход администратора с учетной записью, которой уже выполнен вход, то в прежнем входе при любом действии в консоли управления будет отображена страница входа. Настройка `check-interval` (задается в секундах) указывает в секундах период, как быстро в прежней сессии произойдет выход при появлении новой сессии.

Если по политике безопасности требуется также запретить наличие нескольких сессий для обычных пользователей, то такой режим можно включить избирательно для определенных пользователей при входе в определенные приложения. Это выполняется с помощью настройки *процедуры входа* (страница 263).

Дополнительно в веб-приложении Личный кабинет нужно включить настройку, согласно которой будет происходить досрочный выход из веб-приложения в случае, если учетная запись пользователя заблокирована или была нарушена политика, запрещающая множественный вход пользователя. В конфигурационный файл `blitz.conf` в блок настроек `blitz.prod.local.idp.user-profile` нужно добавить

настройку `check-session-interval`, задающую период проверки веб-приложением активности сессии:

```
"user-profile" : {
  "check-session-interval" : 10,
  …
}
```

Роли и права доступа в консоль

Стандартные роли администраторов описаны в *ранее* (страница 70). В конфигурационном файле `credentials` можно создать дополнительные роли администраторов или исправить права доступа в существующих ролях. Для этого в блоке `roles` нужно скорректировать состав прав доступа (`privileges`), соответствующих роли (`name`).

Пример настройки:

```
"roles" : [
  {
    "name" : "new-role",
    "privileges" : ["w_app","w_system","w_ui","w_user","w_admin","r_audit"]
  }
]
```

В случае создания новых ролей для них также нужно определить *текстовые строки с названием ролей* (страница 301). Пример текстовой строки для новой роли `new_role`:

```
page.admins.role.new-role=имя новой роли
```

Список доступных прав доступа для заполнения настройки `privileges` приведен в таблице ниже.

Права доступа консоли управления Blitz Identity Provider

Право доступа	Доступные разделы консоли управления
`w_app`	Приложения
`w_system`	Источники данных, Аутентификация, Процедуры входа, Поставщики идентификации, SAML, OAuth 2.0, Устройства, Сообщения
`w_ui`	Сервисы самообслуживания, Внешний вид
`w_admin`	Администраторы, События
`w_user`	Пользователи, Группы, Права доступа
`r_user`	Пользователи (только просмотр), Группы (только просмотр), Права доступа (только просмотр)
`r_audit`	События (только просмотр)

Смена пароля администратора

Для изменения пароля администратора консоли выполните следующие действия:

1. Откройте файл `/usr/share/identityblitz/blitz-config/console.conf`.
2. Укажите новый пароль в параметре `pswdHash`. Пароль указывается в открытом виде без шифрования. После применения изменений система его зашифрует.

```
"users" : [
    {
        "pswdHash" : "new$password",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        "roles" : [
            "root"
        ],
        "username" : "admin"
    }
]
```

3. Перезапустите сервис `blitz-console`.

```
sudo systemctl restart blitz-console
```

2.6.4 Настройка Token Exchange

Blitz Identity Provider поддерживает технологию обмена маркеров доступа OAuth 2.0 Token Exchange[80]. Стандартным применением данной технологии в Blitz Identity Provider является взаимодействие *шлюза безопасности Blitz Keeper* (страница 546) с сервисом авторизации для получения доступа к защищаемым сервисам.

Для настройки Token Exchange выполните описанную ниже последовательность действий.

Шаг 1. Создание правила доступа к сервисам

Правила доступа по Token Exchange к защищаемым сервисам создаются в директории `/usr/share/identityblitz/blitz-config/token_exchange/rules/`. Каждое правило создается как отдельный текстовый файл без расширения.

Пример файла с правилом доступа (тип `specialize`):

```
{
    "name": "rule-name",
    "type": "specialize",
    "desc": "",
    "subjectTokenCond": {
        "clientRights": [],
        "userRights": [],
        "scopes": ["openid"],
        "userClaims": {},
        "userGroups": []
    },
    "issue": {
        "ttlInSec": 3600,
        "allowedScopes": ["openid","profile"],
        "allowedClaims": ["sub","global_role","org_id","rights"],
        "addingScopes": [],
        "addingClaims": []
    }
}
```

Пример файла с правилом доступа (тип `impersonate`):

```
{
    "name": "rule-name",
    "type": "impersonate",
    "desc": "",
    "subjectTokenCond": {
```

(continues on next page)

[80] https://tools.ietf.org/html/rfc8693

(продолжение с предыдущей страницы)

```
        "clientRights": [],
        "userRights": [],
        "scopes": ["openid"],
        "userClaims": {},
        "userGroups": []
    },
    "authClientCond": {
        "requiredRights":[
            {
                "rights": ["right1"],
                "target": {
                    "type": "its",
                    "name": "app1"
                }
            }
    },
    "issue": {
        "ttlInSec": 3600,
        "allowedScopes": ["openid","profile"],
        "allowedClaims": ["sub","global_role","org_id","rights"],
        "addingScopes": [],
        "addingClaims": []
    }
}
```

Нужно заполнить следующие атрибуты правила доступа:

- `name` – имя правила, которое должно совпадать с именем файла с правилом доступа;
- `type` – тип правила. Поддерживаются следующие типы правил:
  - `specialize` – по такому правилу приложение запрашивает обмен маркера доступа, выданного этому же приложению. Обмен выполняется с целью специализации маркера доступа – замены в нем разрешений (`scope`), атрибутов (`claims`) и списка получателей (`audience`, `aud`), формат маркера (`jwt` или `opaque`);
  - `impersonate` – по такому правилу приложение запрашивает обмен маркера доступа, выданного другому приложению. Обмен выполняется при условии, что в предоставленном на обмен маркере доступа запрашивающее обмен приложение присутствует в списке получателей (audience, aud). Обмен используется в сценариях, когда приложение А получило исходно маркер доступа, подготовило его для передачи приложению Б (через обмен по типу правила `specialize`), передало приложению Б, так, что приложение Б выпустило на основе полученного маркера доступа свой собственный (через обмен по типу правила `impersonate`).
- `desc` – описание правила. Можно ввести любую текстовую информацию;
- `subjectTokenCond` – условия выполнения правила. Если все указанные в правиле условия будут выполняться, то правило считается выполненным. Если хотя бы одно из условий в правиле не будет выполнено, то все правило считается невыполненным. Условия выполнения правил могут быть следующие:
  - `clientRights` – проверка наличия у приложения указанных прав доступа;

    Пример правила:

```
"clientRights": [
{
    "rights": ["right1"],
    "target": {
    "type": "its",
    "name": "app1"
    }
```

(continues on next page)

(продолжение с предыдущей страницы)

```
}
]
```

В указанном примере проверяется наличие у вызывающего приложения права доступа `right1` в отношении другого приложения (`app1`). Параметр `its` в настройке `target` указывает тип объекта, в отношении которого проверяется наличие права доступа. Возможные значения: `its` – право на приложение; `grps` – право на группу доступа; отсутствие `type` – право на учетную запись пользователя.

– `userRights` – проверка наличия у пользователя указанных прав доступа.

Пример 1 правила:

```
"userRights": [
{
    "rights": ["right2"],
    "target": {
    "type": "grps",
    "name": "org1",
    "ext": "orgs"
    }
}
]
```

В указанном примере проверяется наличие у пользователя права доступа `right2` в отношении группы пользователей (`org1`). В случае типа объекта группы доступа указывается дополнительный параметр `ext`, определяющий профиль группы доступа (см. *Включение отображения групп в blitz.conf* (страница 213)).

Пример 2 правила:

```
"userRights": [
{
    "rights": ["security_administrator"],
    "target": {
    "type": "grps",
    "name": "${org_id}",
    "ext": "orgs"
    }
}
]
```

В указанном правиле проверяется наличие у пользователя права доступа `security_administrator` в отношении группы пользователей из профиля `orgs`, имеющей идентификатор, совпадающий со значением атрибута `org_id` из состава исходного маркера доступа. В отличии от примера 1 в данном примере иллюстрируется возможность в качестве имени объекта права доступа указывать не конкретное значение объекта, а ссылаться на объект на основе значений из присланного маркера доступа (`$org_id`).

Пример 3 правила:

```
"userRights": [
{
    "rights": ["right3"],
    "target": {
    "type": "its",
    "name": "app1"
    }
}
]
```

В данном примере проверяется наличие у пользователя права доступа `right3` в отношении приложения `app1`.

– `scopes` – проверка присутствия в маркере доступа требуемых разрешений (см. *Общие настройки OAuth 2.0* (страница 242));

Пример правила:

```
"scopes": ["scope1"]
```

В данном примере проверяется наличие в исходном маркере доступа разрешения с именем `scope1`.

– `userClaims` – проверка, что у учетной записи пользователя атрибуты имеют указанные значения.

Пример правила:

```
"userClaims": {"role":"FIN"}
```

В данном примере проверяется наличие у пользователя в учетной записи атрибута `role` с заполненным значением `FIN`. Допустимо использовать только атрибуты с типом `String`.

– `userGroups` – проверка, что учетная запись пользователя входит в указанные группы доступа.

Пример правила:

```
"userGroups": [
{
    "name": "admin",
    "profile": "roles"
}
]
```

В данном примере проверяется, что пользователь входит в группу доступа `admin` с профилем `roles`.

- `authClientCond` – условия замены `client_id`. Эти условия проверяются только для правил с типом `impersonate`. В правиле проверяется, что новое приложение имеет права доступа для обмена маркера доступа. Поддерживается условие `requiredRights`.

  Пример правила:

```
"requiredRights":[
  {
    "rights": ["right1"],
    "target": {
      "type": "its",
      "name": "app1"
    }
  }
]
```

  В указанном примере проверяется наличие у вызывающего приложения права доступа `right1` в отношении другого приложения (`app1`). Параметр `its` в настройке `target` указывает тип объекта, в отношении которого проверяется наличие права доступа. Возможные значения: `its` – право на приложение; `grps` – право на группу доступа; отсутствие `type` – право на учетную запись пользователя.

- `issue` – правила выпуска нового маркера доступа, применяемые в случае, если правило было успешно выполнено. Правила выпуска нового маркера доступа состоят из:

  – `ttlInSec` – время жизни (в секундах) выпускаемого маркера доступа;

- `allowedScopes` – разрешения, которые можно оставить в выпускаемом маркере доступа;
- `allowedClaims` – атрибуты пользователя, которые можно оставить в выпускаемом маркере доступа;
- `addingScopes` – добавляемые в маркер доступа разрешения;
- `addingClaims` – добавляемые в маркер доступа атрибуты пользователя.

Шаг 2. Настройка обмена маркеров доступа

Чтобы определить, как будет происходить обмен маркеров доступа по Token Exchange, а именно для каких защищаемых сервисов какие правила доступа будут применяться, необходимо в конфигурационном файле `blitz.conf` добавить блок настроек `blitz.prod.local.idp.token-exchange` следующего вида:

```
"token-exchange" : {
  "resources" : [
    {
      "uri" : "http://secured_service_host/api/service1",
      "methods" : ["GET","POST"],
      "rules" : [
        "rule1",
        "rule2"
      ]
    },
    {
      "audience" : "secured-api",
      "rules" : [
        "rule3"
      ]
    },
    …
  ]
}
```

В блоке `resources` нужно для каждого сервиса заполнить настройки:

- `rules` – перечислить имена правил доступа к сервису. Каждому правилу соответствует свой *файл настроек* (страница 367). Доступ к сервису разрешается, если хотя бы одно из правил из этого списка будет выполненным. Если все перечисленные правила не будут выполнены, то тогда доступ к сервису будет запрещен;
- `uri` – необязательный параметр, может задавать адрес защищаемого сервиса. В задании адреса сервиса допустимо использовать звездочку (`*`) для пропуска одного компонента пути адреса и двойную звездочку (`**`) для пропуска оставшейся части пути адреса сервиса;
- `methods` – необязательный параметр, указывает перечень HTTP-методов вызываемого сервиса;
- `audience` – необязательный параметр, может задавать имя приложения. Данное значение будет включено в выпущенный новый маркер доступа в атрибут `aud`. Обязательно должен быть указан один из параметров `uri` или `audience`.

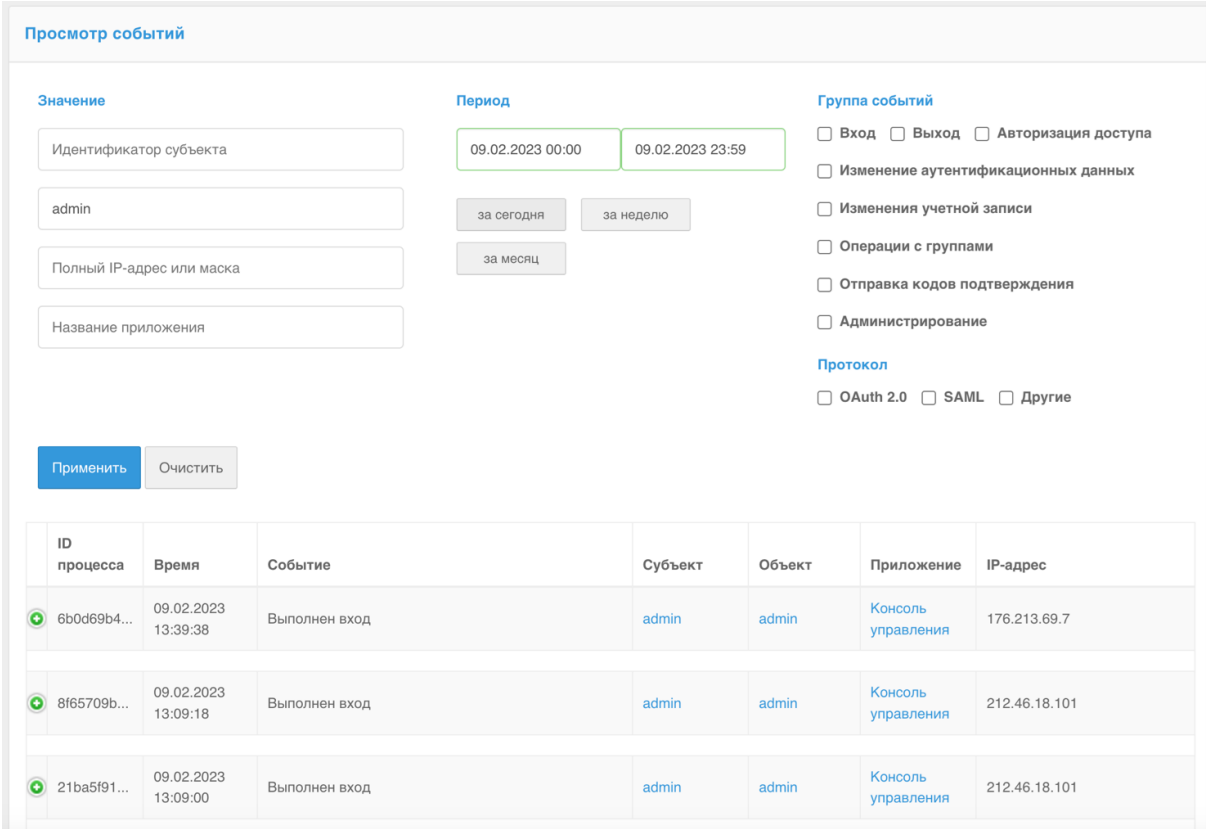
2.7 Безопасность, обслуживание и устранение неисправностей

2.7.1 Просмотр событий безопасности

Для ведения аудита безопасности и для просмотра зарегистрированных в журнале Blitz Identity Provider событий безопасности используется раздел События консоли управления. Здесь имеется возможность осуществлять фильтрацию событий безопасности по различным критериям:

- по пользователю (указание идентификатора пользователя обязательно);
- по диапазону дат;
- по конкретному приложению;
- по группам событий;
- по IP-адресам;
- по протоколам взаимодействия.

После настройки фильтров и их применения предусмотрен просмотр детальной информации о найденных событиях.



2.7.2 Мониторинг функционирования приложений

Стандартный сервис мониторинга

Для мониторинга доступности приложений Blitz Identity Provider предусмотрен сервис `/blitz/metrics`, вызываемый с помощью `HTTP GET`. Рекомендуется, чтобы сервис был доступен на каждом сервере приложения по `HTTP` при вызове из внутренней сети с серверов мониторинга и вместе с тем, чтобы сервис был недоступен при вызове из внешних сетей и с рабочих мест пользователей.

В случае если приложение доступно, то сервис `/blitz/metrics` вернет детальную информацию о метриках функционирования приложения в формате Prometheus[81].

Пример ответа сервиса

```
# HELP blitz_idp_uptime_seconds Uptime
# TYPE blitz_idp_uptime_seconds gauge
blitz_idp_uptime_seconds{blitz_host="papp01.loc",} 63859.0
# HELP blitz_idp_licence_exp_seconds Licence expiration
# TYPE blitz_idp_licence_exp_seconds gauge
blitz_idp_licence_exp_seconds{blitz_host="papp01.loc",} 9.2233720368547776E18
# HELP blitz_idp_config_mtime Last time, a file was changed
# TYPE blitz_idp_config_mtime gauge
# HELP blitz_idp_datasource_latency Latency of an datasource operation
# TYPE blitz_idp_datasource_latency histogram
blitz_idp_datasource_latency_bucket{blitz_host="papp01.loc",ds_type="ldap",ds_name=
↪"389-ds",op_type="read",le="0.005",} 13.0
…
blitz_idp_datasource_latency_bucket{blitz_host="papp01.loc",ds_type="ldap",ds_name=
↪"389-ds",op_type="read",le="+Inf",} 29.0
blitz_idp_datasource_latency_count{blitz_host="papp01.loc",ds_type="ldap",ds_name=
↪"389-ds",op_type="read",} 29.0
blitz_idp_datasource_latency_sum{blitz_host="papp01.loc",ds_type="ldap",ds_name=
↪"389-ds",op_type="read",} 0.3112787189999999
# HELP blitz_idp_mq_connections Amount connections to datasource
# TYPE blitz_idp_mq_connections gauge
blitz_idp_mq_connections{blitz_host="papp01.loc",mq_type="rmq",mq_server="pmq01.
↪loc_5672",} 1.0
# HELP blitz_idp_mq_latency Latency of an mq operation
# TYPE blitz_idp_mq_latency histogram
blitz_idp_mq_latency_bucket{blitz_host="papp01.loc",mq_type="rmq",mq_server="pmq01.
↪loc_5672",broker="blitz.events.direct",op_type="write",le="0.005",} 1.0
…
blitz_idp_mq_latency_bucket{blitz_host="papp01.loc",mq_type="rmq",mq_server="pmq01.
↪loc_5672",broker="blitz.events.direct",op_type="write",le="+Inf",} 3.0
blitz_idp_mq_latency_count{blitz_host="papp01.loc",mq_type="rmq",mq_server="pmq01.
↪loc_5672",broker="blitz.events.direct",op_type="write",} 3.0
blitz_idp_mq_latency_sum{blitz_host="papp01.loc",mq_type="rmq",mq_server="pmq01.
↪loc_5672",broker="blitz.events.direct",op_type="write",} 0.028808135999999998
# HELP blitz_idp_authn_method_app_total Amount of method authentications by app id
# TYPE blitz_idp_authn_method_app_total counter
blitz_idp_authn_method_app_total{blitz_host="papp01.loc",app_id="_blitz_profile",
↪method="sms",status="success",} 2.0
blitz_idp_authn_method_app_total{blitz_host="papp01.loc",app_id="_blitz_profile",
↪method="cls",status="other_error",} 7.0
blitz_idp_authn_method_app_total{blitz_host="papp01.loc",app_id="_blitz_profile",
↪method="password",status="success",} 4.0
blitz_idp_authn_method_app_total{blitz_host="papp01.loc",app_id="_blitz_profile",
↪method="knownDevice",status="other_error",} 3.0
```

(continues on next page)

[81] https://prometheus.io/

(продолжение с предыдущей страницы)

```
# HELP blitz_idp_authn_method_total Amount of authentications by a method
# TYPE blitz_idp_authn_method_total counter
blitz_idp_authn_method_total{blitz_host="papp01.loc",method="password",status=
↪"success",} 4.0
blitz_idp_authn_method_total{blitz_host="papp01.loc",method="knownDevice",status=
↪"other_error",} 3.0
blitz_idp_authn_method_total{blitz_host="papp01.loc",method="cls",status="other_
↪error",} 7.0
blitz_idp_authn_method_total{blitz_host="papp01.loc",method="sms",status="success",
↪} 2.0
# HELP blitz_idp_authn_method_latency Latency of an authentication method
# TYPE blitz_idp_authn_method_latency histogram
blitz_idp_authn_method_latency_bucket{blitz_host="papp01.loc",method="sms",le="1.0
↪",} 0.0
…
blitz_idp_authn_method_latency_bucket{blitz_host="papp01.loc",method="sms",le="+Inf
↪",} 2.0
blitz_idp_authn_method_latency_count{blitz_host="papp01.loc",method="sms",} 2.0
blitz_idp_authn_method_latency_sum{blitz_host="papp01.loc",method="sms",} 28.
↪686999999999998
blitz_idp_authn_method_latency_bucket{blitz_host="papp01.loc",method="password",le=
↪"1.0",} 0.0
…
blitz_idp_authn_method_latency_bucket{blitz_host="papp01.loc",method="password",le=
↪"+Inf",} 4.0
blitz_idp_authn_method_latency_count{blitz_host="papp01.loc",method="password",} 4.
↪0
blitz_idp_authn_method_latency_sum{blitz_host="papp01.loc",method="password",}␣
↪1835.901
# HELP blitz_idp_datasource_connections Amount connections to datasource
# TYPE blitz_idp_datasource_connections gauge
blitz_idp_datasource_connections{blitz_host="papp01.loc",ds_type="ldap",ds_name=
↪"389-ds",} 10.0
# HELP blitz_idp_version Application version
# TYPE blitz_idp_version gauge
blitz_idp_version{blitz_host="papp01.loc",part="major",} 5.0
blitz_idp_version{blitz_host="papp01.loc",part="minor",} 16.0
blitz_idp_version{blitz_host="papp01.loc",part="patch",} 1.0
# HELP blitz_idp_notify_user_total Amount of user notifications by channel
# TYPE blitz_idp_notify_user_total counter
blitz_idp_notify_user_total{blitz_host="papp01.loc",channel="email",} 3.0
blitz_idp_notify_user_total{blitz_host="papp01.loc",channel="sms",} 4.0
blitz_idp_notify_user_total{blitz_host="papp01.loc",channel="push",} 2.0
```

Имя каждой метрики начинается с имени приложения (дефис в имени заменен на подчеркивание): `blitz_idp_%%%`, `blitz_registration_%%%`, `blitz_recovery_%%%`, `blitz_console_%%%`. Список доступных метрик приведен в таблице.

Метрики функционирования Blitz Identity Provider

Право доступа	Тип	Пояснение
uptime_seconds	gauge	Время с момента запуска приложения (в секундах)
licence_exp_secc	gauge	Время до истечения срока действия лицензии (в секундах)
config_mtime	gauge	Время последнего изменения файла конфигурации
datasource_laten	histogr	Задержки ответа от хранилища УЗ по операциям чтения и записи (могут быть типы `ldap`, `jdbc`, `couch`)
mq_connections	gauge	Количество коннектов к MQ (`rmq`, `kafka`)
mq_latency	histogr	Задержки ответа от MQ (`rmq`, `kafka`)
authn_method_a	counte	Количество успешных и неуспешных аутентификаций каждым методом входа в различные приложения
authn_method_t	counte	Общее кол-во успешных и нет аутентификаций разными методами
authn_method_l	histogr	Длительность аутентификации по разным методам входа
datasource_conn	gauge	Кол-во коннектов к хранилищам
version	gauge	Версия приложения
notify_user_total	counte	Кол-во направленных сообщений по разным каналам
authn_method_app_created	gauge	Эти метрики (с суффиксом `_created`) генерируются в связи с особенностями Prometheus и содержат время в `unix timestamp` момента создания метрики
authn_method_created	gauge	
authn_method_latency_created	gauge	
datasource_latency_created	gauge	
mq_latency_created	gauge	
notify_user_created	gauge	

Использование Grafana и Prometheus

Для быстрой настройки мониторинга и визуализации процессов Blitz Identity Provider удобно использовать job-задание Prometheus и шаблон дашборда Grafana, входящие в поставку (`resources.zip`).

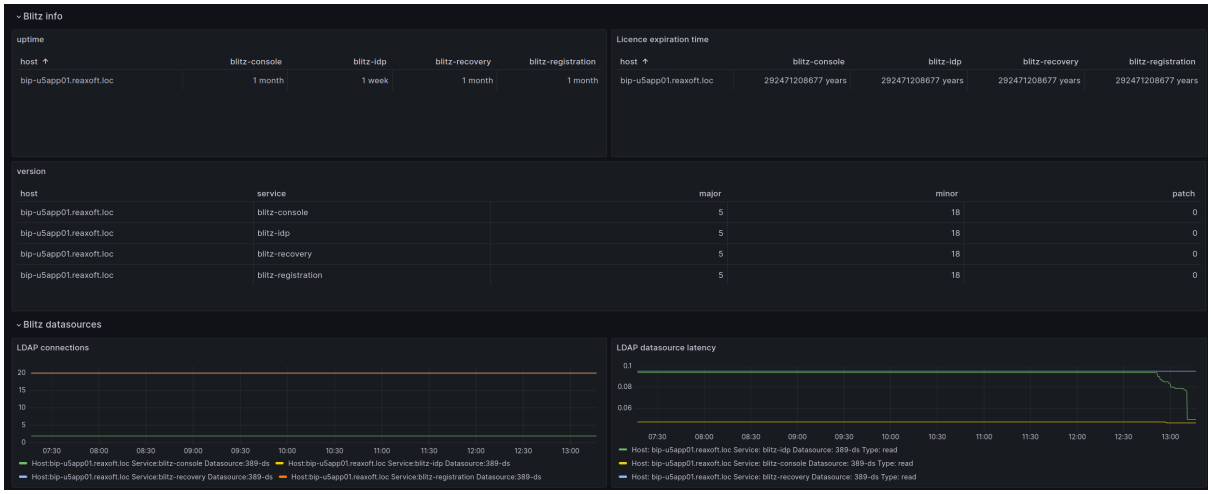
Совет: Визуальное представление данных имеет широкий спектр применения. Оно может быть использовано менеджерами для анализа рабочих процессов, инженерами для отслеживания ситуаций, когда количество аутентификаций превышает пороговое значение (настраиваются оповещения), для контроля за сроком действия лицензии и др. При обновлении удобно отслеживать версии сервисов на большом количестве хостов и время их запуска.

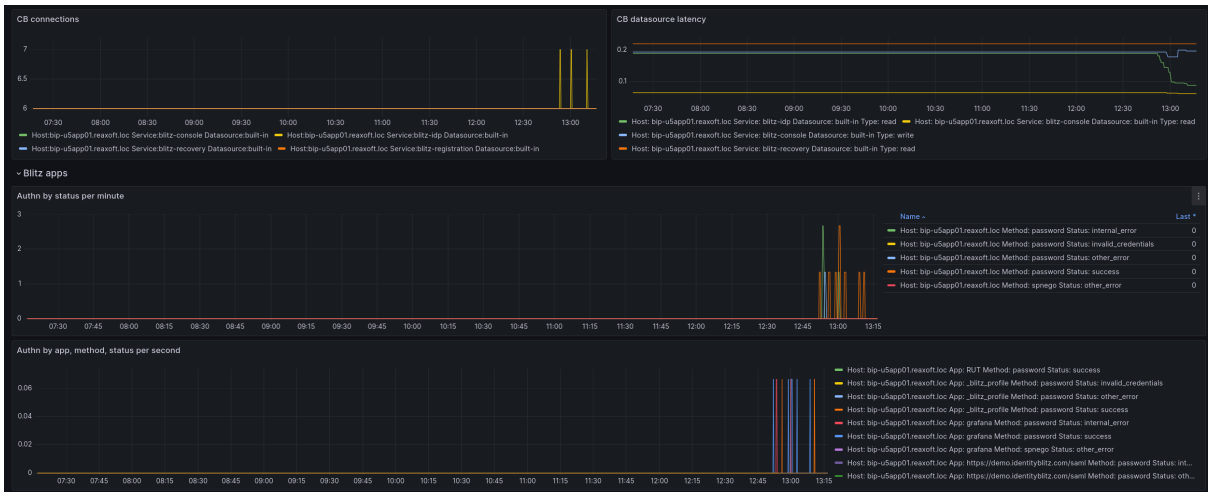
Для настройки визуализации выполните следующие действия:

1. Модифицируйте job-задание `prometheus.yaml` в соответствии с конфигурацией своей системы и добавьте его в Prometheus[82].
2. Модифицируйте шаблон дашборда `blitz-dashboard.json`. Настройте Grafana и добавьте дашборд[83].

Примеры визуализации данных в Grafana:

[82] https://prometheus.io
[83] https://prometheus.io/docs/visualization/grafana/





2.7.3 Решение проблем

Логи работы Blitz Identity Provider записываются в директорию `/var/log/identityblitz` на каждом сервере. Журнал событий каждого приложения называется в соответствии с приложением:

- `blitz-console.log` – журнал событий консоли управления;
- `blitz-idp.log` – журнал событий сервиса аутентификации;
- `blitz-registration.log` – журнал событий сервиса регистрации;
- `blitz-recovery.log` – журнал событий сервиса восстановления доступа;
- `blitz-keeper.log` – журнал событий шлюза безопасности.

При возникновении ошибок, связанных с работой Blitz Identity Provider (записываются в лог как `[ERROR]`), рекомендуется обратиться в техническую поддержку Blitz Identity Provider по адресу support@idblitz.ru. При обращении указать используемую версию Blitz Identity Provider.

При необходимости повысить уровень логирования необходимо в конфигурационном файле `blitz.conf` в блоке `logger` изменить уровни логирования.

По умолчанию выставлены следующие уровни логирования:

```
"levels" : {
    "ROOT" : "TRACE",
    "application" : "TRACE",
    "com.couchbase.client" : "INFO",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "com.couchbase.service" : "INFO",
    "com.couchbase.endpoint" : "INFO",
    "com.couchbase.node": "INFO",
    "com.couchbase.tracing": "INFO",
    "com.identityblitz" : "TRACE",
    "com.identityblitz.idp" : "TRACE",
    "com.identityblitz.idp.events" : "TRACE",
    "com.identityblitz.idp.flow.dynamic" : "TRACE",
    "com.identityblitz.idp.flow.dynamic.extend" : "TRACE",
    "com.identityblitz.idp.task.processing" : "DEBUG",
    "com.identityblitz.login-framework" : "TRACE",
    "com.identityblitz.login-framework.ldap-timings" : "INFO",
    "com.identityblitz.login.store" : "TRACE",
    "com.identityblitz.idp.rabbitmq" : "INFO",
    "com.identityblitz.play.memcached" : "INFO",
    "com.identityblitz.play.memcached.RefreshableMemcachedConnection" : "INFO",
    "com.unboundid.ldap.sdk" : "TRACE",
    "org.asynchttpclient.netty" : "TRACE",
    "org.opensaml" : "INFO",
    "org.opensaml.util.resource" : "INFO",
    "play" : "TRACE",
    "plugin.memcached" : "INFO"
}
```

Для повышения уровня логирования необходимо параметрам `ROOT` и всем `com.identityblitz.*` присвоить значение `TRACE`.

В случае если случайно было произведено изменение конфигурации Blitz Identity Provider в консоли управления, то в скрытой директории `/usr/share/identityblitz/blitz-config/.snapshot` сохранились предыдущие версии конфигурационных файлов `blitz.conf` и `console.conf`. Можно использовать эти файлы для отката к предыдущей конфигурации или для определения отличий с текущими конфигурационными файлами.

Чтобы узнать, в какое время и кем был изменен конфигурационный файл, в начало конфигурационных файлов `blitz.conf` и `console.conf` помещаются комментарии с указанием времени редактирования и автора изменений. Пример записи аудита изменения конфигурационного файла приведен ниже:

```
################################################################################
↪#######
# modified: 2021-05-09 20:55:55 MSK
# author: admin
# ip: 0:0:0:0:0:0:0:1
# user agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
################################################################################
↪#######
```

2.7.4 Шлюз безопасности

В качестве шлюза безопасности используется отдельно устанавливаемый модуль *Blitz Keeper* (страница 546).

2.8 Рекомендации ФСТЭК

Ниже приведены рекомендации по обеспечению защиты информации согласно требованиям ФСТЭК.

2.8.1 Идентификация и аутентификация субъектов и объектов доступа (ИАФ)

ИАФ.1

Идентификация и аутентификация пользователей, являющихся работниками оператора

Рекомендация по настройке:

- *Настроить методы аутентификации пользователей* (страница 89).
- Для администраторов консоли управления *настроить вход через Blitz* (страница 363).
- Настроить через процедуры входа для администраторов и пользователей *требования к прохождению двухфакторной аутентификации* (страница 263).
- *Задать* (страница 224) для подключаемых приложений `client_id` и `client_secret`.

ИАФ.3

Управление идентификаторами, в том числе создание, присвоение, уничтожение идентификаторов

Рекомендация по настройке:

- *Задать атрибут* (страница 76), который будет использоваться в качестве идентификатора учетной записи.
- Настроить запрет на *повторное использование идентификатора после удаление учетной записи* (страница 334) и *блокирование неактивных учетных записей* (страница 333).

ИАФ.4

Управление средствами аутентификации, в том числе хранение, выдача, инициализация, блокирование средств аутентификации и принятие мер в случае утраты и (или) компрометации средств аутентификации

Рекомендация по настройке:

- *Установить* (страница 311) парольную политику.
- *Управлять* (страница 201) учетными записями пользователей.

ИАФ.5

Защита обратной связи при вводе аутентификационной информации

Специальная настройка не требуется.

ИАФ.6

Идентификация и аутентификация пользователей, не являющихся работниками оператора (внешних пользователей)

Рекомендация по настройке:

- Настроить вход через внешние поставщики идентификации: *ЕСИА* (страница 154), *СУДИР* (страница 172), *другая установка продукта* (страница 184).

2.8.2 Управление доступом субъектов к объектам доступа

УПД.1

Управление (заведение, активация, блокирование и уничтожение) учетными записями пользователей, в том числе внешних пользователей

Рекомендация по настройке:

- При необходимости разделения учетных записей на внешние и внутренние *завести* (страница 72) атрибут с типом учетной записи и *настроить* (страница 263) политику доступа пользователей в приложения.
- Для временных учетных записей *настроить* (страница 365) правила блокирования входа по истечение срока действия записей.
- Для использования функций работы с группами пользователей *настроить* (страница 213) группы пользователей.
- *Управлять* (страница 201) учетными записями через консоль управления.

УПД.2

Реализация необходимых методов (дискреционный, мандатный, ролевой или иной метод), типов (чтение, запись, выполнение или иной тип) и правил разграничения доступа

Рекомендация по настройке:

- Настроить *права доступа* (страница 311), *разрешения* (страница 242).
- Установить *разрешения приложений* (страница 238).
- *Настроить* (страница 546) шлюз безопасности.
- Настроить *атрибуты пользователей* (страница 72) и *группы пользователей* (страница 213), *процедуры входа в приложения* (страница 263).

УПД.4

Разделение полномочий (ролей) пользователей, администраторов и лиц, обеспечивающих функционирование информационной системы

Рекомендация по настройке:

- *Использовать* (страница 431) сервисы изменения атрибутов, включения и исключения пользователей в группы, назначения и отзыва прав доступа.

УПД.5

Назначение минимально необходимых прав и привилегий пользователям, администраторам и лицам, обеспечивающим функционирование информационной системы

Рекомендация по настройке:

- *Назначать* (страница 70) роли администраторов.
- *Управлять* (страница 201) атрибутами пользователей.
- *Назначать* (страница 495) права доступа с использованием сервисов.

УПД.6

Ограничение неуспешных попыток входа в информационную систему (доступа к информационной системе)

Рекомендация по настройке:

- *Настроить* (страница 99) политику ограничения числа попыток входа с последующим блокированием учетной записи.

УПД.7

Предупреждение пользователя при его входе в информационную систему о том, что в информационной системе реализованы меры защиты информации, и о необходимости соблюдения им установленных оператором правил обработки информации

Рекомендация по настройке:

- Настроить для приложений экран согласия пользователя: см. *Настройка OAuth 2.0 и OpenID Connect 1.0* (страница 238), *Общие настройки OAuth 2.0* (страница 242), *Настройки текстов интерфейса* (страница 301).

УПД.8

Оповещение пользователя после успешного входа в информационную систему о его предыдущем входе в информационную систему)

Рекомендация по настройке:

- *Настроить* (страница 197) доступ к аудиту по себе для пользователей.

УПД.9

Ограничение числа параллельных сеансов доступа для каждой учетной записи пользователя информационной системы

Рекомендация по настройке:

- *Настроить* (страница 365) политику ограничения числа параллельных сеансов.

УПД.10

Блокирование сеанса доступа в информационную систему после установленного времени бездействия (неактивности) пользователя или по его запросу

Рекомендация по настройке:

- *Настроить* (страница 89) период неактивности.

2.8.3 Регистрация событий безопасности (РСБ)

РСБ.1

Определение событий безопасности, подлежащих регистрации, и сроков их хранения

Специальная настройка не требуется.

РСБ.2

Определение состава и содержания информации о событиях безопасности, подлежащих регистрации

Специальная настройка не требуется.

РСБ.3

Сбор, запись и хранение информации о событиях безопасности в течение установленного времени хранения

Рекомендация по настройке:

- *Просмотр* (страница 372) событий безопасности периодически осуществлять в консоли управления.

РСБ.4

Реагирование на сбои при регистрации событий безопасности, в том числе аппаратные и программные ошибки, сбои в механизмах сбора информации и достижение предела или переполнения объема (емкости) памяти

Рекомендация по настройке:

- *Просмотр* (страница 376) журналов событий на предмет возникновения ошибок.

РСБ.5

Мониторинг (просмотр, анализ) результатов регистрации событий безопасности и реагирование на них

Рекомендация по настройке:

- *Просмотр* (страница 372) событий безопасности периодически осуществлять в консоли управления.

РСБ.6

Генерирование временных меток и (или) синхронизация системного времени в информационной системе

Рекомендация по настройке:

- При установке ПО сконфигурировать использование сервиса точного времени (NTP).

РСБ.7

Защита информации о событиях безопасности

Рекомендация по настройке:

- *Настроить* (страница 19) резервное копирование СУБД

РСБ.8

Обеспечение возможности просмотра и анализа информации о действиях отдельных пользователей в информационной системе

Рекомендация по настройке:

- *Просмотр* (страница 372) событий безопасности периодически осуществлять в консоли управления.

Глава 3

Интеграция

3.1 Подготовка к интеграции

3.1.1 Выбор протокола взаимодействия

При интеграции приложения с Blitz Identity Provider для проведения идентификации и аутентификации пользователя следует выбрать один из протоколов взаимодействия:

- OpenID Connect 1.0 (OIDC)[84] / OAuth 2.0[85] – современный SSO-протокол, изначально ориентированный на работу с веб-приложениями и мобильными приложениями в сети Интернет.

  Совет: Если создается новое приложение, то рекомендуется подключить его к Blitz Identity Provider с использованием OIDC/OAuth 2.0.

- SAML 1.0/1.1/2.0[86] – SSO-протокол, позволяющий подключить различное корпоративное ПО или облачные приложения к сервису входа.

  Внимание: Подключаемое приложение должно иметь встроенную поддержку SAML или такая поддержка может быть добавлена в качестве дополнительной опции или через установку интеграционного коннектора/плагина.

Выбор протокола во многом зависит от того, какое приложение требуется подключить:

- если приложение поддерживает один из SSO-протоколов, то стоит подключать его с использованием данного протокола;
- если предложение не поддерживает протоколы, то следует провести его доработку – в этом случае рекомендуется поддержать взаимодействие по OIDC;
- если приложение только создается, то на этой стадии целесообразно поддержать один из SSO-протоколов – поддержку OIDC реализовать проще, однако при использовании доступных библиотек SAML можно использовать и этот протокол.

В таблице ниже приведены некоторые особенности протоколов OIDC и SAML.

[84] https://openid.net/specs/openid-connect-core-1_0.html
[85] https://tools.ietf.org/html/rfc6749
[86] https://www.oasis-open.org/standards#samlv2.0

Особенности протоколов подключения

	OIDC/OAuth 2.0	SAML 1.0/1.1/2.0
Способ обеспечения доверия между приложением и Blitz Identity Provider	Секрет приложения (обычно в виде строки), известный Blitz Identity Provider	Электронная подпись. И запросы на аутентификацию, и ответы – это подписанные XML-документы
Способ взаимодействия	Через веб-браузер пользователя проходит аутентификация. Для завершения аутентификации серверная часть приложения должна сформировать HTTP-запрос в адрес Blitz Identity Provider	Обычно запрос на аутентификацию и ответ проходят через веб-браузер пользователя. Приложение и Blitz Identity Provider могут не иметь сетевой связности
Получение сведений о пользователе	Два способа получения данных о пользователе: • Приложение обращается к REST-сервису Blitz Identity Provider и получает данные о пользователе в формате JSON. Приложение может продолжать получать данные о пользователе, даже когда пользователь завершает свою онлайн-сессию • Приложение получает данные пользователя из маркера идентификации (id_token в форме JWT), полученного от Blitz Identity Provider по результатам входа	Данные о пользователе содержатся в ответе на запрос на аутентификацию в формате XML. Приложение может получать от Blitz Identity Provider данные только в момент входа пользователя
Поддерживаемые приложения	Веб-приложения и мобильные приложения	Веб-приложения

Примечание: OIDC позволяет реализовать все основные сценарии SAML, но при этом используется более простой JSON/REST-протокол. Существенное преимущество OIDC – поддержка мобильных приложений.

Важно: Если подключаемое к Blitz Identity Provider приложение доработать невозможно, но при этом приложение представляет собой веб-приложение, развернутое в собственной инфраструктуре (on-premise), то подключить приложение к Blitz Identity Provider можно с использованием веб-прокси и специально реализованного в Blitz Identity Provider *протокола Simple* (страница 250).

3.2 Интеграция приложения по OIDC

3.2.1 Как правильно зарегистрировать приложение

Аутентификация в терминологии OIDC/OAuth 2.0 является результатом взаимодействия трех сторон:

- сервиса авторизации (`Authorization Server`) или поставщика ресурса (`Resource Server`), в качестве которых выступает Blitz Identity Provider;
- системы-клиента (`Client`), в качестве которой выступает приложение, которое запрашивает доступ ресурсу (информации и данным пользователя);
- владельца ресурса (`Resource Owner`), в качестве которого выступает пользователь, так как в ходе аутентификации он разрешает доступ к данным о себе.

Первым шагом при подключении приложения является его *регистрация* (страница 238) в качестве системы-клиента в Blitz Identity Provider. В запросах на проведение аутентификации будут использоваться и учитываться данные, заданные при регистрации приложения:

Веб-приложение

- идентификатор приложения (`client_id`);
- секрет приложения (`client_secret`).
- разрешенные адреса возврата (списки `redirect_uri` и `post_logout_redirect_uri`);
- перечень запрашиваемых разрешений (список `scope`);
- информация о нестандартных режимах, необходимых приложению:
  - приложению необходимо получать `refresh_token` – по умолчанию приложению `refresh_token` возвращаться не будет; при выборе этого режима нужно дополнительно указать требуемый срок действия `refresh_token` (по умолчанию срок действия маркера будет 1 день, максимально можно запросить срок действия 365 дней);
  - приложению необходимо использовать нестандартный сценарий взаимодействия (например, `Implicit Flow`, `Hybrid Flow`) – по умолчанию приложению разрешено использовать только `Authorization Code Flow`;
  - приложению нужно получать маркер доступа в формате `JWT` – по умолчанию маркер доступа предоставляется в формате `opaque`;
  - приложению нужно получать маркер доступа (`access_token`) с нестандартным сроком действия – стандартно маркер доступа действует 1 час;
- перечень дополнительных атрибутов, которые Blitz Identity Provider должен добавить в маркер идентификации (дополнительные атрибуты для передачи в составе `id_token`);
- режим входа (вход как физического лица или как представителя организации).

Мобильное приложение

- идентификатор мобильного приложения (`software_id`);
- первичный маркер доступа (`Initial Access Token`);
- метаданные приложения в форме JWS-токена (`software_statement`).
- разрешенные адреса возврата (списки `redirect_uri` и `post_logout_redirect_uri`);
- перечень запрашиваемых разрешений (список `scope`);
- нестандартные режимы, необходимые приложению:

  - приложению необходимо получать `refresh_token` – по умолчанию приложению `refresh_token` возвращаться не будет; при выборе этого режима нужно дополнительно указать требуемый срок действия `refresh_token` (по умолчанию срок действия маркера будет 1 день, максимально можно запросить срок действия 365 дней);
  - приложению необходимо использовать нестандартный сценарий взаимодействия (например, `Implicit Flow`, `Hybrid Flow`) – по умолчанию приложению разрешено использовать только `Authorization Code Flow`;
  - приложению нужно получать маркер доступа в формате `JWT` – по умолчанию маркер доступа предоставляется в формате `opaque`;
  - приложению нужно получать маркер доступа (`access_token`) с нестандартным сроком действия – стандартно маркер доступа действует 1 час;
- перечень дополнительных атрибутов, которые Blitz Identity Provider должен добавить в маркер идентификации (дополнительные атрибуты для передачи в составе `id_token`);
- режим входа (вход как физического лица или как представителя организации).

Примечание: При разработке мобильного приложения можно использовать как общие `Initial Access Token` и `software_statement` для своих iOS/Android-реализаций, так и запросить получение различных наборов `Initial Access Token` и `software_statement` для каждой ОС и, возможно, каждой редакции (телефон/планшет) и даже версии приложения. Для простоты дальнейшего изложения в тексте документа будет подразумеваться, что мобильное приложение использует один общий `Initial Access Token` и один общий `software_statement`.

При создании в мобильных приложениях функции входа с использованием Blitz Identity Provider рекомендуется учитывать следующие особенности:

- пользователям мобильных приложений неудобно вводить при каждом входе логин и пароль на веб-странице аутентификации Blitz Identity Provider. Вместо этого им привычнее при повторных входах использовать ПИН-код приложения или Touch ID/Face ID;
- пользователь может использовать свою учетную запись Blitz Identity Provider для входа в несколько установок одного и того же мобильного приложения (например, войти в приложение, установленное на iPhone, и войти в это же приложение, установленное на iPad). Пользователь должен иметь возможность отозвать выданные этим установкам приложений права доступа к своим сведениям в Blitz Identity Provider;
- по причинам безопасности нежелательно хранить на устройстве пользователя (внутри сборки мобильного приложения) пароль приложения (`client_secret`), используемый для взаимодействия приложения с Blitz Identity Provider.

Чтобы учесть изложенные выше особенности, в Blitz Identity Provider предусмотрен ряд специальных механизмов, предназначенных для использования мобильными приложениями.

Рекомендуемый сценарий взаимодействия мобильного приложения с Blitz Identity Provider описан в *Подключение мобильного приложения* (страница 408).

Ниже вы найдете информацию о том, как определить, какие разрешенные адреса возврата, разрешения `scope`, дополнительные атрибуты в `id_token` вы можете задать при регистрации приложения в Blitz Identity Provider.

Как определить адреса возврата

Запрос на проведение идентификации/аутентификации пользователя содержит ссылку возврата при авторизации (`redirect_uri`), куда должен быть возвращен пользователь после прохождения идентификации/аутентификации. Допустимые ссылки возврата должны соответствовать зарегистрированным в Blitz Identity Provider разрешенным префиксам.

Если в запросе на идентификацию/аутентификацию указана ссылка возврата, и она не соответствует ни одному из указанных префиксов, то в идентификации/аутентификации будет отказано.

В зависимости от типа подключаемого приложения рекомендуется использовать следующие префиксы ссылок возврата:

- При подключении веб-приложений в качестве префиксов ссылок возврата использовать доменные имена приложений. Например, если после проведения аутентификации требуется вернуть пользователя на `https://domain.com/callback`, то в качестве префикса ссылки возврата следует указать `https://domain.com/`.

  Предупреждение: При подключении к продуктивной среде Blitz Identity Provider веб-приложение должно использовать в качестве `redirect_uri` и `post_logout_redirect_uri` только HTTPS-обработчики. Использование HTTP для взаимодействия с продуктивной средой Blitz Identity Provider запрещено.

- При подключении мобильных приложений в качестве префиксов ссылок возврата рекомендуется указать сами ссылки возврата одного из типов: ссылки типа `private-use URI scheme`» (например, `com.example.app:/oauth2redirect/example-provider`) или ссылки типа `Universal links` (например, `https://app.example.com/oauth2redirect/example-provider`).

  Примечание: Ссылки типа `Universal links` доступны начиная с iOS 9 и Android 6.0 и являются предпочтительными для использования. Ссылки `private-use URI scheme` рекомендуется использовать только в случае, если приложение должно работать на более ранних версиях iOS/Android.

Запрос на проведение логаута содержит ссылку возврата при логауте (`post_logout_redirect_uri`). Эта ссылка указывает, куда должен быть возвращен пользователь после успешно выполненного логаута. Допустимые ссылки возврата должны соответствовать зарегистрированным в Blitz Identity Provider разрешенным префиксам (префикс должен содержать доменное имя приложения и часть пути, минимум, `https://domain.com/`). Если в запросе на логаут указана ссылка возврата, и она не соответствует ни одному из указанных префиксов, то будет отображена ошибка.

Какие разрешения можно запросить

Разрешения (`scope` в терминологии OIDC/OAuth 2.0) определяют, какие данные и какие именно права на выполнение каких операций получит приложение по результатам аутентификации.

Перечень предусмотренных в Blitz Identity Provider разрешений представлен в таблице.

Доступные разрешения (scope)

Разрешение	Описание	Состав получаемых атрибутов
`openid`	Техническое разрешение, указывающее на то, что аутентификация проводится согласно спецификации OIDC	При запросе этого `scope` Blitz Identity Provider предоставляет приложению `id_token`. Из `id_token` приложение может получить нужные ему *атрибуты пользователя* (страница 400).
`profile`	Основные данные профиля пользователя	Список данных: • `sub`– уникальный идентификатор • `family_name` – фамилия • `given_name` – имя • `middle_name` – отчество • `email` – служебный адрес электронной почты • `phone_number` – номер мобильного телефона
`usr_grps`	Получение списка групп пользователя	`groups` – список групп, в которые включен пользователь. Каждая запись в списке включает следующие атрибуты организации: • `id` – идентификатор группы • `name` – имя группы
`native`	Разрешение на выполнение сквозного входа в веб-приложение из мобильного приложения	Актуально только для *мобильных приложений* (страница 416).

Какие дополнительные атрибуты можно включить в id_token

Обычно нет необходимости получать атрибуты пользователя непосредственно из маркера идентификации (`id_token`) – более простым и рекомендуемым способом является *получение данных пользователя* (страница 425) через вызов REST-сервиса.

Если все же необходимо получить сведения о пользователе *в составе id_token* (страница 400), то доступные атрибуты выбираются из следующего списка.

Возможные дополнительные атрибуты пользователя в id_token

Атрибут	Описание
`family_name`	Фамилия
`given_name`	Имя
`middle_name`	Отчество
`email`	Адрес электронной почты
`phone_number`	Мобильный телефон

Следующие атрибуты заполняются только в том случае, если пользователь вошел в Blitz Identity Provider через ЕСИА в качестве сотрудника организации.

org_id	Идентификатор организации в Blitz Identity Provider
global_role	Выбранная роль при входе через ЕСИА: • P – физическое лицо • B – индивидуальный предприниматель • L – сотрудник юридического лица • A – сотрудник органа государственной власти
org_shortname	ОГРН организации (по сведениям из учетной записи ЕСИА)
org_fullname	ОГРН организации (по сведениям из учетной записи ЕСИА)
org_ogrn	ОГРН организации (по сведениям из учетной записи ЕСИА)
org_inn	ИНН организации (по сведениям из учетной записи ЕСИА). При аутентификации юридического лица с помощью электронной подписи ИНН организации передается в формате 00 + 10 цифр ИНН юридического лица, при аутентификации юридического лица с помощью учетной записи ЕСИА - в формате 10 цифр ИНН юридического лица.

Совет: Blitz Identity Provider также позволяет поместить элементы дизайна приложения на страницу входа Blitz Identity Provider. При желании создать для подключаемой системы персонифицированную страницу входа нужно адаптировать шаблон оформления страницы входа под дизайн подключаемой системы. Шаблон оформления страницы входа представляет собой zip-архив, внутри которого записаны HTML каркаса страницы входа и используемые на странице таблица стилей, изображения, JavaScript обработчики.

Подготовленный архив темы страницы входа нужно *загрузить* (страница 290) в Blitz Identity Provider.

3.2.2 Подключение веб-приложения

Совет: См. *описание* (страница 229) принципа взаимодействия веб-приложения с Blitz Identity Provider по OIDC.

Настройки подключения

Для подключения мобильного приложения к Blitz Identity Provider потребуются данные, полученные при его *регистрации в продукте* (страница 385):

- идентификатор, присвоенный приложению в Blitz Identity Provider (client_id);
- секрет приложения (client_secret);
- зарегистрированные для приложения URL возврата при авторизации;
- зарегистрированные для приложения URL возврата при логауте;
- зарегистрированные для приложения разрешения (scope).

В целях взаимодействия с Blitz Identity Provider веб-приложение должно использовать следующие адреса:

- URL для проведения авторизации и аутентификации:
  - https://login-test.company.com/blitz/oauth/ae (тестовая среда)
  - https://login.company.com/blitz/oauth/ae (продуктивная среда)
- URL для получения и обновления маркера доступа:
  - https://login-test.company.com/blitz/oauth/te (тестовая среда)
  - https://login.company.com/blitz/oauth/te (продуктивная среда)

- URL для получения данных пользователя:
  - `https://login-test.company.com/blitz/oauth/me` (тестовая среда)
  - `https://login.company.com/blitz/oauth/me` (продуктивная среда)
- URL для получения данных о маркере доступа:
  - `https://login-test.company.com/blitz/oauth/introspect` (тестовая среда)
  - `https://login.company.com/blitz/oauth/introspect` (продуктивная среда)
- URL для выполнения логаута:
  - `https://login-test.company.com/blitz/oauth/logout` (тестовая среда)
  - `https://login.company.com/blitz/oauth/logout` (продуктивная среда)

Все эти URL, а также дополнительные сведения, размещены по адресу динамически обновляемых настроек (метаданных) каждой среды Blitz Identity Provider:

Совет: См. RFC 8414 OAuth 2.0 Authorization Server Metadata[87].

- `https://login-test.company.com/blitz/.well-known/openid-configuration` (тестовая среда)
- `https://login.company.com/blitz/.well-known/openid-configuration` (продуктивная среда)

Разработчики приложений могут не прописывать все указанные URL в конфигурации своего приложения, а использовать в настойках единую ссылку на метаданные Blitz Identity Provider.

Готовые библиотеки

Для интеграции приложения с Blitz Identity Provider можно использовать одну из множества готовых OAuth 2.0 библиотек[88] или реализовать взаимодействие самостоятельно.

Получение кода авторизации

Для проведения идентификации и аутентификации пользователя приложение должно направить пользователя на URL для получения в Blitz Identity Provider кода авторизации, передав в качестве параметров:

- `client_id` – идентификатор клиента;
- `response_type` – тип ответа (принимает значение `code`, `token`, `code token`, `code id_token`, `code id_token token`, `id_token token`, `id_token`);

  Важно: Значение параметра response_type указывает выбранный приложением способ взаимодействия с Blitz Identity Provider:

  - `code` – Authorization Code Flow;
  - `code token`, `code id_token token`, `code id_token token` – Hybrid Flow;
  - `id_token token`, `id_token` – OIDC Implicit Flow;
  - `token` – OAuth 2.0 Implicit Flow.

  ---

[87] https://tools.ietf.org/html/rfc8414
[88] https://oauth.net/code/#client-libraries

- response_mode (необязательный параметр) – позволяет явно указать требуемый способ передачи кода авторизации. При обычном подключении приложения к Blitz Identity Provider данный параметр передаваться не должен, так как рекомендуется использовать стандартные способы передачи кода авторизации (query – для Authorization Code Flow и fragment – для Implicit/Hybrid Flow).

  Возможные значения параметра response_mode:

  – query – значение кода авторизации (code) возвращается на redirect_uri приложения в форме query-параметра. Стандартный режим для Authorization Code Flow.

  – fragment – значение кода авторизации (code) возвращается на redirect_uri приложения в форме fragment-параметра (#). Стандартный режим для Implicit Flow.

  – form_post – в этом режиме параметры ответа на авторизацию кодируются как значения HTML-формы, которые автоматически отправляются в User Agent и передаются клиенту через метод HTTP POST, при этом результирующие параметры кодируются в теле с помощью формата application/x-www-form-urlencoded.

- scope – запрашиваемые разрешения, для проведения аутентификации должно быть передано разрешение openid и необходимые дополнительные scope для получения данных пользователя, например, profile (при запросе нескольких scope они передаются одной строкой и отделяются друг от друга пробелом);

- redirect_uri – ссылка для возврата пользователя в приложение, ссылка должна соответствовать одному из зарегистрированных значений;

- state – набор случайных символов, имеющий вид 128-битного идентификатора запроса (используется для защиты от перехвата), это же значение будет возвращено в ответе – опциональный параметр;

- access_type (необязательный параметр) – требуется ли приложению получать refresh_token, необходимый для получения сведений о пользователе в дальнейшем, когда пользователь будет оффлайн. Принимает значение online или offline, refresh_token предоставляется при access_type=offline. Если значение не задано, то поведение определяется настройкой, заданной для указанного приложения в Blitz Identity Provider;

- prompt (необязательный параметр) – указывает Blitz Identity Provider требуемый режим входа. Возможные значения параметра prompt:

  – none – запрет на аутентификацию.

    Если при выполнении запроса у Blitz Identity Provider возникнет потребность отобразить пользователю экран запроса идентификации/аутентификации, то Blitz Identity Provider не будет этого делать, а вернет системе на ее redirect_uri ошибку login_required. Вызов с параметром prompt=none нужно делать в случае, если приложение хочет проверить наличие у пользователя сессии Blitz Identity Provider, но не хочет, чтобы при выполнении такой проверки пользователю отобразился экран входа Blitz Identity Provider.

  – select_account – запрос смены текущего пользователя.

    Blitz Identity Provider отобразит пользователю экран выбора аккаунта, чтобы пользователь мог войти под другой учетной записью.

  – login – запрет на SSO.

    Если при выполнении запроса Blitz Identity Provider выяснит, что пользователь уже проходит идентификацию/аутентификацию ранее, то Blitz Identity Provider явно потребует от пользователя пройти идентификацию/аутентификацию заново. При этом Blitz Identity Provider дополнительно проверит, что вход будет осуществлен именно тем же самым пользователем, пользовательская сессия которого открыта.

    Если при повторной идентификации/аутентификации пользователь выполнит вход под другой учетной записью, то Blitz Identity Provider вернет системе на ее redirect_uri ошибку

login_required. Вызов с параметром prompt=login нужно делать в случае, если приложение хочет явно запросить у пользователя идентификацию/аутентификацию, например, при доступе к требующей повышенной защиты функции приложения.

Примечание: Для prompt=login для приложения можно при необходимости включить иной сценарий обработки ситуации, что пользователь вошел под другой учетной записью, чем был ранее залогинен в сессии. А именно, можно включить, чтобы при вызове prompt=login осуществлялся принудительный логаут текущей сессии и создание сессии под новой учетной записью. Такое поведение не является рекомендуемым, но может быть включено для приложения по отдельному запросу.

- nonce (необязательный параметр) – строка, используемая для привязки сессии приложения с маркером идентификации. При стандартном подключении приложения к Blitz Identity Provider с использованием Authorization Code Flow параметр nonce использовать нет необходимости.

  При подключении по Implicit Flow или Hybrid Flow данный параметр должен передаваться. Значение nonce должно быть случайной текстовой строкой.

- display (необязательный параметр) – параметр в значении script передается только в случае запуска процесса входа через *HTTP API* (страница 526).

- bip_action_hint (необязательный параметр) – указывает Blitz Identity Provider, что страница входа должна открыться в одном из специальных режимов:
  - open_reg – открыть в режиме регистрации пользователя; при использовании этого режима можно дополнительно указать параметр login_hint со значением email пользователя, и тогда поле «Адрес электронной почты» будет перезаполнено указанным значением email;
  - open_recovery – открыть в режиме восстановления пароля; при использовании этого режима можно дополнительно указать параметр login_hint со значением email пользователя, и тогда поле «Логин» будет перезаполнено указанным значением email;
  - used_externalIdps:esia:esia_1 – открыть в режиме входа через ЕСИА;
  - used_externalIdps:esiadp:esiadp_1 – вход с использованием учетной записи ЕСИА (через получение согласия на доступ к цифровому профилю);
  - used_externalIdps:sbrf:sbrf_1 – открыть в режиме входа через Сбер ID;
  - used_externalIdps:sbb:sbb_1 – открыть в режиме входа через СберБизнес ID;
  - used_externalIdps:tcs:tcs_1 – открыть в режиме входа через Тинькофф ID;
  - used_externalIdps:vtb:vtb_1 – открыть в режиме входа через ВТБ ID;
  - used_externalIdps:alfa:alfa_1 – открыть в режиме входа через Альфа ID;
  - used_externalIdps:mos:mos_1 – открыть в режиме входа через Mos ID (СУДИР);
  - used_externalIdps:apple:apple_1 – открыть в режиме входа через Apple ID;
  - used_externalIdps:facebook:facebook_1 – открыть в режиме входа через Facebook[1];
  - used_externalIdps:google:google_1 – открыть в режиме входа через Google;
  - used_externalIdps:mail:mail_1 – открыть в режиме входа через Mail ID;
  - used_externalIdps:vkid:vkid_1 – открыть в режиме входа через VK ID;
  - used_externalIdps:ok:ok_1 – открыть в режиме входа через Одноклассники;
  - used_externalIdps:vk:vk_1 – открыть в режиме входа через VK;
  - used_externalIdps:yandex:: yandex_1 – открыть в режиме входа через Яндекс;
  - used_password – открыть в режиме входа по паролю (поведение по умолчанию);

[1] Meta признана экстремистской организацией и запрещена в России, деятельность принадлежащих ей соц.сетей Facebook и Instagram также запрещена в РФ.

  - `used_webAuthn` – открыть в режиме входа с использованием FIDO2 ключа (Passkey);
  - `used_x509` – открыть в режиме входа по электронной подписи;
  - `used_qrCode` – открыть в режиме входа по QR-коду;
  - `used_spnego` – открыть в режиме входа по сеансу операционной системы;
  - `used_sms` – открыть в режиме входа по коду в SMS;
  - `used_outside_methodname` – открыть в режиме входа через внешний метод аутентификации с именем `methodname`.

- `bip_user_hint` (необязательный параметр) – передается идентификатор (sub) учетной записи пользователя, которая должна быть выбрана автоматически при открытии экрана входа.

  Идентификатор должен соответствовать одной из запомненных на устройстве учетных записей или страница входа будет открыта в режиме входа нового пользователя;

- `login_hint` (необязательный параметр) – передается значение, которое должно быть заполнено в поле ввода логина, в случае если страница входа открыта в режиме входа нового пользователя.

  Если нужно заполнить логин в случае, когда уже есть запомненный пользователь, то нужно использовать параметр `login_hint` в комбинации с параметром `bip_user_hint`;

- `bip_extIdps_user_choose_hint` (необязательный параметр) – передается идентификатор (sub) учетной записи пользователя, которая должна быть выбрана автоматически в случае входа пользователя через внешний поставщик идентификации, к которому привязано несколько учетных записей Blitz Identity Provider;

- `code_challenge_method` (необязательный параметр) – передается значение “S256”, если подключенное приложение поддерживает спецификацию PKCE для дополнительной защиты взаимодействия с Blitz Identity Provider.

  Совет: См. RFC 7636 Proof Key for Code Exchange by OAuth Public Clients[89].

  Для подключения веб-приложений применение PKCE не является обязательным.

  Для подключения мобильных приложений к Blitz Identity Provider должен использоваться PKCE.

- `code_challenge` (необязательный параметр) – при использовании PKCE в этот параметр передается значение, вычисленное от `code_verifier` по следующей формуле:

  Совет: При отладке удобно использовать онлайн-калькулятор[90].

```
code_challenge=BASE64URL-ENCODE(SHA256(ASCII(code_verifier)))
```

  Примечание: Запрещается открывать страницу входа во фрейме. Пользователь должен видеть URL страницы входа, а также иметь возможность убедиться в наличии HTTPS-соединения веб-порталом `login.company.com`.

Пример запроса на получение кода авторизации (запрошена идентификация/аутентификация и маркер доступа с разрешениями `openid` и `profile`):

```
https://login.company.com/blitz/oauth/ae?client_id=ais&response_type=code&
↪scope=openid+profile&access_type=offline&state=342a2c0c-d9ef-4cd6-b328-
↪b67d9baf6a7f&redirect_uri=https%3A%2F%2Fapp.company.com%2Fre
```

[89] https://tools.ietf.org/html/rfc7636
[90] https://example-app.com/pkce

Пример ответа со значением кода авторизации (`code`) и параметром `state`:

```
https://app.company.com/re?code=f954…nS0&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f
```

Возможные ошибки при вызове `/oauth/ae` соответствуют RFC 6749 и описаны здесь[91].

Пример запроса на получение кода авторизации, при котором Blitz Identity Provider не должен открыть страницу входа в случае, если пользователь еще не проходил идентификацию/аутентификацию в текущем веб-браузере:

```
https://login.company.com/blitz/oauth/ae?client_id=ais&response_type=code&
↪scope=openid+profile&access_type=offline&state=342a2c0c-d9ef-4cd6-b328-
↪b67d9baf6a7f&prompt=none&redirect_uri=https%3A%2F%2Fapp.company.com%2Fre
```

Пример ответа с ошибкой, если для получения кода авторизации пользователь должен явно пройти идентификацию/аутентификацию на странице входа Blitz Identity Provider, а запрос был выполнен с параметром `prompt=none`:

```
https://app.company.com/re?error=login_required&error_
↪description=The+Authorization+Server+requires+End-User+authentication…&
↪state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f
```

Пример запроса на получение кода авторизации, при котором Blitz Identity Provider должен осуществить вход в режиме входа через ЕСИА:

```
https://login.company.com/blitz/oauth/ae?client_id=ais&response_type=code&
↪scope=openid+profile&access_type=offline&state=342a2c0c-d9ef-4cd6-b328-
↪b67d9baf6a7f&bip_action_hint=used_externalIdps:esia:esia_1&redirect_uri=https%3A
↪%2F%2Fapp.company.com%2Fre
```

Пример запроса на получение маркера доступа и маркера идентификации с использованием OIDC Implicit Flow:

```
https://login.company.com/blitz/oauth/ae?client_id=ais&response_type=id_token
↪%20token&scope=openid+profile&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f&nonce=n-
↪0S6_WzA2Mj&redirect_uri=https%3A%2F%2Fapp.company.com%2Fre
```

Пример ответа от Blitz Identity Provider с маркерами доступа и идентификации, полученными с использованием OIDC Implicit Flow:

```
https://app.company.com/re#access_token=SlAV32hkKG&token_type=Bearer&id_
↪token=eyJ0…NiJ9.eyJ1c…I6IjIifX0.DeWt4Qu…ZXso&expires_in=3600&state=342a2c0c-d9ef-
↪4cd6-b328-b67d9baf6a7f
```

Пример запроса на получение кода авторизации и маркера идентификации с использованием OIDC Hybrid Flow:

```
https://login.company.com/blitz/oauth/ae?client_id=ais&response_type=code%20id_
↪token&scope=openid+profile&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f&nonce=n-
↪0S6_WzA2Mj&redirect_uri=https%3A%2F%2Fapp.company.com%2Fre
```

Пример ответа от Blitz Identity Provider с маркерами доступа и идентификации, полученными с использованием OIDC Hybrid Flow:

```
https://app.company.com/re#code=f954…FxS0&id_token=eyJ0…NiJ9.eyJ1c…I6IjIifX0.
↪DeWt4Qu…ZXso&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f
```

[91] https://tools.ietf.org/html/rfc6749#section-4.1.2.1

Получение маркеров

В целях проведения результата идентификации/аутентификации пользователя и получения его данных Blitz Identity Provider выпускает приложению различные маркеры.

Используемые в Blitz Identity Provider маркеры

Название	Обозначение	Предназначение и срок действия
Маркер доступа	`access_token`	Получение доступа к защищенному ресурсу, например, к данным пользователя. Маркер действителен 3600 секунд.
Маркер обновления	`refresh_token`	Обновление маркера доступа. Маркер `refresh_token` предоставляется, только если для приложения при регистрации была указана необходимость получения `refresh_token`, или если в запросе на получение кода авторизации был указан параметр `access_type=offline`. Маркер действителен до момента использования, но не дольше 365 дней.
Маркер идентификации	`id_token`	Получение идентификационной информации, например, идентификатора пользователя. Маркер действителен 3 часа.

Обмен кода авторизации на маркеры

После получения кода авторизации приложение должно обменять его на маркеры.

Внимание: Сервис получения маркеров должен обязательно вызываться с серверов подключенного к Blitz Identity Provider приложения. Вызов сервиса из выполняемого на стороне веб-браузера программного кода (например, из JavaScript кода веб-страницы) **ЗАПРЕЩАЕТСЯ**. Полученный маркер доступа (`access_token`) должен обрабатываться серверной частью приложения и не должен передаваться через браузер пользователя.

Метод `POST https://login.company.com/blitz/oauth/te`

Заголовки `Authorization` со значением `Basic {secret}`, где `secret` – это `client_id:client_secret` (например, `app:topsecret`) в формате Base64.

Тело запроса

- `code` – значение кода авторизации, который был ранее получен;
- `grant_type` – принимает значение `authorization_code`, если код авторизации обменивается на маркер доступа;
- `redirect_uri` – ссылка, по которой должен быть направлен пользователь после того, как даст разрешение на доступ (то же самое значение, которое было указано в запросе на получение кода авторизации);
- `code_verifier` (только если используется PKCE) – значение проверочного кода, использованного при расчете `code_challenge` при получении кода авторизации.

Возвращает

- В случае успеха - маркер доступа, маркер обновления и маркер идентификации.

Совет: Используя полученный маркер доступа, приложение может *запросить* (страница 425) актуальные данные пользователя из Blitz Identity Provider.

- Если код авторизации был уже использован, не совпал `redirect_uri` с ранее использованным в вызове к `/oauth/ae`, или истек срок действия кода, либо переданный `code_verifier` не соответствует `code_challenge`, то в качестве ответа будет возвращена ошибка. Возможные ошибки при вызове `/oauth/te` соответствуют RFC 6749 и описаны здесь[92].

Примеры

Запрос

```
POST /blitz/oauth/te HTTP/1.1
Authorization: Basic cG9ydGFsLmlhc2l1LmxvY2FsOnBvcnRhbC5pYXNpdS5sb2NhbA==
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&code=FLZHS…GU&redirect_uri=https%3A%2F%2Fapp.company.
↪com%2Fre
```

Ответ

```
{
    "id_token": "eyJhbGciOiJSUzI1NiJ9.eyJub…n0=.Ckt…sQ",
    "access_token": "dO-xym…BE",
    "expires_in": 3600,
    "refresh_token": "1lEWX…Iw",
    "token_type": "Bearer"
}
```

Ошибка

```
{
    "error": "invalid_grant",
    "error_description": "The provided authorization grant … is invalid, expired,␣
↪revoked…"
}
```

Обновление маркера доступа

Метод `POST https://login.company.com/blitz/oauth/te`

Заголовки `Authorization` со значением `Basic {secret}`, где `secret` – это `client_id:client_secret` (например, `app:topsecret`) в формате Base64.

Тело запроса

- `refresh_token` – маркер обновления;
- `grant_type` – принимает значение `refresh_token`, если маркер обновления обменивается на маркер доступа.

[92] https://tools.ietf.org/html/rfc6749#section-5.2

Список 1: Пример запроса

```
POST /blitz/oauth/te HTTP/1.1
Authorization: Basic cG9ydGFsLmlhc2l1LmxvY2FsOnBvcnRhbC5pYXNpdS5sb2NhbA==
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token&refresh_token=jj2DA…bQ
```

Обмен маркера доступа

Приложение может обменять `access_token` с одним набором разрешений (`scopes`) и утверждений (`claims`) на `access_token` с другим набором разрешений и утверждений с использованием OAuth 2.0 Token Exchange[93]. Это может быть полезно перед передачей `access_token` от получившего его приложения другому приложению, чтобы приложение получило сокращенный набор разрешений и сведений о пользователе.

Внимание: Для использования обмена маркера доступа приложению должно быть предоставлено специальное разрешение на использование OAuth 2.0 Token Exchange (разрешен grant_type – `urn:ietf:params:oauth:grant-type:token-exchange`). Также должны быть заданы настройки правил обмена маркеров доступа.

Метод `POST https://login.company.com/blitz/oauth/te`

Заголовки `Authorization` со значением `Basic {secret}`, где `secret` – это `client_id:client_secret` (например, `app:topsecret`) в формате Base64.

Тело запроса

Внимание: Должен быть указан один из параметров `resource` или `audience`.

- `grant_type` – принимает значение `urn:ietf:params:oauth:grant-type:token-exchange`.
- `resource` – принимает имя ресурса, для передачи которому запрашивается обмен маркера доступа.
- `audience` – принимает имена приложений, для передачи которым запрашивается маркер доступа.
- `subject_token_type` – передается требуемый тип получаемого маркера. В текущей версии Blitz Identity Provider поддерживается только тип `urn:ietf:params:oauth:token-type:access_token`.
- `subject_token` – передается значение заменяемого маркера доступа (`access_token`).
- Необязательный параметр `scope` – указывает перечень запрашиваемых `scope` в новом маркере. Если данный параметр не указан, то в новый маркер будут включены все `scope`, разрешенные правилом обмена.
- Необязательный параметр `token_format` – указывает требуемый формат для выпускаемого маркера доступа. Возможные значения: `jwt` или `opaque`. Если данный параметр не указан, то новый маркер доступа будет выпущен в том же формате, что и маркер доступа, переданный в `subject_token`.

[93] https://www.rfc-editor.org/rfc/rfc8693.txt

Примеры

Запрос

Список 2: Стандартный запрос

```
POST /blitz/oauth/te HTTP/1.1
Authorization: Basic cG9…A==
Content-Type: application/x-www-form-urlencoded

grant_type=urn:ietf:params:oauth:grant-type:token-exchange&resource=…&subject_
↪token_type=urn:ietf:params:oauth:token-type:access_token&subject_token=eyJ…vA
```

Список 3: Запрос с передачей audience, token_format и scope

```
POST /blitz/oauth/te HTTP/1.1
Authorization: Basic cG9…A==
Content-Type: application/x-www-form-urlencoded

grant_type=urn:ietf:params:oauth:grant-type:token-exchange&token_format=opawue&
↪audience=system1 system2&scope=openid profile&subject_token_
↪type=urn:ietf:params:oauth:token-type:access_token&subject_token=uuy…OE
```

Ответ

```
{
    "access_token": "eyJr...-g",
    "expires_in": 3600,
    "scope": "openid new_scope",
    "token_type": "Bearer",
    "issued_token_type": "urn:ietf:params:oauth:token_type:access_token"
}
```

Ошибка

Список 4: Не найдено правил, разрешающих запрошенный обмен маркера доступа

```
{
    "error": "invalid_target",
    "error_description": "Access denied for resource or audience"
}
```

Список 5: Маркер доступа просрочен

```
{
    "error": "bad_access_token",
    "error_description": "Access token 'CmJ…Dk' not found"
}
```

Использование OAuth 2.0 Resource Owner Password Credentials

Если приложению предоставлено специальное разрешение на использование OAuth 2.0 Resource Owner Password Credentials (ROPC) (разрешен grant_type – `password`), то приложение может запросить получение маркера доступа следующим образом.

Метод `POST https://login.company.com/blitz/oauth/te`

Заголовки `Authorization` со значением `Basic {secret}`, где `secret` – это `client_id:client_secret` (например, `app:topsecret`) в формате Base64.

Тело запроса

- `grant_type` – принимает значение `password`;
- `username` – содержит логин пользователя;
- `password` – содержит пароль пользователя;
- `scope` – содержит список запрашиваемых разрешений.

Возвращает

- В случае успеха - маркер доступа.
- В случае неудачи - ошибку. Возможные значения для `error_description` при проблеме с учетной записью:
  - `Invalid user credentials` – неправильный логин или пароль;
  - `User locked` – учетная запись заблокирована;
  - `User locked by inactivity` – учетная запись заблокирована по причине длительной неактивности;
  - `Password method locked` – для учетной записи включен запрет на использование парольной аутентификации;
  - `Password method not configured` – метод парольной аутентификации не сконфигурирован;
  - `Password expired` – срок действия пароль истек;
  - `Need password change` – требуется обязательная смена пароля при входе.

Запрос

```
POST /blitz/oauth/te HTTP/1.1
Authorization: Basic cG9…A==
Content-Type: application/x-www-form-urlencoded

grant_type=password&username=testuser&password=testpwd1&scope=profile
```

Ответ

```
{
    "access_token": "dO-xym…BE",
    "expires_in": 3600,
    "scope": "profile",
    "token_type": "Bearer"
}
```

Ошибка

```
{
    "error": "invalid_grant",
    "error_description": "Invalid user credentials"
}
```

Маркер идентификации

Для получения данных об идентификации и аутентификации приложение может самостоятельно анализировать содержание маркера идентификации (`id_token`).

Совет: Вместо анализа `id_token` рекомендуется использовать запрос на *актуализацию данных пользователя* (страница 425) по маркеру доступа.

Структура маркера Маркер идентификации состоит из трех частей:

- заголовок (`header`), в котором содержится общая информация о типе маркера, в том числе об использованных в ходе его формирования криптографических операциях;
- набор утверждений (`payload` / `claim set`) с содержательными сведениями о маркере;
- подпись (`signature`), которая удостоверяет, что маркер выдан Blitz Identity Provider и не был изменен при передаче.

Части маркера разделены точкой, он имеет вид:

```
HEADER.PAYLOAD.SIGNATURE
```

Маркер передается в виде строки в формате `Base64url`.

Заголовок маркера

- `alg` – описание алгоритма шифрования (параметр `alg`); в настоящее время в Blitz Identity Provider поддерживается алгоритм электронной подписи `RSA SHA-256`, рекомендуемый спецификацией (соответствует значению `RS256`);
- `kid` – идентификатор ключа, использованного для подписи маркера.

Набор утверждений Атрибуты:

- `exp` – время прекращения действия, указывается в секундах с 1 января 1970 г. 00:00:00 GMT;
- `iat` – время выдачи, указывается в секундах с 1 января 1970 г. 00:00:00 GMT;
- `sub` – идентификатор субъекта, в качестве значения указывается значение идентификатора пользователя;
- `ua_id` – идентификатор устройства пользователя;

- aud – адресат маркера, указывается client_id приложения, направившего запрос на аутентификацию;
- iss – организация, выпустившая маркер, указывается URL issuer, по умолчанию https://login.company.com/blitz;
- nonce – строка безопасности, указывается значение nonce, которое было передано приложением к Blitz Identity Provider в исходном запросе к /oauth/ae. Используется только при Implicit или Hybrid Flow. При получении приложением маркера с использованием Implicit или Hybrid Flow приложение должно сопоставить nonce из состава маркера идентификации с nonce из своего запроса;
- at_hash – половина хэша маркера доступа, передается только при использовании Implicit или Hybrid Flow. Представляет собой закодированную в Base64 левую половину значения функции SHA-256 от access_token. Приложение, получившее маркер доступа с использованием Implicit или Hybrid Flow должно извлечь из маркера идентификации значение at_hash и сравнить с маркером доступа.
- c_hash – половина хэша кода авторизации, передается только в случае использования Hybrid Flow. Представляет собой закодированную в Base64 левую половину (128 бит) значения функции SHA-256 от кода авторизации (code); Приложение, получившее код авторизации с использованием Hybrid Flow, должно извлечь из маркера идентификации значение c_hash и сравнить с кодом авторизации.
- amr – пройденные методы аутентификации, указывается список пройденных пользователем методов аутентификаций. Список может включать следующие идентификаторы методов:
  - password – вход с использованием пароля;
  - cls:<метод> (например, cls:password) – автоматический вход с запомненного устройства (в названии идентификатора после двоеточия указан метод аутентификации, первично пройденной пользователем, в результате чего произошло запоминание пользователя на данном устройстве);
  - css – автоматический вход по результатам регистрации пользователя, восстановления пароля или перехода в веб-приложение из мобильного приложения, использующего вызов с использованием scope=native;
  - sms – подтверждение входа с помощью кода в SMS-сообщении (второй фактор аутентификации);
  - email – подтверждение входа с помощью кода в сообщении электронной почты (второй фактор аутентификации);
  - push – подтверждение входа с помощью кода в push-уведомлении в мобильное приложение (второй фактор аутентификации);
  - hotp – подтверждение входа с помощью кода, сгенерированного HOTP-генератором кодов подтверждения (второй фактор аутентификации);
  - totp – подтверждение входа с помощью кода, сгенерированного программным TOTP-генератором кодов подтверждения (второй фактор аутентификации);
  - tls – вход в режиме автоматической аутентификации с использованием TLS Proxy;
  - spnego – вход с использованием сеанса операционной системы;
  - userApp – вход в мобильное приложение привязанной к устройству учетной записью пользователя (Touch ID/Face ID/ПИН-код);
  - webAuthn – вход с использованием FIDO2 ключа (Passkey) или подтверждение входа с помощью U2F-ключа;
  - x509 – вход с использованием электронной подписи;
  - qrCode – вход по QR-коду;
  - externalIdps:esia:esia_1 – вход с использованием учетной записи ЕСИА;
  - externalIdps:esiadp:esiadp_1 – вход с использованием учетной записи ЕСИА (через получение согласия на доступ к цифровому профилю);

  - externalIdps:sbrf:sbrf_1 – вход с использованием учетной записи Сбер ID;
  - externalIdps:sbb:sbb_1 – вход с использованием учетной записи СберБизнес ID;
  - externalIdps:tcs:tcs_1 – вход с использованием учетной записи Тинькофф ID;
  - externalIdps:vtb:vtb_1 – вход с использованием учетной записи ВТБ ID;
  - externalIdps:alfa:alfa_1 – вход с использованием учетной записи Альфа ID;
  - externalIdps:mos:mos_1 – вход с использованием учетной записи Mos ID (СУДИР);
  - externalIdps:apple:apple_1 – вход с использованием учетной записи Apple ID;
  - externalIdps:facebook:facebook_1 – вход с использованием учетной записи в социальной сети Facebook[с. 392, 1];
  - externalIdps:google:google_1 – вход с использованием учетной записи Google;
  - externalIdps:mail:mail_1 – вход с использованием учетной записи Mail ID;
  - externalIdps:vkid:vkid_1 – вход с использованием учетной записи VK ID;
  - externalIdps:ok:ok_1 – вход с использованием учетной записи в социальной сети Одноклассники;
  - externalIdps:vk:vk_1 – вход с использованием учетной записи в социальной сети VK;
  - externalIdps:yandex:yandex_1 – вход с использованием учетной записи Яндекс;
  - outside_methodname – признак, что в процессе входа пользователь использовал внешний метод аутентификации с именем methodname.

- sid – идентификатор сессии пользователя;
- дополнительные атрибуты в соответствии с заявкой на подключение приложения к Blitz Identity Provider (см. возможные атрибуты для включения в id_token *здесь* (страница 388)).

Список 6: Пример набора утверждений

```
{
    "exp": 1445004777,
    "iat": 1444994212,
    "ua_id": "f8a235ff-cb85-4c4b-b55d-544f9358a8d7",
    "sub": "3d10f626-ea77-481d-a50bd4a4d432d86b",
    "amr": [
        "externalIdps:esia:esia_1"
    ],
    "aud": [
        "ais"
    ],
    "iss": "https://login.company.com/blitz",
    "sid": "5a600d12-4b14-447e-ba21-2dc40344a44a"
}
```

Подпись маркера Осуществляется по алгоритму, который указывается в параметре alg маркера. Подпись вычисляется от двух предыдущих частей маркера (HEADER.PAYLOAD). Сертификат открытого ключа Blitz Identity Provider, необходимый для проверки подписи, можно загрузить по следующим ссылкам (находится в атрибуте x5c, идентификатор ключа находится в атрибуте kid):

- https://login-test.company.com/blitz/.well-known/jwks (тестовая среда)
- https://login.company.com/blitz/.well-known/jwks (продуктивная среда)

Работа с маркером идентификации

1. После получения маркера идентификации приложению рекомендуется произвести валидацию маркера идентификации, которая включает в себя следующие проверки:

1. Получение идентификатора Blitz Identity Provider (`sub`), содержащегося в маркере идентификации, и получение иных необходимых приложению дополнительных атрибутов пользователя.
2. Проверка идентификатора приложения, т.е. именно приложение должно быть указано в качестве адресата маркера идентификации.
3. Проверка подписи маркера идентификации (с использованием указанного в маркере алгоритма).
4. Проверка, что текущее время должно быть не позднее, чем время прекращения срока действия маркера идентификации.

После валидации маркера идентификации приложение может считать пользователя аутентифицированным.

2. Для анализа содержания маркера идентификации, а также для упрощения разработки модулей по его проверке можно воспользоваться доступными онлайн-декодерами и библиотеками.

Совет: См. ресурсы http://jwt.io/ и http://kjur.github.io/jsjws/mobile/tool_jwt.html#verifier.

Проверка маркера доступа через сервис интроспекции

Данные о маркере доступа (`access_token`) необходимо проверять в следующих случаях:

- приложению требуется отслеживать срок действия маркера, чтобы оперативно менять его на новый;
- к приложению предъявляются повышенные требования к безопасности, и приложение хочет через проверку маркера убедиться, что маркер не аннулирован досрочно. Аннулирование маркера доступам (`access_token`) или маркера идентификации (`id_token`) может произойти в целях безопасности в случае, если произошли сброс/изменение пароля учетной записи пользователя или если учетная запись пользователя была заблокирована;
- приложение является поставщиком ресурсов и предоставляет доступ к этим ресурсам по предъявлению маркера доступа, выданного Blitz Identity Provider приложению, запрашивающему ресурс.

Метод `POST https://login.company.com/blitz/oauth/introspect`

Совет: См. RFC 7662 OAuth 2.0 Token Introspection[94].

Сервис интроспекции может быть вызван любой системой, зарегистрированной в Blitz Identity Provider, для проверки любого маркера доступа (система может проверить маркер, выданный другой системе). Проверять можно не только маркер доступа, но и маркер обновления.

Заголовки

- `Authorization` со значением `Basic {secret}`, где `secret` – это `client_id:client_secret` (например, `app:topsecret`) в формате Base64;
- `Content-Type` со значением `application/x-www-form-urlencoded`.

Тело запроса

- `token` – маркер доступа, данные о котором требуется просмотреть.
- Необязательный параметр `token_type_hint` – тип маркера доступа (например, `access_token`), предназначен для ускорения поиска.

Возвращает Данные о маркере доступа:

- `active` – признак действительности маркера доступа, принимает значения `true` или `false`. Маркер действителен, если он выдан сервисом авторизации Blitz Identity Provider, не был отозван и срок его действия не истек;

[94] https://tools.ietf.org/html/rfc7662

- `scope` – область доступа, на которую выдан маркер доступа. Передается в виде перечня разрешений;
- `client_id` – идентификатор системы-клиента, которая получила данный маркер доступа;
- `sub` – идентификатор пользователя (владельца ресурса, предоставившего доступ к своим данным), определенный как базовый идентификатор в Blitz Identity Provider. Значение параметра возвращается только в том случае, если он может быть передан в рамках `scope` по предъявленному маркеру доступа;
- `jti` – идентификатор маркера доступа (в виде строки);
- `token_type` – тип предъявленного маркера доступа;
- `iat` – время выдачи маркера (в Unix Epoch);
- `exp` – время окончания действия маркера (в Unix Epoch).

Примеры

Запрос

```
POST /blitz/oauth/introspect HTTP/1.1
Authorization: Basic cG9ydGFsLmlhc2l1LmxvY2FsOnBvcnRhbC5pYXNpdS5sb2NhbA==
Content-Type: application/x-www-form-urlencoded

token=MkvRf…No
```

Ответ

Список 7: Действующий access_token

```
{
   "sub": "3d10f626-ea77-481d-a50bd4a4d432d86b",
   "scope": "openid profile",
   "jti": "10jdlNohfHzuv3xoFurvWSPheEJEC7KHdHr-dcaVyYYvV3h0l2sh",
   "token_type": "Bearer",
   "client_id": "ais",
   "active": true,
   "iat": 1699938503,
   "exp": 1699942103
}
```

Список 8: Действующий id_token

```
{
    "exp": 1699939472,
    "iat": 1699935872,
    "jti": "fU2FTCzm9G5I4YC6VDFnfjFY5QeIULwHlYo_BH6OuCQ",
    "token_type": "id_token",
    "active": true,
    "client_id": "ais",
    "sub": "3d10f626-ea77-481d-a50bd4a4d432d86b"
}
```

Список 9: Действующий refresh_token

```
{
    "sub": "3d10f626-ea77-481d-a50bd4a4d432d86b",
    "scope": "openid profile",
    "jti": "10jdlNohfHzuv3xoFurvWSPheEJEC7KHdHr-dcaVyYYvV3h0l2sh",
    "token_type": "refresh_token",
    "client_id": "ais",
    "active": true,
    "iat": 1699938503,
    "exp": 1699942103
}
```

Список 10: Недействительный маркер доступа

```
{
    "active": false
}
```

Проверка маркера доступа приложением

При регистрации приложения в Blitz Identity Provider можно указать, что приложение должно получать маркер доступа (`access_token`) в формате `JWT`. В этом случае приложение получает возможность самостоятельно проверить маркер доступа, выполнив его разбор.

Структура первично полученного маркера доступа будет аналогична структуре *этого маркера идентификации* (страница 400). Во вторичных маркерах доступа, полученных в результате обмена маркера обновления (`refresh_token`), не будет содержаться сессионная информация (будут отсутствовать `amr` и дополнительные атрибуты пользователя).

Маркеры доступа в формате `JWT` следует использовать, только в случае если у приложения на это есть особые причины. В остальных случаях рекомендуется использовать обычные маркеры доступа в формате `opaque`.

Логаут

Если приложение предоставляет пользователю возможность инициировать выход из приложения (логаут), то приложению для обеспечения выхода недостаточнозавершить локальную сессию. Необходимо также вызвать в Blitz Identity Provider операцию логаута.

Если этого не сделать, то может возникнуть ситуация, что пользователь в приложении нажал кнопку Выход, после чего сразу попробовал нажать кнопку Вход, и вместо ожидаемого запроса идентификации и аутентификации сработал `Single Sign-On`, и пользователь сразу автоматически оказался авторизованным.

Для инициирования логаута в Blitz Identity Provider приложение после закрытия своей локальной сессии должно направить пользователя в Blitz Identity Provider на URL для выполнения логаута, передав в качестве параметров:

Примечание: Вызов логаута выполняется в соответствии со спецификацией OpenID Connect RP-Initiated Logout 1.0[95].

- Необязательный параметр `id_token_hint` - Blitz Identity Provider проверяет, что `id_token` из значения параметра выпущен им. Допустимые адреса возврата при логауте и дизайн страницы выхода используются в соответствии с настроенным приложением с `client_id` из поля `aud` из `id_token`.

[95] https://openid.net/specs/openid-connect-rpinitiated-1_0.html

- Необязательный параметр `client_id` – допустимые адреса возврата при логауте и дизайн страницы выхода используются в соответствии с указанным `client_id`.
- Необязательный параметр `post_logout_redirect_uri` – адрес возврата в приложение после логаута. Если параметр не задан, то перенаправление в приложение после логаута не осуществляется. Если задан, то проверяется, что значение соответствует хотя бы одному разрешенному префиксу возврата для приложения, соответствующего переданному в `id_token_hint` приложению (поле `aud` из `id_token`) или переданному `client_id`. При передаче параметра `post_logout_redirect_uri` обязательно также передать параметр `id_token_hint` или `client_id`.
- `state` - набор случайных символов, имеющий вид 128-битного идентификатора запроса. Это же значение будет возвращено в ответе при перенаправлении пользователя на `post_logout_redirect_uri`.

Пример запроса логаута:

```
https://login.company.com/blitz/oauth/logout?id_token_hint=eyJhbGciOiJSUzI1NiJ9.
↪eyJub…n0=.Ckt…sQ&post_logout_redirect_uri=https://app.company.com/redirect_uri&
↪state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f
```

Если Blitz Identity Provider успешно завершит логаут, то он перенаправит пользователя по переданному URL обратно в приложение.

Альтернативный пример запроса логаута:

```
https://login.company.com/blitz/oauth/logout?client_id=test-app&post_logout_
↪redirect_uri=https://app.company.com/redirect_uri&state=342a2c0c-d9ef-4cd6-b328-
↪b67d9baf6a7f
```

Допустимые префиксы страниц возврата должны быть зарегистрированы в настройках Blitz Identity Provider, иначе при логауте будет выдана ошибка.

Приложения, подключенные к Blitz Identity Provider по OIDC, могут подписаться на уведомление их о логауте пользователя из Blitz Identity Provider. Поддерживаются следующие возможности:

- Уведомление через веб-браузер (Front channel) См. OpenID Connect Front-Channel Logout 1.0[96].
- Уведомление через сервер (Back channel). См. OpenID Connect Back-Channel Logout 1.0[97].

Для уведомления через веб-браузер в настройках приложения в Blitz Identity Provider регистрируется обработчик «Ссылка для очистки сессии пользователя в браузере (Front channel)». Если обработчик зарегистрирован и в процессе сессии пользователь входил в приложение, то при вызове пользователем логаута Blitz Identity Provider через браузер на странице выхода пользователя через фрейм `<iframe src= "ссылка">` вызовет через HTTP GET указанный в настройке обработчик приложения. В случае если была отмечена настройка «Добавлять идентификатор сессии и эмитента в ссылку очистки сессии в браузере (Front channel)», то дополнительно будут переданы следующие параметры в запросе:

- `iss` – идентификатор поставщика идентификации;
- `sid` – идентификатор сессии пользователя.

Пример вызова ссылки для очистки сессии пользователя в браузере (Front channel):

```
https://app.company.com/front_channel_logout?iss=https://login.company.com/blitz&
↪sid=4ac78c75-b99d-44dc-9304-d2599c829440
```

В ответ на вызов приложение должно завершить локальную сессию и вернуть ответ `HTTP 200 OK`. Также в ответ должны быть включены заголовки:

[96] https://openid.net/specs/openid-connect-frontchannel-1_0.html
[97] https://openid.net/specs/openid-connect-backchannel-1_0.html

```
Cache-Control: no-cache, no-store
Pragma: no-cache
```

Примечание: При реализации на стороне приложения обработчика приема уведомления через веб-браузер следует учитывать особенности современных браузеров, которые противодействуют передаче cookies при вызове обработчиков в фрейме на URL-домены, отличные от URL-домена родительской страницы:

– чтобы cookie стороннего сайта могла быть переданы из фрейма, у cookie должен быть установлен флаг `SameSite=None` и флаг `Secure`, в момент установки или перезаписи cookie не должен передаваться заголовок `X-Frame-Options`, а сам обработчик должен быть доступен по HTTPS;

– вызов обработчика не будет производиться в некоторых браузерах в случае открытия страницы в режиме «инкогнито».

Для уведомления через сервер в настройках приложения в Blitz Identity Provider регистрируется обработчик «Ссылка для очистки сессии пользователя в приложении (Back channel)». Если обработчик зарегистрирован и в процессе сессии пользователь входил в приложение, то при вызове пользователем логаута сервер Blitz Identity Provider вызовет сервер приложения через HTTP POST на указанный в настройке обработчик приложения. В вызов будет передан маркер логаута `logout_token`, представляющий собой JWT-токен, в теле которого содержатся следующие параметры:

- `iss` – идентификатор поставщика идентификации;
- `aud` – идентификаторы оповещаемых приложений;
- `iat` – время выпуска маркера обновления;
- `jti` – идентификатор маркера логаута;
- `events` – константное значение `http://schemas.openid.net/event/backchannel-logout` согласно спецификации OpenID Connect Back-Channel Logout 1.0;
- `sid` – идентификатор сессии пользователя;
- `sub` – идентификатор пользователя.

В маркере обновления присутствует либо `sub` (если не включена настройка «Добавлять идентификатор сессии и эмитента в ссылку очистки сессии в приложении (Back channel)», либо `sid` (если настройка «Добавлять идентификатор сессии и эмитента в ссылку очистки сессии в приложении (Back channel)» включена).

Пример вызова сервиса очистки сессии пользователя в приложении (Back channel):

```
POST /back_channel_logout HTTP/1.1
Host: app.company.com
Content-Type: application/x-www-form-urlencoded

logout_token=eyJ…J9.eyJ…J9.RV8…Nw
```

Пример разобранного тела маркера логаута при выключенной настройке «Добавлять идентификатор сессии и эмитента в ссылку очистки сессии в приложении (Back channel)»:

```
{
    "iss": "https://login.company.com/blitz",
    "aud": [
        "ais"
    ],
    "iat": 1646979918,
    "jti": "ee75ccd8-ad30-4175-9a61-3ae06c1a6730",
    "events": {
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        "http://schemas.openid.net/event/backchannel-logout": {}
    },
    "sub": "3d10f626-ea77-481d-a50bd4a4d432d86b"
}
```

Пример разобранного тела маркера логаута при включенной настройке «Добавлять идентификатор сессии и эмитента в ссылку очистки сессии в приложении (Back channel)»:

```
{
    "iss": "https://login.company.com/blitz",
    "aud": [
        "ais"
    ],
    "iat": 1646979918,
    "jti": "ee75ccd8-ad30-4175-9a61-3ae06c1a6730",
    "events": {
        "http://schemas.openid.net/event/backchannel-logout": {}
    },
    "sid": "4ac78c75-b99d-44dc-9304-d2599c829440"
}
```

В ответ на вызов приложение должно:

1. Проверить подпись маркера логаута по аналогии с тем, как выполняется *проверка подписи маркера идентификации* (страница 400).
2. Проверить, что:

- `iss` соответствует идентификатору развернутой системы `issuer`;
- `aud` включает идентификатор вызванного приложения;
- маркер обновления выпущен (`iat`) не ранее 2 минут назад;
- `sid` или `sub` соответствуют действующим сессиям пользователя.

3. Если какие-то проверки маркера логаута неуспешны, то вернуть код `HTTP 400 Bad Request`.
4. Если все проверки успешны, то завершить локальную сессию пользователя и вернуть `HTTP 200 OK` в случае успеха или `HTTP 501 Not Implemented` в случае, если сессию завершить не удалось.

   Рекомендуется включить в ответ заголовки:

```
Cache-Control: no-cache, no-store
Pragma: no-cache
```

3.2.3 Подключение мобильного приложения

Совет: См. *описание* (страница 231) принципа взаимодействия мобильного приложения с Blitz Identity Provider по OIDC.

Настройки подключения

Для подключения мобильного приложения к Blitz Identity Provider потребуются данные, полученные при его *регистрации в продукте* (страница 385):

- идентификатор, присвоенный приложению в Blitz Identity Provider (`software_id`);
- первичный маркер доступа (`Initial Access Token`);
- метаданные приложения (`software_statement`);
- зарегистрированные для приложения URL возврата при авторизации;
- зарегистрированные для приложения URL возврата при логауте;
- зарегистрированные для приложения разрешения (`scope`).

В целях взаимодействия с Blitz Identity Provider приложение должно использовать следующие адреса:

- URL для проведения авторизации и аутентификации:
  - `https://login-test.company.com/blitz/oauth/ae` (тестовая среда)
  - `https://login.company.com/blitz/oauth/ae` (продуктивная среда)
- URL для получения и обновления маркера доступа:
  - `https://login-test.company.com/blitz/oauth/te` (тестовая среда)
  - `https://login.company.com/blitz/oauth/te` (продуктивная среда)
- URL для получения данных пользователя:
  - `https://login-test.company.com/blitz/oauth/me` (тестовая среда)
  - `https://login.company.com/blitz/oauth/me` (продуктивная среда)
- URL для динамической регистрации экземпляра мобильного приложения:
  - `https://login-test.company.com/blitz/oauth/register` (тестовая среда)
  - `https://login.company.com/blitz/oauth/register` (продуктивная среда)
- URL для получения данных о маркере доступа:
  - `https://login-test.company.com/blitz/oauth/introspect` (тестовая среда)
  - `https://login.company.com/blitz/oauth/introspect` (продуктивная среда)
- URL для выполнения логаута:
  - `https://login-test.company.com/blitz/oauth/logout` (тестовая среда)
  - `https://login.company.com/blitz/oauth/logout` (продуктивная среда)

Все эти URL, а также дополнительные сведения, размещены по адресу динамически обновляемых настроек (метаданных) каждой среды Blitz Identity Provider:

Совет: См. RFC 8414 OAuth 2.0 Authorization Server Metadata[98].

- `https://login-test.company.com/blitz/.well-known/openid-configuration` (тестовая среда)
- `https://login.company.com/blitz/.well-known/openid-configuration` (продуктивная среда)

Разработчики приложений могут не прописывать все указанные URL в конфигурации своего приложения, а использовать в настойках единую ссылку на метаданные Blitz Identity Provider.

[98] https://tools.ietf.org/html/rfc8414

Готовые библиотеки

Для интеграции мобильных приложений с Blitz Identity Provider будет полезен информационный ресурс `https://appauth.io/`, предоставляющий SDK для iOS/Android.

Динамическая регистрация экземпляра приложения

Предварительные условия для динамической регистрации экземпляра мобильного приложения:

- пользователь должен установить мобильное приложение;
- мобильное приложение должно иметь следующие данные:
  - идентификатор мобильного приложения (`software_id`);
  - первичный маркер доступа (`Initial Access Token`);
  - метаданные мобильного приложения (`software_statement`).

Мобильное приложение должно отправить HTTP-запрос методом POST в Blitz Identity Provider по адресу сервиса динамической регистрации `/blitz/oauth/register`.

Должны быть переданы параметры:

- идентификатор мобильного приложения (`software_id`);
- метаданные мобильного приложения (`software_statement`);
- тип устройства, на котором работает мобильное приложение (`device_type`) – одно из возможных значений, представленных в таблице:

Используемые в Blitz Identity Provider маркеры

Тип устройства (`device_type`)	Описание
`iphone`	Смартфоны семейства iPhone
`ipad`	Планшеты семейства iPad
`android_phone`	Смартфоны под управлением ОС Android
`android_tab`	Планшеты под управлением ОС Android
`win_mobile`	Устройства под управлением Windows 10 Mobile

Запрос на динамическую регистрацию должен содержать заголовок `Authorization` с первичным маркером доступа (тип – `Bearer`), выданным приложению.

Пример запроса:

```
POST /blitz/oauth/register HTTP/1.1
Content-Type: application/json
Authorization: Bearer NINxnizbgYYQg94vEd6MjkTPxR3r2s9IAHBO92AszgTIqItY

{
   "software_id": "CSI",
   "device_type": "iphone",
   "software_statement": "eyJ0e…xQ"
}
```

При успешном выполнении запроса Blitz Identity Provider возвращает экземпляру мобильного приложения перечень утверждений, среди которых для дальнейшей работы необходимы следующие (их нужно защищенным образом сохранить в устройстве пользователя):

- идентификатор экземпляра мобильного приложения (`client_id`);
- секрет экземпляра мобильного приложения (`client_secret`);

- маркер управления конфигурацией (`registration_access_token`);
- URL управления конфигурацией (`registration_client_uri`).

Пример ответа:

```
{
    "grant_types": [
        "authorization_code"
    ],
    "registration_client_uri": "https://login.company.com/blitz/oauth/register/dyn~
↪CSI~4e6904c5-ef29-4ae5-8d30-99c359b8270f",
    "scope": "openid profile",
    "registration_access_token": "eyJ0e…tw",
    "client_id": "dyn~CSI~4e6904c5-ef29-4ae5-8d30-99c359b8270f",
    "software_id": "CSI",
    "software_version": "1",
    "token_endpoint_auth_method": "client_secret_basic",
    "response_types": [
        "code"
    ],
    "redirect_uris": [
        "com.example.app:/oauth2redirect/example-provider"
    ],
    "client_secret": "3r0tt2OlyeGecWq",
    "client_secret_expires_at": 0
}
```

Первичный вход пользователя

Получив (страница 410) пару `client_id` / `client_secret` экземпляр мобильного приложения должен провести идентификацию и аутентификацию пользователя согласно спецификациям OIDC/OAuth 2.0 и с учетом дополнительной спецификации RFC 7636 Proof Key for Code Exchange by OAuth Public Clients[99] (мобильное приложение при взаимодействии с Blitz Identity Provider должно использовать PKCE).

Сценарий идентификация и аутентификации включает следующие шаги:

- запрос на получение кода авторизации;
- получение маркера доступа;
- получение данных пользователя в обмен на маркер доступа.

Первичный вход пользователя в мобильное приложение должен произойти в течение 1 часа с завершения динамической регистрации в Blitz Identity Provider экземпляра мобильного приложения. Иначе `client_id` будет аннулирован и потребуется повторная динамическая регистрация.

Получение кода авторизации

Для проведения аутентификации экземпляр мобильного приложения должен вызвать штатный браузер мобильной платформы и перенаправить в нем пользователя на URL Blitz Identity Provider сервиса проведения авторизации и аутентификации (`/blitz/oauth/ae`).

При использовании браузера мобильным приложением следует учесть следующие особенности:

- для iOS необходимо использовать встроенный браузер: класс `SFSafariViewController` или класс `SFAuthenticationSession` (`in-app browser tab pattern`);
- для Android необходимо использовать встроенный браузер: функция `Android Custom Tab` (реализует `in-app browser tab pattern`).

[99] https://tools.ietf.org/html/rfc7636

Внимание: Использование Embedded-браузера не допускается.

В качестве параметров запроса следует указать:

- `client_id` – идентификатор экземпляра мобильного приложения;
- `response_type` – тип ответа (принимает значение `code`);
- `scope` – запрашиваемые разрешения, должно быть передано разрешение `openid` и необходимые дополнительные `scope` для получения данных пользователя (эти `scope` должны быть предусмотрены метаданными);
- `redirect_uri` – ссылка для возврата пользователя в приложение, ссылка должна соответствовать одному из указанных в метаданных значений. Чтобы после авторизации Blitz Identity Provider смог обратно вызвать мобильное приложение, следует использовать следующие схемы:
  - для iOS:

    Совет: Пример реализации – см.: https://github.com/openid/AppAuth-iOS

    * вариант 1 – использовать private-use URI scheme (`custom URL scheme`). Вид ссылок возврата: `com.example.app:/oauth2redirect/example-provider` (регистрируются в Info.plist ключи типа CFBundleURLTypes);
    * вариант 2 – использовать URI вида `https` (`Universal links`). Вид ссылок возврата: `https://app.example.com/oauth2redirect/example-provider` (используется функция «Universal links», URL регистрируются в `entitlement-файле` в приложении и ассоциированы с доменом приложения). Этот способ предпочтительнее для iOS 9 и выше.
  - для Android:

    Совет: Пример реализации – см.: https://github.com/openid/AppAuth-Android

    * вариант 1 – использовать private-use URI scheme (`custom URL scheme`). Вид ссылок возврата: `com.example.app:/oauth2redirect/example-provider` (поддержка ссылок с помощью Android Implicit Intents, ссылки регистрируются в manifest);
    * вариант 2 – использовать URI вида `https` (`Universal links`). Вид ссылок возврата: `https://app.example.com/oauth2redirect/example-provider` (доступно начиная с Android 6.0, ссылки регистрируются в manifest). Этот способ предпочтительнее для Android 6.0 и выше.
- `state` – набор случайных символов, имеющий вид 128-битного идентификатора запроса (используется для защиты от перехвата), это же значение будет возвращено в ответе – опциональный параметр;
- `access_type` (необязательный параметр) – требуется ли приложению получать refresh_token, необходимый для получения сведений о пользователе в дальнейшем, когда пользователь будет оффлайн. Принимает значение “online”/“offline”, refresh_token предоставляется при access_type=offline. Если значение не задано, то поведение определяется настройкой, заданной для указанного приложения в Blitz Identity Provider;
- `code_challenge_method` – метод шифрования идентификатора запроса, следует указывать “S256”;
- `code_challenge` – зашифрованный идентификатор запроса. Идентификатор запроса (`code_verifier`) должен быть запомнен экземпляром мобильного приложения для последующей передачи в запрос на получение маркера доступа. Шифрованное значение вычисляется следующим образом:

```
code_challenge=BASE64URL-ENCODE(SHA256(ASCII(code_verifier)))
```

Пример запроса на получение кода авторизации (запрошена аутентификация и маркер доступа с разрешениями `openid` и `profile`, используется PKCE):

```
https://login.company.com/blitz/oauth/ae?scope=openid+profile
&access_type=online&response_type=code
&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f
&client_id=dyn~CSI~4e6904c5-ef29-4ae5-8d30-99c359b8270f
&code_challenge_method=S256&code_challenge=qjrzSW9gMiUgpUvqgEPE4
&redirect_uri=https%3A%2F%2Fapp.example.com%2Foauth2redirect%2Fexample-provider
```

Пример ответа со значением кода авторизации (`code`) и параметром `state`:

```
https://app.example.com/oauth2redirect/example-provider?
↪code=f954nEzQ08DXju4wxGbSSfCX7TkZ1GvXUR7TzVus8fGnu4AUl-YIosgax-
↪BLXMeQQAlasD6CN2qG_0KXK5NIjARoKykhuR9IpbuzqeFxS0&state=342a2c0c-d9ef-4cd6-b328-
↪b67d9baf6a7f
```

Возможные ошибки при вызове `/oauth/ae` соответствуют RFC 6749 и описаны здесь[100].

Получение маркеров экземпляром приложения

После получения кода авторизации экземпляр мобильного приложения должен обменять его на маркеры. Для этого экземпляр должен сформировать запрос методом POST на URL для получения маркера. Запрос должен содержать заголовок `Authorization` со значением `Basic {secret}`, где `secret` – это `client_id:client_secret` (например, `dyn~CSI~4e69…Wq`) в формате Base64.

Пример заголовка:

```
Authorization: Basic ZHluOkNTSTo…dx
```

Тело запроса должно содержать следующие параметры:

- `code` – значение кода авторизации, который был ранее получен экземпляром мобильного приложения от Blitz Identity Provider;
- `grant_type` – значение `authorization_code`;
- `redirect_uri` – должно быть то же самое значение, которое было указано в запросе на получение кода авторизации;
- `code_verifier` – идентификатор запроса, сгенерированный экземпляром мобильного приложения при запросе на получение кода авторизации.

Пример запроса:

```
POST /blitz/oauth/te HTTP/1.1
Authorization: Basic ZHluOkNTSTo…dx
Content-Type: application/x-www-form-urlencoded
grant_type=authorization_code&code=FLZHS…GU
&redirect_uri=https%3A%2F%2Fapp.example.com%2Foauth2redirect%2Fexample-provider
&code_verifier=M25iVXpKU3puUjFaYWg3T1NDTDQtcW1ROUY5YXlwalNoc0hhakxifmZHag
```

В ответ возвращается маркер доступа и маркер идентификации.

Пример ответа с успешным выполнением запроса:

[100] https://tools.ietf.org/html/rfc6749#section-4.1.2.1

```
{
    "id_token": "eyJhb…J9. eyJub…0=.Ckt_dr…sQ",
    "access_token": "dO-xym…BE",
    "expires_in": 3600,
    "scope": "openid profile",
    "token_type": "Bearer"
}
```

После получения маркера доступа экземпляр мобильного приложения становится связанным с учетной записью пользователя. Рекомендуется, чтобы мобильное приложение предложило пользователю установить ПИН код или включить Touch ID/Face ID.

Также с помощью полученного маркера доступа приложение может *запросить данные о пользователе* (страница 425).

Если код авторизации был уже использован, не совпал `redirect_uri` с ранее использованным в вызове к `/oauth/ae`, или истек срок действия кода, либо переданный `code_verifier` не соответствует `code_challenge`, то в качестве ответа будет возвращена ошибка.

Пример ответа с ошибкой:

```
{
    "error": "invalid_grant",
    "error_description": "The provided authorization grant… is invalid, expired,␣
↪revoked…"
}
```

Возможные ошибки при вызове `/oauth/te` соответствуют RFC 6749 и описаны здесь[101].

Повторный вход пользователя

При каждом входе пользователя в экземпляр мобильного приложения, если с устройства доступен выход в сеть Интернет, следует производить аутентификацию пользователя посредством вызова сервиса Blitz Identity Provider. В частности, при каждом входе в экземпляр мобильного приложения необходимо проверить ПИН-код пользователя или Touch ID/Face ID, после чего извлечь защищенно хранимые на устройстве `client_id` / `client_secret` и сделать запрос в Blitz Identity Provider на проведение повторного входа пользователя. Использовать полученный в ответ от Blitz Identity Provider маркер доступа для получения актуальных данных пользователя.

Запрос в Blitz Identity Provider на проведение повторного входа должен быть выполнен методом POST на URL для получения маркера (`/oauth/te`). Запрос должен содержать заголовок `Authorization` со значением `Basic {secret}`, где `secret` – это `client_id:client_secret` экземпляра мобильного приложения в формате Base64.

Тело запроса должно содержать параметры:

- `grant_type` – значение `client_credentials`;
- `scope` – перечень запрашиваемых экземпляром мобильного приложения разрешений.

Пример запроса:

```
POST /blitz/oauth/te HTTP/1.1
Authorization: Basic cG9ydGFsLmlhc2l1LmxvY2FsOnBvcnRhbC5pYXNpdS5sb2NhbA==
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials&scope=profile
```

В ответ возвращается маркер доступа и информация об этом маркере.

[101] https://tools.ietf.org/html/rfc6749#section-5.2

Пример ответа с успешным выполнением запроса:

```
{
    "access_token": "dO-xym…BE",
    "expires_in": 3600,
    "scope": "openid profile",
    "token_type": "Bearer"
}
```

Используя полученный маркер доступа, экземпляр мобильного приложения может *запросить* (страница 425) актуальные данные пользователя из Blitz Identity Provider, чтобы при необходимости визуализировать или обновить эти данные в устройстве.

Если пользователь в Blitz Identity Provider отозвал у экземпляра мобильного приложения право авторизации в Blitz Identity Provider, то в результате вызова Blitz Identity Provider экземпляр мобильного приложения получит ошибку.

Пример ответа с ошибкой:

```
{
    "error": "invalid_client",
    "error_description": "Client authentication failed…"
}
```

Возможные ошибки при вызове `/oauth/te` соответствуют RFC 6749 и описаны здесь[102].

Переключение или выход пользователя

Если в мобильном приложении предусмотрена функция выхода или смены пользователя, то при вызове пользователем такой функции мобильное приложение должно также вызвать Blitz Identity Provider и удалить выпущенную для данного экземпляра мобильного приложения пару `client_id` / `client_secret`. Если это не будет сделано, то при выходе пользователя из мобильного приложения, пользователь в веб-приложении Blitz Identity Provider *Настройки безопасности* все равно будет видеть, что мобильное приложение все еще привязано к его учетной записи.

Примечание: Стандартный адрес имеет вид: `https://login.company.com/blitz/profile`.

Чтобы удалить из Blitz Identity Provider выпущенную для экземпляра мобильного приложения пару `client_id`/ `client_secret`, мобильное приложение должно отправить в Blitz Identity Provider запрос методом DELETE на URL управления конфигурацией (`registration_client_uri`), полученный и запомненный мобильным приложением при *вызове динамической регистрации* (страница 410) в Blitz Identity Provider экземпляра мобильного приложения. Запрос должен содержать заголовок `Authorization` со значением `Bearer {registration_access_token}`, где `registration_access_token` – это маркер управления конфигурацией, также полученный и запомненный в процессе динамической регистрации. Запрос не требует указания параметров.

Пример запроса:

```
DELETE /blitz/oauth/register/dyn~CSI~4e6904c5-ef29-4ae5-8d30-99c359b8270f HTTP/1.1
Authorization: Bearer eyJ0e…tw
```

Если после удаления пары `client_id` / `client_secret` мобильное приложение сразу запросит получение новой пары `client_id` / `client_secret`, и запросит вход пользователя, то если предыдущий вход выполнялся в этой же браузерной сессии, то сработает SSO и пользователь автоматически войдет прежним аккаунтом. Обычно это нежелательное поведение для входа сразу после выхода, так как ожидается, что пользователь захочет войти под другим аккаунтом. Поэтому после выхода рекомендуется запрашивать новый вход одним из следующих способов:

[102] https://tools.ietf.org/html/rfc6749#section-5.2

- При запросе кода авторизации указывать в запросе дополнительный параметр `prompt=login`. Тогда Blitz Identity Provider предложит текущему пользователю пройти аутентификацию, даже если активна Blitz Identity Provider сессия. Также пользователь может на странице входа выбрать *Сменить аккаунт*, чтобы войти под другой учетной записью.
- При запросе кода авторизации указать в запросе дополнительный параметр `prompt=select_account`. Так Blitz Identity Provider сразу предложит пользователю выбрать аккаунт из числа запомненных или войти новым аккаунтом. Пользователю не придется дополнительно нажимать кнопку *Сменить аккаунт* на странице входа.

Открытие веб-ресурсов из приложения

В некоторых мобильных приложениях разработчикам может потребоваться предусмотреть функцию открытия веб-ресурсов, также требующих идентификации/аутентификации пользователя, и использующих для этой цели Blitz Identity Provider (режим сквозной аутентификации).

При доступе к веб-ресурсу пользователь, вошедший в мобильное приложение, может столкнуться с ситуацией, что Blitz Identity Provider повторно потребует у него пройти идентификацию/аутентификацию в веб-ресурсе в результате запроса соответствующим веб-приложением идентификации/аутентификации пользователя в Blitz Identity Provider. Чтобы такого не произошло, мобильное приложение может непосредственно перед вызовом веб-ресурса запросить в Blitz Identity Provider получение маркера доступа (`access_token`) на специальное разрешение (`scope`) с именем `native`.

Получить маркер доступа можно способом, описанным в *Повторный вход пользователя* (страница 414) или *Получение маркеров* (страница 395) (при наличии у приложения `refresh_token`).

Пример запроса:

```
POST /blitz/oauth/te HTTP/1.1
Authorization: Basic cG9ydGFsLmlhc2l1LmxvY2FsOnBvcnRhbC5pYXNpdS5sb2NhbA==
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials&scope=native
```

В ответ возвращается не только маркер доступа и информация об этом маркере, но и специальный атрибут – маркер сквозного входа `css` (`cookie short session`).

Пример ответа с получением атрибута `css`:

```
{
    "access_token": "dO-xym…BE",
     "css": "nUngQ…LA",
    "expires_in": 3600,
    "scope": "native",
    "token_type": "Bearer"
}
```

После этого мобильное приложение может открывать веб-ресурс. При этом в запускаемом веб-браузере мобильное приложение должно предварительно установить cookie со следующими параметрами:

- имя cookie – `css`;
- домен cookie – `login.company.com`;
- путь (path) cookie – `/blitz`;
- флаги `HTTPOnly=true` и `Secure=true`;
- значение cookie – значение, полученное в параметре `css` при получении от Blitz Identity Provider маркера доступа на scope с именем `native`.

Если запущенный веб-ресурс в течение 300 секунд с момента запуска инициирует в Blitz Identity Provider идентификацию/аутентификацию, и cookie была корректно установлена, то Blitz Identity Provider по запросу веб-приложения проведет автоматическую сквозную идентификацию и аутентификацию пользователя под учетной записью, с которой пользователь ранее входил в экземпляр мобильного приложения, вызвавшего веб-ресурс.

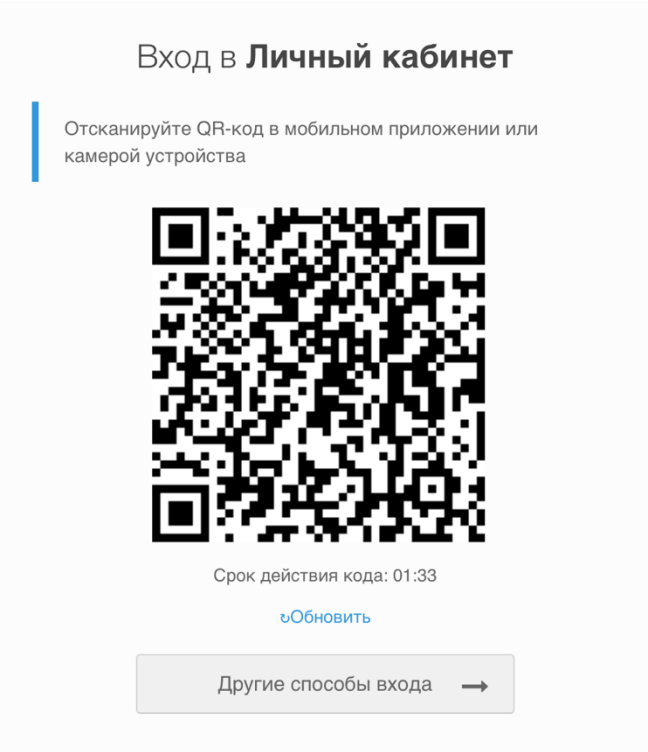
Вход в приложение по QR-коду

Вход по QR-коду может использоваться в Blitz Identity Provider как первый фактор аутентификации (альтернатива вводу логина/пароля). При выборе этого способа входа Blitz Identity Provider формирует и отображает пользователю QR-код, в котором закодирован запрос на вход (Рисунок 6). Срок действия QR-кода ограничен, а сформированный запрос является одноразовым. По истечение срока действия отображенного QR-кода пользователю предоставляется возможность запросить отображение нового QR-кода.

Закодированная в QR-коде ссылка имеет вид: `QR_URL?code=b0671081-cb73-4839-8bc1-8cf020457228`, например:

```
https://login.company.com/blitz/login/qr?code=b0671081-cb73-4839-8bc1-8cf020457228
```

Значение QR_URL может быть настроено таким образом, чтобы в случае наведения смартфона на QR-код с использованием стандартного приложения камеры пользователю могла быть отображена веб-страница с инструкцией по получению правильного мобильного приложения для загрузки QR-кодов или возможность вызова подходящего мобильного приложения через Universal Link.



Процесс входа по QR-коду на стороне мобильного приложения состоит из следующих шагов:

1. Перед фотографированием QR-кода мобильным приложением пользователь должен быть залогинен в мобильное приложение с использованием Blitz Identity Provider, и мобильное приложение должно получить в Blitz Identity Provider действующий маркер доступа со scope с именем `blitz_qr_auth` (разрешение на проведение входа с использованием QR-кода).
2. При фотографировании QR-кода мобильное приложение должно отбросить значение QR_URL (оно не нужно приложению и должно быть проигнорировано) и приложение должно считать значение переданного в ссылке параметра `code`.

3. После считывания QR-кода мобильное приложение должно вызвать в Blitz Identity Provider сервис получения сведений о запросе входа, передав в сервис значение полученного кода, а также заголовок с маркером доступа и заголовок текущего языка пользователя.

Пример вызова:

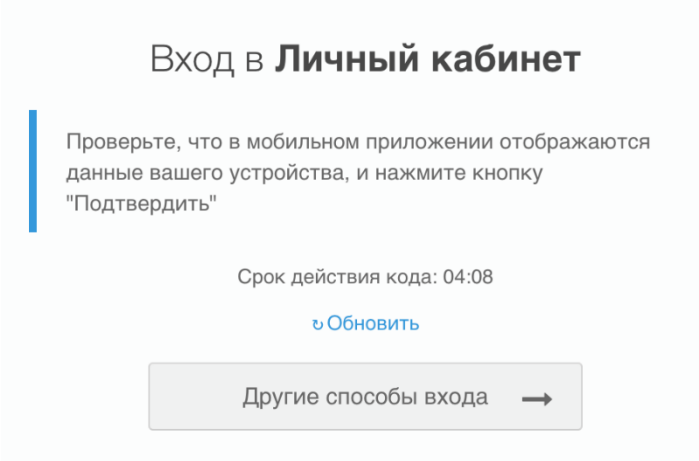
```
curl --location --request GET 'https://login.company.com/blitz/api/v3/auth/qr/
↪b0671081-cb73-4839-8bc1-8cf020457228' \
--header 'Accept-Language: ru' \
--header 'Authorization: Bearer eyJhb…tA'
```

В ответ вернется JSON, содержащий информацию об IP-адресе, операционной системе и браузере устройства, на котором пользователь пытается войти с использованием входа по QR-коду, а также имя приложения, в которое пользователь пытается войти.

Пример успешного ответа:

```
{
    "ip": "83.220.238.103",
    "rp_name": "User profile",
    "ip_city": "Москва",
    "browser": "Chrome 109",
    "ip_state": "Москва",
    "os": "macOS 10.15.7",
    "ip_lng": "37.6171",
    "device_type": "pc",
    "ip_lat": "55.7483",
    "ip_country": "Россия",
    "rp_id": "\_blitz_profile",
    "device_name": "macOS Big Sur (11)",
    "ip_radius": "20",
    "device": "PC"
}
```

Также пользователю в веб-странице будет показан экран, что ожидается подтверждение входа.



Пользователю в мобильном приложении нужно отобразить имя приложения (`rp_name`), IP-адрес (`ip`), геоданные (`ip_country`, `ip_state`, `ip_city` – текстовое описание адреса или показать на карте по координатам `ip_lat`, `ip_lng`), используемое устройство (`device_name`), браузер (`browser`).

Возможные значения `device_type` сейчас: `kindle`, `mobile`, `tablet`, `iphone`, `windowsPhone`, `pc`, `ipad`, `playStation`, `unknown`. Можно их использовать в визуализации сообщения или можно просто вывести имя устройства текстовой строкой из device.

Пример ответа при недействительном маркере доступа:

```
{
    "type": "security_error",
    "error": "bad_access_token",
    "desc": "expired_access_token"
}
```

Пример ответа при просроченном QR-коде:

```
{
    "type": "process_error",
    "error": "qr_session_expired",
    "desc": "Error while getting QR authentication session"
}
```

Пример ответа при несуществующем коде:

```
{
    "params": {},
    "desc": "Error while getting QR authentication session",
    "error": "qr_session_not_found"
}
```

Пример ответа при вызове по уже использованной QR-сессии (когда уже подтвердили или уже отклонили вход):

```
{
    "type": "process_error",
    "error": "qr_session_already_completed",
    "desc": "Error while getting QR authentication session"
}
```

1. Мобильное приложение должно отобразить пользователю полученные из JSON от Blitz Identity Provider сведения о входе, а также выбор действия: «Разрешить» или «Отклонить». В случае «Отклонить» запросить причину отклонения («Вход вызван по ошибке» или «Я не запрашивал вход»).
2. В зависимости от решения пользователя мобильное приложение должно вызвать в Blitz Identity Provider сервис подтверждения или отказа входа. При вызове должен использоваться маркер доступа со scope с именем `blitz_qr_auth`.

Пример вызова при подтверждении входа:

```
curl --location --request POST 'https://login.company.com/blitz/api/v3/auth/qr/
↪5e20b01e-5c7c-4101-8292-98e6865c7bfb/confirm' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer eyJhb…cQ'
```

Если успешно, то вернется `HTTP 204 No Content` без body. Также пользователь войдет в приложение.

Если код просрочен, то вернется:

```
{
    "type": "process_error",
    "error": "qr_session_expired",
    "desc": "Error while confirming QR authentication session"
}
```

Если код не существует, то вернется:

```
{
    "params": {},
    "desc": "Error while confirming QR authentication session",
    "error": "qr_session_not_found"
}
```

Пример ответа при недействительном маркере доступа:

```
{
    "type": "security_error",
    "error": "bad_access_token",
    "desc": "expired_access_token"
}
```

Пример ответа при вызове по уже использованной QR-сессии (когда уже подтвердили или уже отклонили вход):

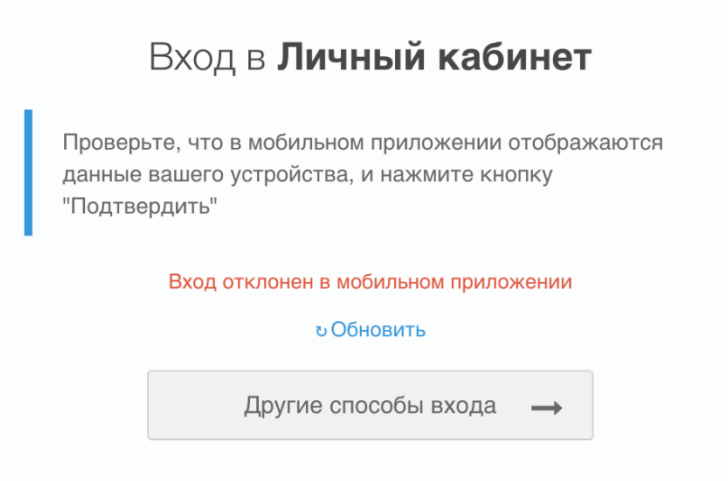
```
{
    "type": "process_error",
    "error": "qr_session_already_completed",
    "desc": "Error while getting QR authentication session"
}
```

Пример вызова при отклонении входа:

```
curl --location --request POST 'https://login.company.com/blitz/api/v3/auth/qr/
↪845f2334-fa6b-40c0-9a71-f57997166e39/refuse' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer eyJhb…bQ' \
--data-raw '{
"cause_id": "mistake",
"desc": "Вход вызван по ошибке"
}'
```

При отклонении входа нужно обязательно передавать в теле запроса JSON с атрибутом `cause_id`. Рекомендуется при отклонении входа пользователем спросить причину. Если пользователь сообщит, что «передумал» (или «вызвал вход по ошибке»), то заполнить `cause_id=mistake`. Но если пользователь сообщит, что он не инициировал вход, то заполнить `cause_id=unauthorized`. Параметр `desc` опционален – можно указать любую текстовую строку.

В случае успешного вызова вернется `HTTP 204 No Content` без body. Также пользователю будет показан экран с ошибкой:



В случае если код просрочен, то вернется ошибка:

```
{
    "type": "process_error",
    "error": "qr_session_expired",
    "desc": "Error while refusing QR authentication session"
}
```

Если код не существует, то вернется:

```
{
    "params": {},
    "desc": "Error while refusing QR authentication session",
    "error": "qr_session_not_found"
}
```

Пример ответа при недействительном маркере доступа:

```
{
    "type": "security_error",
    "error": "bad_access_token",
    "desc": "expired_access_token"
}
```

Пример ответа при вызове по уже использованной QR-сессии (когда уже подтвердили или уже отклонили вход):

```
{
    "type": "process_error",
    "error": "qr_session_already_completed",
    "desc": "Error while getting QR authentication session"
}
```

3.2.4 Подключение приложений умных устройств (IoT)

Общие сведения

В Blitz Identity Provider реализована поддержка возможности авторизации приложений умных устройств (приложений голосовых помощников, Smart TV, чат-ботов) с использованием учетной записи пользователя на другом устройстве. Для такой авторизации используется спецификация RFC 8628 OAuth 2.0 Device Authorization Grant[103].

Настройки подключения

В целях взаимодействия с Blitz Identity Provider приложение должно использовать следующие адреса:

- URL для получения кода подтверждения авторизации (OAuth 2.0 Device Authorization Grant):
  - `https://login-test.company.com/blitz/oauth/da` (тестовая среда)
  - `https://login.company.com/blitz/oauth/da` (продуктивная среда)
- URL для получения и обновления маркера доступа:
  - `https://login-test.company.com/blitz/oauth/te` (тестовая среда)
  - `https://login.company.com/blitz/oauth/te` (продуктивная среда)

[103] https://www.ietf.org/rfc/rfc8628.html

- URL для получения данных пользователя:
  - `https://login-test.company.com/blitz/oauth/me` (тестовая среда)
  - `https://login.company.com/blitz/oauth/me` (продуктивная среда)
- URL для получения данных о маркере доступа:
  - `https://login-test.company.com/blitz/oauth/introspect` (тестовая среда)
  - `https://login.company.com/blitz/oauth/introspect` (продуктивная среда)
- URL для выполнения логаута:
  - `https://login-test.company.com/blitz/oauth/logout` (тестовая среда)
  - `https://login.company.com/blitz/oauth/logout` (продуктивная среда)

Все эти URL, а также дополнительные сведения, размещены по адресу динамически обновляемых настроек (метаданных) каждой среды Blitz Identity Provider:

Совет: См. RFC 8414 OAuth 2.0 Authorization Server Metadata[104].

- `https://login-test.company.com/blitz/.well-known/openid-configuration` (тестовая среда)
- `https://login.company.com/blitz/.well-known/openid-configuration` (продуктивная среда)

Разработчики приложений могут не прописывать все указанные URL в конфигурации своего приложения, а использовать в настойках единую ссылку на метаданные Blitz Identity Provider.

Получение кода авторизации

Для инициирования авторизации приложение умного устройства должно сделать запрос в адрес Blitz Identity Provider на сервис получения кода подтверждения авторизации (`/oauth/da`). Запрос должен быть сделан методом POST. Запрос должен содержать заголовок `Authorization` со значением `Basic {secret}`, где `secret` – это `client_id:client_secret` (например, `app:topsecret`) в формате Base64.

Пример заголовка:

```
Authorization: Basic ZHluOkNTSTo…dx
```

Тело запроса должно содержать следующие параметры:

- `client_id` – идентификатор приложения;
- `scope` – запрашиваемые разрешения.

Пример запроса:

```
POST /blitz/oauth/da HTTP/1.1
Authorization: Basic ZHluOkNTSTo…dx
Content-Type: application/x-www-form-urlencoded

client_id=test-app&scope=profile
```

В ответ Blitz Identity Provider вернет данные, необходимые для подтверждения входа на другом устройстве:

- `device_code` – код устройства;
- `user_code` – отображаемый пользователю код подтверждения запроса авторизации;

[104] https://tools.ietf.org/html/rfc8414

- `verification_uri` – ссылка на страницу, на которой пользователь может ввести код подтверждения запроса авторизации;
- `verification_uri_complete` – ссылка на страницу, в которой в качестве параметра уже подставлен код подтверждения запроса авторизации;
- `expires_in` – время жизни пользовательского кода в секундах;
- `interval` – рекомендуемый период ожидания в секундах при опрашивании приложением ввода пользователем кода подтверждения запроса авторизации.

Пример ответа с успешным выполнением запроса:

```
{
    "device_code": "7Lz30lK57bWaKHBYxM8kW7KpOFvDg_4ujz3LpQxcleE",
    "user_code": "934-367-578",
    "verification_uri": "https://device.company.com",
    "verification_uri_complete": "https://device.company.com?uc=934-367-578",
    "expires_in": 300,
    "interval": 5
}
```

Получив ответ приложение умного устройства должно инструктировать пользователя, чтобы он перешел по ссылке `verification_uri` и ввел код из `user_code`.

Примечание: Ссылка в `verification_uri` выводится в соответствии с настройками, заданными в Blitz Identity Provider. Рекомендуется настроить, чтобы эта ссылка была короткой и удобной для ввода пользователям, а также хорошо воспринималась на слух или красиво отображалась на экране Смарт ТВ. С данной ссылки должна быть настроена переадресация на обработчик ввода пользователем кода подтверждения, расположенный на странице `https://login.company.com/blitz/oauth/device?ci=client_id`, где вместо `client_id` нужно задать идентификатор зарегистрированного в Blitz Identity Provider приложения, из настроек которого будут браться разрешенные способы входа и настройки внешнего вида страницы входа.

В зависимости от типа умного устройства нужно выбрать наиболее удобный для пользователя способ. Например:

- При авторизации в Smart TV приложение может отрисовать пользователю QR-код, в котором закодировать ссылку из `verification_uri_complete`. Тогда пользователю нужно будет навести камеру телефона на QR-код и пройти авторизацию на телефоне.
- При авторизации в чат-боте приложение может отрисовать пользователю кнопку, открывающую в браузере ссылку из `verification_uri_complete`. Тогда пользователю нужно будет пройти авторизацию в браузере своего устройства.
- При авторизации в приложении голосового помощника приложение может проинструктировать пользователя, на какой сайт он должен перейти, и озвучить код, который пользователь должен ввести, либо приложение может отправить пользователю SMS-сообщение или письмо по электронной почте с инструкцией.

Получение маркера безопасности

После предоставления пользователю инструкций приложение умного устройства должно с интервалом из параметра `interval` начать осуществлять опрос Blitz Identity Provider для получения маркеров безопасности. Для этого приложение должно обращаться в Blitz Identity Provider методом POST на URL для получения маркера (`/oauth/te`). Запрос должен содержать заголовок `Authorization` со значением `Basic {secret}`, где `secret` – это `client_id:client_secret` экземпляра мобильного приложения в формате Base64.

Тело запроса должно содержать параметры:

- `grant_type` – значение `urn:ietf:params:oauth:grant-type:device_code`;
- `device_code` – ранее полученный код устройства.

Пример запроса:

```
POST /blitz/oauth/te HTTP/1.1
Authorization: Basic cG9…A==
Content-Type: application/x-www-form-urlencoded

grant_type=urn:ietf:params:oauth:grant-type:device_code&device_code=Yrn…\_0
```

Если пользователь еще не подтвердил авторизацию, то Blitz Identity Provider вернет следующий ответ с ошибкой:

```
{
    "error": "authorization_pending",
    "error_description": "The authorization request is still pending"
}
```

Если срок действия пользовательского кода истек или код неправильный, то Blitz Identity Provider вернет следующий ответ с ошибкой:

```
{
    "error": "invalid_grant",
    "error_description": "The provided authorization grant (e.g., authorization␣
↪code, resource owner credentials) or refresh token is invalid, expired, revoked,␣
↪does not match the redirection URI used in the authorization request, or was␣
↪issued to another client."
}
```

Если пользователь подтвердил авторизацию, то Blitz Identity Provider вернет приложению маркер доступа и информацию о нем, а также маркер обновления.

Пример ответа с успешным выполнением запроса:

```
{
    "access_token": "eyJ…tA",
    "refresh_token": "wVE…cw",
    "scope": "profile",
    "token_type": "Bearer",
    "expires_in": 3600
}
```

Используя полученный маркер доступа, приложение умного устройства может *запросить* (страница 425) актуальные данные пользователя из Blitz Identity Provider.

3.2.5 Получение атрибутов пользователя

Для запроса данных о пользователе необходимо выполнить запрос методом GET по URL-адресу получения данных пользователя (`/oauth/me`). В запрос должен быть добавлен следующий заголовок:

```
Authorization: Bearer <access token>
```

В заголовке `<access token>` – это маркер доступа, полученный от Blitz Identity Provider (см. *Получение маркеров* (страница 395) и *Получение маркеров экземпляром приложения* (страница 413)).

Пример запроса:

```
GET /blitz/oauth/me HTTP/1.1
Authorization: Bearer NINxn…tY
Cache-Control: no-cache
```

В ответе будут отображены только те данные, которые *определены в scope* (страница 387), на который получен маркер доступа.

Пример ответа:

```
{
    "family_name": "Иванов",
    "given_name": "Иван",
    "middle_name": "Иванович",
    "email": "iivanov@company.com",
    "phone_number": "79162628910",
    "sub": "3d10f626-ea77-481d-a50bd4a4d432d86b"
}
```

Учетная запись пользователя может быть включена в группы пользователей. Чтобы получить список групп, в которые включен пользователь, маркер доступа должен быть получен с scope с именем `usr_grps`.

Пример ответа по пользователю, включенному в группы доступа:

```
{
    "family_name": "Иванов",
    "given_name": "Иван",
    "middle_name": "Иванович",
    "email": "iivanov@company.com",
    "phone_number": "79162628910",
    "sub": "3d10f626-ea77-481d-a50bd4a4d432d86b",
    "groups": [
        {
            "id": "564486ff-af0a-3fb1-3f09-e7c5f7f9833e",
            "name": "Тестовая организация",
            "OGRN": "1234567890123",
            "INN": "9876543210"
        }
    ]
}
```

3.2.6 Обеспечение безопасности подключения

Оператором приложения, подключенного к Blitz Identity Provider, должно обеспечиваться соблюдение следующих требований к безопасности:

1. Должна обеспечиваться конфиденциальность полученного для приложения при регистрации в Blitz Identity Provider значения `client_secret`:
    - Запрещается предавать значение `client_secret` лицам, не связанным с обеспечением эксплуатации приложения.
    - Запрещается использовать `client_secret` в клиентской части ПО (код, выполняемый на стороне браузера, мобильного приложения, десктопного приложения). Применяться `client_secret` должен только в серверных компонентах приложения. Исключение – `client_secret`, полученный мобильным или десктопным приложением с помощью операции динамической регистрации, такой `client_secret` можно хранить и обрабатывать в мобильном или десктопном приложении.
    - В случае если `client_secret` скомпрометирован, то должна быть подана заявка на замену `client_secret` приложения. В Blitz Identity Provider предусмотрена возможность «плавной замены» `client_secret`, а именно, приложению может быть присвоен дополнительный `client_secret` на время, пока будет выполняться перенастройка приложения с прежнего на новое значение `client_secret`.
2. Должна обеспечиваться конфиденциальность полученных приложением от Blitz Identity Provider маркеров доступа (`access_token`) и маркеров обновления (`refresh_token`).
    - Нужно избегать использования маркеров доступа в браузерной части приложения. Если все-таки это необходимо (SPA-приложение), то использующий маркер доступа JS-код должен предусматривать защиту от возможности получения значения маркера доступа из браузерной консоли.
    - Запрещено хранить/обрабатывать маркер обновления на стороне браузерной части приложения – маркер обновления должен использоваться исключительно в серверных компонентах приложения. При хранении маркеров обновления в приложении (в БД, файлах и т.д.) доступ к хранимым маркерам обновления должен быть ограничен.
3. Взаимодействие приложения с Blitz Identity Provider в продуктивном контуре должно осуществляться исключительно с использованием защищенного соединения (HTTPS). Запрещено использовать HTTP в обработчиках приложения (адреса возврата `redirect_uri`, `post_logout_redirect_uri`).
4. Приложению запрещено открывать страницу входа Blitz Identity Provider во фрейме.
5. При подключении мобильных приложений к Blitz Identity Provider:
    - использованием PKCE является обязательным;
    - запрещено использовать Embedded-браузер.

3.3 Интеграция приложения по SAML

3.3.1 Как правильно зарегистрировать приложение

Аутентификация в терминологии SAML является результатом взаимодействия трех сторон:

- поставщик идентификации (`Identity Provider`), в качестве которого выступает Blitz Identity Provider;
- поставщик услуги (`Service Provider`), в качестве которого выступает подключаемое приложение;
- веб-браузер пользователя (`User Agent`).

Первым шагом при подключении приложения является его *регистрация* (страница 238) в качестве поставщика услуг в Blitz Identity Provider. Нужно предварительно подготовить XML-файл с метаданными поставщика услуг или значения параметров, необходимые для самостоятельной подготовки метаданных.

Метаданные поставщика услуг описывают настройки подключения приложения к Blitz Identity Provider (например, URL конечных точек приложения, ключи для проверки ЭП). Для описания метаданных используется язык XML.

Совет: См. подробнее про метаданные SAML[105].

Внимание: Метаданные должны быть подготовлены по результатам выполнения работ по *добавлению поддержки протокола* (страница 428).

Если приложение является готовым ПО, поддерживающим SAML, то метаданные должны быть получены согласно документации на это ПО. Обычно такое ПО предоставляет URL, по которому может быть получены метаданные.

Если ПО подключаемого приложения не предусматривает выгрузку метаданных, но в документации на ПО описаны параметры, которые должны быть настроены для подключения приложения, то можно указать эти параметры, так, чтобы метаданные на их основе были самостоятельно подготовлены Администратором Blitz Identity Provider.

В этом случае необходимо указать следующие параметры:

1. Идентификатор поставщика услуг (`entityID`) – следует указать, только если приложению необходим конкретный `entityID`. Иначе `entityID` будет самостоятельно присвоен Администратором Blitz Identity Provider.
2. Сертификат открытого ключа приложения (поставщика услуг) – должен быть указан только в случае, если приложение подписывает SAML-запрос при отправке к Blitz Identity Provider.

   Примечание: Сертификат поставщика услуг отличается от TLS-сертификата подключаемого веб-сайта. Обычно это самоподписанный сертификат с длительным сроком действия.

   Важно: Должны использоваться ключи RSA-2048.

   Примечание: Допустимо использовать самоподписанные сертификаты с длительным сроком действия.

3. URL для приема от Blitz Identity Provider SAML-ответа – приложение должно предоставлять обработчик, осуществляющий прием от Blitz Identity Provider SAML-ответов с результатами входа. Обычно эта настройка приложения называется `Assertion Consumer Service`.
4. URL для приема от Blitz Identity Provider запроса на логаут – выборочная настройка. Если приложение поддерживает единый логаут, то оно может предоставлять обработчик единого логаута. Обычно эта настройка приложения называется `Single Logout Service Location`.
5. URL для перенаправления пользователя в приложение после успешного логаута – опциональная настройка. Если приложение поддерживает единый логаут и может инициировать единый выход, то оно может предоставлять URL для возврата пользователя после логаута. Обычно эта настройка приложения называется `Single Logout Service Response Location`.
6. Перечень запрашиваемых атрибутов (`SAML Assertion`).

[105] https://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

Доступные атрибуты пользователя

Атрибут	Описание
`logonname`	Логин пользователя в домене
`surname`	Фамилия
`firstname`	Имя
`middlename`	Отчество
`email`	Служебный адрес электронной почты

7. Признак необходимости передачи атрибутов в зашифрованном виде.

> **Примечание:** Атрибуты в SAML-сообщении всегда передаются подписанными. Включать шифрование атрибута целесообразно, если пользователь не должен иметь возможности прочитать значение атрибута.

3.3.2 Подключение приложения по SAML

Данные для подключения

Для подключения приложения к Blitz Identity Provider потребуются данные, полученные в ходе его *регистрации в продукте* (страница 426):

- идентификатор, присвоенный приложению в Blitz Identity Provider (`entityID`);
- файл метаданных поставщика услуг.

Приложение взаимодействует с сервисами Blitz Identity Provider, используя следующие адреса:

- метаданные Blitz Identity Provider:
  - `https://login-test.company.com/blitz/saml/profile/Metadata/SAML` (тестовая среда)
  - `https://login.company.com/blitz/saml/profile/Metadata/SAML` (продуктивная среда)
- URL для аутентификации:
  - `https://login-test.company.com/blitz/saml/profile/SAML2/Redirect/SSO` (тестовая среда)
  - `https://login.company.com/blitz/saml/profile/SAML2/Redirect/SSO` (продуктивная среда)
- URL для логаута:
  - `https://login-test.company.com/blitz/saml/profile/SAML2/Redirect/SLO` (тестовая среда)
  - `https://login.company.com/blitz/saml/profile/SAML2/Redirect/SLO` (продуктивная среда)
- URL издателя:
  - `https://login-test.company.com/blitz/saml/` (тестовая среда)
  - `https://login.company.com/blitz/saml/` (продуктивная среда)

Если приложение поддерживает протокол подключения SAML, то указанных данных должно быть достаточно для конфигурирования приложения. Если приложение не поддерживает протокол SAML, следует произвести его доработку согласно рекомендациям, изложенным в разделах *Готовые библиотеки* (страница 430) и *Принцип интеграции* (страница 431).

Типичные вопросы о том, как настроить приложение для подключения к Blitz Identity Provider по протоколу SAML:

Где найти метаданные поставщика идентификации?

Чтобы загрузить метаданные, пройдите по ссылке `https://login.company.com/blitz/saml/profile/Metadata/SAML` и скопируйте открытый XML документ в приложение.

Где найти сертификат SAML поставщика идентификации?

Откройте XML документ с метаданными поставщика идентификации. Найдите раздел `<ds:X509Certificate></ds:X509Certificate>` – в нем и располагается сертификат SAML поставщика идентификации. Пример:

```
▼<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
 xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:shibmd="urn:mace:shibboleth:metadata:1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" entityID="https://sudir.mos.ru/blitz/saml">
  ▼<IDPSSODescriptor protocolSupportEnumeration="urn:mace:shibboleth:1.0
   urn:oasis:names:tc:SAML:1.1:protocol urn:oasis:names:tc:SAML:2.0:protocol">
    ▼<Extensions>
       <shibmd:Scope regexp="false">0.1</shibmd:Scope>
     </Extensions>
    ▼<KeyDescriptor>
      ▼<ds:KeyInfo>
        ▼<ds:X509Data>
          ▼<ds:X509Certificate>
             MIIDDzCCAfegAwIBAgIJANjxtiKgDpaeMA0GCSqGSIb3DQEBBQUAMBcxFTATBgNV
             BAMTDHN1ZGlyLm1vcy5ydTAeFw0xODA2MjAxNjQ2MDZaFw0yODA2MTcxNjQ2MDZa
             MBcxFTATBgNVBAMTDHN1ZGlyLm1vcy5ydTCCASIwDQYJKoZIhvcNAQEBBQADggEP
             ADCCAQoCggEBANk5Ue/3dmNTLdTzKNrgKLM7lpdnBFNJNjDkkkBF2GodQ+r+ePLz
             thw5Gn9G4uLmwFol13fU6usbEdi2IDzg3M5slT8YbCxzvaw7ddNU9Jdh1YAqIrXT
             VvtRCajyZk3AwraXNj1Ai9Qq8XuXSlEtlymvdUAeY1SScKDpNYIM8cqdHmvSXKvx
             FggJn+S1l6MEDv/0quM2MvOhgLuP7i6J8wNXD4P4fz8+oNGPcqLwn90fIGgFyPBE
             nQ2vmEn0NRotwQCnYcIApEq9jMBGiMi2yQtIsjFYDjjdqBqau/cXuVybiYA8om3W
             cyMIDFdcJ2RAAHtzNdXN8xnnv8IMrqRqG/MCAwEAAaNeMFwwOwYDVR0RBDQwMoIw
             c3VkaXIubW9zLnJ1O1VSSTpodHRwczovL3N1ZGlyLm1vcy5ydS9ibGl0ei9zYW1s
             MB0GA1UdDgQWBBRw3ACqmoCP31aMlW/KtwFsQLZ7iDANBgkqhkiG9w0BAQUFAAOC
             AQEAJ72xDGx37QBdHIyDiOhwe1Kxibvwm5DZxQ6Sc6YTS6fnCWdJeUlLJ82yKOIW
             Hwfnre+nRRuAHLA9DhaZIYmBvUuqE1tBYadwqiKS0l518khE509jnMyizWMiwRPK
             IUz730BQUDl3zsT+WvO21Xced8PKR73Y2XZCnIybDbYNipy1ST9V0/bkB1S6VR8x
             OOiOr89rgY/1EWXRnQn+9Wm2tQZXbdCTHOBg7kCg4M4OnqyOi1rfUvoHboeVrLUA
             ap/b+fHRdL2p08qCJOSCRhPwETuyYolqt3DSYJqqTDui1Tyg8i61j65xL0lJER9J
             48L3KzS5SY/DUHYmfLfddIRb/Q==
          </ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </KeyDescriptor>
    <ArtifactResolutionService Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
    Location="https://sudir.mos.ru/blitz/saml/profile/SAML1/SOAP/ArtifactResolution" index="1"/>
    <ArtifactResolutionService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
    Location="https://sudir.mos.ru/blitz/saml/profile/SAML2/SOAP/ArtifactResolution" index="2"/>
    <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
    Location="https://sudir.mos.ru/blitz/saml/profile/SAML2/Redirect/SLO"
    ResponseLocation="https://sudir.mos.ru/saml/profile/SAML2/Redirect/SLO"/>
    <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Plain-Redirect"
    Location="https://sudir.mos.ru/blitz/saml/profile/SAML2/Redirect/Plain/SLO"
    ResponseLocation="https://sudir.mos.ru/saml/profile/SAML2/Redirect/Plain/SLO"/>
    <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
    Location="https://sudir.mos.ru/blitz/saml/profile/SAML2/SOAP/SLO"/>
    <NameIDFormat>urn:mace:shibboleth:1.0:nameIdentifier</NameIDFormat>
```

Иногда для корректной загрузки в приложение перед строкой с сертификатом нужно вставить строку `-----BEGIN CERTIFICATE-----`, а после – `-----END CERTIFICATE-----`

Где найти адреса SAML-обработчиков поставщика идентификации?

Запросы на идентификацию/аутентификацию приложение должно отправлять на следующие обработчики (`SingleSignOnService`) в ПРОД-среде:

- `https://login.company.com/blitz/saml/profile/SAML2/Redirect/SSO` – для приема сжатых с помощью алгоритма Deflate запросов – стандартный SAML-обработчик.
- `https://login.compan y.com/blitz/saml/profile/SAML2/Redirect/Plain/SSO` – для приема несжатых запросов – следует использовать только в случае, если подключаемое приложение не использует `deflate`.

Запросы на единый логаут приложение должно отправлять на следующие обработчики (`SingleLogoutService`) в ПРОД-среде:

- `https://login.company.com/blitz/saml/profile/SAML2/Redirect/SLO` – для приема сжатых с помощью алгоритма Deflate запросов – стандартный SAML-обработчик.
- `https://login.compan y.com/blitz/saml/profile/SAML2/Redirect/Plain/SLO` – для приема несжатых запросов – следует использовать только в случае, если подключаемое приложение не использует `deflate`.

В ТЕСТ-среде аналогичные адреса начинаются с `https://login-test.company.com`.

Какой entity ID у поставщика идентификации?

Blitz Identity Provider как поставщик идентификации имеет следующие entityID:

- Для ПРОД-среды – `https://login.company.com/blitz/saml`
- Для ТЕСТ-среды – `https://login-test.company.com/blitz/saml`

Готовые библиотеки

Так как самостоятельная разработка программного интерфейса клиента SAML является трудоемкой задачей, а ошибки в реализации чреваты угрозами безопасности, при интеграции приложения по SAML рекомендуется использовать существующие популярные библиотеки SAML-клиентов:

- OIOSAML[106] (Java, .NET),
- OpenSAML[107] (Java),
- Spring Security SAML[108] (Java),
- SimpleSAMLphp[109] (PHP),
- ruby-saml[110] (Ruby on Rails).

Далее приводятся ключевые сведения, необходимые для понимания процесса аутентификации по протоколу SAML.

[106] https://digitaliser.dk/group/42063/resources
[107] https://wiki.shibboleth.net/confluence/display/OS30/Home
[108] https://spring.io/projects/spring-security-saml
[109] https://simplesamlphp.org/
[110] https://rubygems.org/gems/ruby-saml/

Принцип интеграции

Для подключения к Blitz Identity Provider в целях идентификации и аутентификации пользователей приложение может использовать стандарт SAML[111] версий 1.0, 1.1, 2.0.

При этом процесс взаимодействия приложения и Blitz Identity Provider должен быть построен в соответствии с профилем SAML Web Browser SSO Profile[112].

Стандарт SAML основан на XML и определяет способы обмена информацией об аутентификации пользователей и их идентификационных данных (атрибуты, полномочия).

Для возможности осуществлять взаимодействия поставщик услуг и поставщик идентификации предварительно должны обменяться настройками взаимодействия, описанными в форме XML-документов и называемых метаданными. Поставщик услуг должен получить настройки Blitz Identity Provider, называемые *метаданными поставщика идентификации* (страница 426).

Идентификация и аутентификация

См. *описание* (страница 233) принципа взаимодействия веб-приложения с Blitz Identity Provider по SAML.

Логаут

Подключенное к Blitz Identity Provider по SAML приложение также может предусматривать возможность реализации единого выхода (логаута). Для этих целей Blitz Identity Provider поддерживает SAML Single Logout Profile[113]. Приложение может направить в Blitz Identity Provider SAML-запрос `<LogoutRequest>` и в случае успешного завершения единого логаута получить от Blitz Identity Provider SAML-ответ `<LogoutResponse>`. Если приложение должно быть задействовано в едином логауте, инициированным другим приложением, подключенным к Blitz Identity Provider, то оно также должно предусматривать возможность обработки запросов `<LogoutRequest>`, поступивших к приложению от Blitz Identity Provider. В случае успешного завершения локальной сессии приложение должно уведомлять Blitz Identity Provider путем отправки ему SAML-ответа `<LogoutResponse>`.

3.4 API управления пользователями

3.4.1 Общие сведения

Версии REST API

В настоящий момент в Blitz Identity Provider доступны следующие версии REST API, различающиеся способом авторизации:

Предупреждение: Сервисы версий `v1` и `v2` после появления аналогов в более новой `v3` будут помечены как устаревшие, и будет рекомендовано перейти с их использования на сервисы `v3`.

- `v1` – REST-сервисы, доступные по адресам:
  - `https://login.company.com/blitz/reg/api/v1/`,
  - `https://login.company.com/blitz/api/v1/`.

[111] http://saml.xml.org/saml-specifications
[112] https://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf
[113] https://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

Для авторизации вызова этих сервисов используется HTTP Basic авторизация. Для приложения, которое будет вызывать REST-сервисы, необходимо в настройках приложения задать пароль на вкладке REST настроек протоколов приложения. Приложению будут доступны все REST-сервисы `v1`.

Совет: Если какие-то из сервисов использовать не планируется, запретите их вызов через настройки веб-сервера (nginx).

- `v2` – REST-сервисы, доступные по адресу `https://login.company.com/blitz/api/v2/`. Для авторизации вызова большинства этих сервисов используется HTTP Basic авторизация, а для части сервисов – OAuth 2.0.
- `v3` – REST-сервисы, доступные по адресу `https://login.company.com/blitz/api/v3/`. Для авторизации вызова этих сервисов используется OAuth 2.0 и полученные от Blitz Identity Provider маркеры безопасности. Доступ приложений к различным REST-сервисам регулируется через разрешения (`scope`).

Режимы доступа к REST API

Предоставляемые Blitz Identity Provider сервисы `https://login.company.com/blitz/api/v3/` можно вызывать в двух режимах:

- пользовательский режим,
- системный режим.

Пользовательский режим доступа

В пользовательском режиме сервис вызывается с правами в отношении учетной записи текущего авторизованного пользователя. При вызове сервиса должны передаваться следующие заголовки:

- `Authorization: Bearer <маркер доступа с пользовательскими разрешениями>` – заголовок авторизации, содержащий маркер доступа с *разрешениями* (страница 432) текущего пользователя.
- `X-Forwarded-For: <IP-адрес пользователя>` – заголовок, в котором должно быть передано значение IP-адреса пользователя. Данное значение будет записано в событие безопасности Blitz Identity Provider.
- `User-Agent: <значение User-Agent пользователя>` – заголовок, в котором должно быть передано значение `User-Agent` устройства пользователя. Данное значение будет записано в событие безопасности Blitz Identity Provider.

Возможные разрешения пользователя

Изменение пароля

`blitz_change_password`

Для использования сервиса `POST /blitz/api/v2/users/{subjectId}/password`.

Создание новых прав

Для использования сервисов:

- `PUT /blitz/admin/api/v3/rights/right7`,
- `GET /blitz/admin/api/v3/rights/right7`,
- `DELETE /blitz/admin/api/v3/rights/right6`.

Управление правами учетной записи

`blitz_user_rights`

Для использования сервисов:

- `GET /blitz/api/v3/rights/of/{subjectId}`,
- `POST /blitz/api/v2/users/rights/change`.

Получение атрибутов

`blitz_api_user`

Для использования сервиса `GET /blitz/api/v3/users/{subjectId}`.

Изменение атрибутов

`blitz_api_user_chg`

Для использования сервиса `POST /blitz/api/v3/users/{instanceId}`.

Получение настроек двухфакторной аутентификации, разрешений, контрольного вопроса

`blitz_api_usec`

Для использования сервисов:

- `GET /blitz/api/v3/users/{subjectId}/auth`,
- `GET /blitz/api/v3/users/{subjectId}/totps`,
- `GET /blitz/api/v3/users/{subjectId}/acls`,
- `GET /blitz/api/v3/users/{subjectId}/secQsn`,
- `POST /blitz/api/v3/users/{subjectId}/secQsn /check`.

Изменение пароля, сброс сессий, изменение контрольного вопроса, настроек двухфакторной аутентификации, отзыв разрешений

`blitz_api_usec_chg`

Для использования сервисов:

- `POST /blitz/api/v3/users/{instanceId}/pswd`,
- `POST /blitz/api/v3/users/{instanceId}/sessions/reset`,
- `POST /blitz/api/v3/users/{instanceId}/secQsn`,
- `POST /blitz/api/v3/users/{subjectId}/auth`,
- `GET /blitz/api/v3/users/{subjectId}/totps /attach/qr`,

- POST /blitz/api/v3/users/{subjectId /totps /attach/qr,
- DELETE /blitz/api/v3/users/{subjectId} /secQsn,
- DELETE /blitz/api/v3/users/{subjectId} /totps/{id},
- DELETE /blitz/api/v3/users/{subjectId} /acls/{id}.

Получение запомненных устройств

blitz_api_uapps

Для использования сервиса GET /blitz/api/v3/users/{subjectId}/apps.

Удаление запомненных устройств

blitz_api_uapps_chg

Для использования сервиса DELETE /blitz/api/v3/users/{subjectId}/apps/{id}.

Получение событий безопасности

blitz_api_uaud

Для использования сервиса GET /blitz/api/v3/users/{subjectId}/audit.

Получение списка учетных записей внешних поставщиков

blitz_api_ufa

Для использования сервиса GET /blitz/api/v3/users/{subjectId}/fa.

Изменение списка учетных записей внешних поставщиков

blitz_api_ufa_chg

Для использования сервисов:

- POST /blitz/api/v3/users/{subject Id}/fa/{fpType}/{fpName}/{sid},
- DELETE /blitz/api/v3/users/{subjectId}/fa/{fpType}/{fpName}/{sid}.

Вход с использованием QR-кода

blitz_qr_auth

Для использования сервисов:

- GET /blitz/api/v3/auth/qr/{QR_code},
- POST /blitz/api/v3/auth/qr/{QR_code}/confirm,
- POST /blitz/api/v3/auth/qr/{QR_code}/refuse.

Маркер доступа на пользовательские разрешения приложение получает в момент идентификации и аутентификации пользователя.

Примечание: Описание механизмов идентификации и аутентификации приведено в разделах:

- *Получение кода авторизации* (страница 390)

- *Получение маркеров* (страница 395)

Системный режим доступа

В данном разделе приведен перечень разрешений, которые может получить приложение для доступа к REST API.

Возможные системные разрешения (разрешения, получаемые на приложение)

Доступ к сервисам работы с организациями

`blitz_groups`

Для использования сервисов:

- `GET /blitz/api/v2/grps/{id}`,
- `POST /blitz/api/v2/grps`,
- `POST /blitz/api/v2/grps/{id}?profile={profile}`,
- `DELETE /blitz/api/v2/grps/{id}?profile={profile}`,
- `GET /blitz/api/v2/grps/{id}/members`,
- `POST /blitz/api/v2/grps/{id}/members/add?profile={profile}`,
- `POST /blitz/api/v2/grps/{id}/members/rm?profile={profile}`.

Назначение и отзыв прав доступа

`blitz_rights_full_access`

Для использования сервисов:

- `PUT /blitz/api/v3/rights`,
- `DELETE /blitz/api/v3/rights`,
- `GET /blitz/api/v3/rights/on`,
- `GET /blitz/api/v3/rights/of`.

Отзыв прав доступа ведомых учетных записей

`blitz_rm_rights`

Для использования сервиса `POST /blitz/api/v2/users/rights/change`.

Получение атрибутов любого пользователя

`blitz_api_sys_users`

Для использования сервиса `GET /blitz/api/v3/users/{subjectId}`.

Изменение атрибутов любого пользователя

`blitz_api_sys_users_chg`

Для использования сервиса `POST /blitz/api/v3/users/{instanceId}`.

Регистрация учетной записи

`blitz_api_sys_users_reg`

Для использования сервиса `PUT /blitz/api/v3/users`.

Получение настроек двухфакторной аутентификации, разрешений любого пользователя, контрольного вопроса

`blitz_api_sys_usec`

Для использования сервисов:

- `GET /blitz/api/v3/users/{subjectId}/auth`,
- `GET /blitz/api/v3/users/{subjectId}/totps`,
- `GET /blitz/api/v3/users/{subjectId}/acls`,
- `GET /blitz/api/v3/users/{subjectId}/state`,
- `GET /blitz/api/v3/users/{subjectId}/secQsn`,
- `POST /blitz/api/v3/users/{subjectId}/secQsn/check`.

Изменение пароля, настроек двухфакторной аутентификации и контрольного вопроса, сброс сессий, отзыв разрешений любого пользователя

`blitz_api_sys_usec_chg`

Для использования сервисов:

- `POST /blitz/api/v3/users/{instanceId}/pswd`,
- `POST /blitz/api/v3/users/{instanceId}/sessions/reset`,
- `POST /blitz/api/v3/users/{subjectId}/auth`,
- `POST /blitz/api/v3/users/{subjectId}/state`,
- `GET /blitz/api/v3/users/{subjectId}/totps/attach/qr`,
- `POST /blitz/api/v3/users/{subjectId}/totps/attach/qr`,
- `POST /blitz/api/v3/users/{subjectId}/secQsn`,
- `DELETE /blitz/api/v3/users/{subjectId}/totps/{id}`,
- `DELETE /blitz/api/v3/users/{subjectId}/acls/{id}`,
- `DELETE /blitz/api/v3/users/{subjectId}/secQsn`.

Получение устройств любого пользователя

`blitz_api_sys_uapps`

Для использования сервиса:

`GET /blitz/api/v3/users/{subjectId}/apps`.

Удаление устройств любого пользователя

`blitz_api_sys_uapps_chg`

Для использования сервиса:

`DELETE /blitz/api/v3/users/{subjectId}/apps/{id}`.

Получение событий безопасности любого пользователя

`blitz_api_sys_uaud`

Для использования сервиса:

`GET /blitz/api/v3/users/{subjectId}/audit`.

Получение списка учетных записей внешних поставщиков

`blitz_api_sys_ufa`

Для использования сервиса:

`POST /blitz/api/v3/users/{subjectId}/fa/{fpType}/{fpName}/{sid}`.

Изменение списка учетных записей внешних поставщиков

`blitz_api_sys_ufa_chg`

Для использования сервиса:

`DELETE /blitz/api/v3/users/{subjectId}/fa/{fpType}/{fpName}/{sid}`.

Получение маркера доступа от любого внешнего поставщика

`fed_tkn_any`

Blitz Identity Provider можно *настроить* (страница 143) таким образом, что будут сохраняться маркеры доступа пользователя от внешних поставщиков идентификации. Разрешение позволяет получить сохраненный маркер доступа любого поставщика.

Для использования сервиса:

`GET /blitz/api/v3/users/${subjectId}/fedToken/${fedPointType}/${fedPointName}`.

Получение маркера доступа от определенного внешнего поставщика

`fed_tkn_${fedPointType}_${fedPointName}`

Blitz Identity Provider можно *настроить* (страница 143) таким образом, что будут сохраняться маркеры доступа пользователя от внешних поставщиков идентификации. Разрешение позволяет получить маркер доступа пользователя от внешнего поставщика идентификации с типом `${fedPointType}` и именем `${fedPointName}`, например, `fed_tkn_esia_esia_1` для сети `esia:esia_1`.

Для использования сервиса:

`GET /blitz/api/v3/users/${subjectId}/fedToken/${fedPointType}/${fedPointName}`.

Чтобы получить маркер доступа на системное разрешение, приложение должно выполнить запрос для получения маркера:

- Запрос `POST https://login.company.com/blitz/oauth/te`.
- Запрос должен содержать заголовок `Authorization` со значением `Basic {secret}`, где `secret` – это `client_id:client_secret` (например, `app:topsecret`) в формате Base64.
- Тело запроса должно содержать следующие параметры:
  - `grant_type` – принимает значение `client_credentials`;
  - `scope` – запрашиваемое системное разрешение.
- В ответ приложение получит маркер доступа `access_token`, время его жизни `expires_in` и тип маркера `token_type`.

Совет: Рекомендуется, чтобы приложение кэшировало полученный маркер доступа для многократного использования на время, немного меньшее, чем параметр `expires_in`, после чего осуществляло получение нового маркера доступа для обновления в кэше.

- Возможные ошибки при вызове `/oauth/te` соответствуют RFC 6749 и описаны здесь[114].

Примеры

Заголовок

```
Authorization: Basic YWlzOm…XQ=
```

Запрос

```
POST blitz/oauth/te HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Authorization: Basic ZG5ld…lg

grant_type=client_credentials&scope=blitz_groups
```

[114] https://tools.ietf.org/html/rfc6749#section-5.2

Ответ

```
{
   "access_token":"QFiJ9mPgERPuusd36mQvD4mfzYolH_CmuddAJ3YKTOI",
   "expires_in":3600,
   "scope":"blitz_groups",
   "token_type":"Bearer"
}
```

Ошибка

При попытке вызова REST-сервиса с просроченным маркером доступа к нему: `HTTP 401 Unauthorized`.

3.4.2 Учетные записи

Данный раздел содержит REST API для управления учетными записями пользователей.

Регистрация

Метод `PUT https://login.company.com/blitz/reg/api/v3/users`

Регистрация учетной записи пользователя.

Необходимые разрешения: `blitz_api_sys_users_reg`.

Заголовки Для отправки письма на английском языке укажите заголовок `Accept-Language: en` (актуально только для версии `v3`).

Тело запроса

Блок user.attrs

Атрибуты регистрируемой учетной записи:

- `first_name` – фамилия;
- `name` – имя;
- `middle_name` – отчество;
- `phone_number` – номер мобильного телефона в виде составного объекта с атрибутами:
  - `value` – номер телефона в формате 7XXXXXXXXXX;
  - `verified` – признак, что телефон подтвержден – `true` или `false`;
- `email` – адрес электронной почты в виде составного объекта с атрибутами:
  - `value` – адрес электронной почты;
  - `verified` – признак, что адрес подтвержден – `true` или `false`;

Блок user.credentials

Опциональный блок.

- `password` – пароль для создаваемой учетной записи пользователя (должен соответствовать настроенной парольной политике).

Блок actions

Опциональный блок.

Действия, выполняемые после регистрации учетной записи:

- `bindDynClient` - после регистрации учетной записи необходимо ассоциировать с ней ранее выпущенный свободный динамический `client_id` экземпляра мобильного приложения.

  Используется при регистрации пользователя из мобильного приложения.

  Параметры:

  - `type` – имя действия. Должно быть передано значение `bindDynClient`;
  - `client_id` – значение, содержащее динамический `client_id`.

```
"actions": [
    {
        "type": "bindDynClient",
        "client_id": "dyn~test_app~af…59"
    }
]
```

Примеры

Регистрация с подтвержденным email и телефоном

Запрос

```
PUT /blitz/reg/api/v3/users HTTP/1.1
Authorization: Bearer cNw…Nz
Content-Type: application/json

{
    "user": {
        "attrs": {
            "sub": "BIP-9TZYWXQ",
            "family_name": "Иванов",
            "given_name": "Иван",
            "middle_name": "Иванович",
            "email": {
                "value": "ivan.ivanov@example.com",
                "verified": true
            },
            "phone_number": {
                "value": "79991234567",
                "verified": true
            }
        },
        "credentials": {
            "password": "Qwerty_123"
        }
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    }
}
```

Ответ

```
{
    "instanceId": "Yml…Yw",
    "subject": "BIP-9TZYWXQ",
    "context": "M0F…pQ",
    "cookies": [
        {
            "name": "css",
            "value": "cp0…1o"
        }
    ],
    "instructions": []
}
```

Ошибки

Список 11: Пароль не соответствует парольной политике

```
{
    "errors": [
        {
            "errMsg": "Пароль не соответствует парольным политикам: длина менее 8␣
↪символов, не содержит цифру, прописную букву, специальный символ.",
            "field": "password"
        }
    ],
    "context": ""
}
```

Список 12: Нарушена уникальность полей

```
{
    "errors": [
        {
            "errMsg": "Пользователь с таким значением уже зарегистрирован. Для␣
↪дальнейшей регистрации введите другое значение",
            "field": "phone_number"
        },
        {
            "errMsg": "Пользователь с таким значением уже зарегистрирован. Для␣
↪дальнейшей регистрации введите другое значение",
            "field": "email"
        },
        {
            "errMsg": "Пользователь с таким значением уже зарегистрирован. Для␣
↪дальнейшей регистрации введите другое значение",
            "field": "sub"
        }
    ],
    "context": ""
}
```

Регистрация с неподтвержденными email и телефоном

Запрос

```
PUT /blitz/reg/api/v3/users HTTP/1.1
Authorization: Bearer cNw…Nz
Content-Type: application/json

{
    "user": {
        "attrs": {
            "sub": "BIP-1TZYWXQ",
            "family_name": "Иванов",
            "given_name": "Иван",
            "middle_name": "Иванович",
            "email": {
                "value": "ivan.ivanov@example.com",
                "verified": false
            },
            "phone_number": {
                "value": "79991234567",
                "verified": false
            }
        },
        "credentials": {
            "password": "Qwerty_123"
        }
    }
}
```

Ответ №1

Если регистрация вызвана с передачей неподтвержденных телефона и/или email, то сервис отправит пользователю проверочный SMS с кодом подтверждения и/или email с кодом подтверждения и вернет сервисные атрибуты `instructions` и `context`.

Ответ, когда требуется ввод пользователем проверочных кодов:

```
{
    "context": "NIi…qQ",
    "instructions": [
        {
            "mobile": "+79991234567",
            "exp": 1690444604,
            "attemts": 3,
            "name": "mbl-enter-code"
        },
        {
            "email": "ivan.ivanov@example.com",
            "exp": 1690644970,
            "attemts": 3,
            "name": "eml-enter-code"
        }
    ]
}
```

Сервис регистрации может быть настроен так, что регистрация пользователя производится сразу, а контакты в учетную запись прописываются после подтверждения, в этом случае сервис регистрации вернет параметры зарегистрированной учетной записи (`instanceId`, `subject`, `cookies`), а также инструкции для опционального подтверждения контактов в учетной записи:

```
{
    "instanceId": "Yml…Yw",
    "subject": "BIP-1TZYWXQ",
    "context": "NIi…qQ",
    "cookies": [
        {
            "name": "css",
            "value": "t8_…84"
        }
    ],
    "instructions": [
        {
            "mobile": "+79991234567",
            "exp": 1690444604,
            "attemts": 3,
            "name": "mbl-enter-code"
        },
        {
            "email": "ivan.ivanov@example.com",
            "exp": 1690644970,
            "attemts": 3,
            "name": "eml-enter-code"
        }
    ]
}
```

Коды подтверждения

При получении в ответе №1 инструкций `eml-enter-code` и/или `mbl-enter-code` нужно запросить у пользователя ввод кода подтверждения, отправленного на email и на мобильный телефон. После ввода каждого кода вызвать сервис для подтверждения контакта, указанного при регистрации, передав в URL запроса значение из параметра `context`, а в теле запроса – введенный пользователем код подтверждения:

Список 13: Запрос на подтверждение email

```
POST /blitz/reg/api/v3/users/YNx9…Dw HTTP/1.1
Authorization: Bearer cNw…Nz
Content-Type: application/json

{
    "email_code":"269302"
}
```

Список 14: Ответ, если введен неправильный код из email

```
{
    "instructions": [
        {
            "email":"mail123@example.com",
            "exp":1655283696,
            "attemts":2,
            "name":"eml-try-again"},
        {
            "mobile":"79988984169",
            "exp":1655280756,
            "attemts":3,
            "name":"mbl-try-again"
        }
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    ],
    "context":"kE6r…7g"
}
```

Список 15: Ответ, если истек срок действия или превышено число попыток (будет общая ошибка `eml-expired`)

```
{
    "instructions": [
        {
            "email":"mail123@example.com",
            "name":"eml-expired"
        },
        {
            "mobile":"79988984169",
            "exp":1655280756,
            "attemts":3,"name":"mbl-try-again"
        }
    ],
    "context":"kE6r…7g"
}
```

Список 16: Запрос для инициирования повторной отправки кода по email (в качестве значения параметра указать любой код)

```
POST /blitz/reg/api/v3/users/YNx9…Dw HTTP/1.1
Authorization: Bearer cNw…Nz
Content-Type: application/json

{
    "email_code_resend":"123456"
}
```

В случае если email успешно подтвержден, и осталось подтвердить телефон, то в ответе сервиса исчезнет инструкция про подтверждение email, и останется только инструкция про телефон:

Список 17: Ответ, если email подтвержден, но нужно подтвердить номер телефона

```
{
    "instructions": [
        {
            "mobile":"79988984169",
            "exp":1655280756,
            "attemts":3,
            "name":"mbl-try-again"
        }
    ],
    "context":"kE6r…7g"
}
```

Список 18: Запрос на подтверждение номера телефона

```
POST /blitz/reg/api/v3/users/YNx9…Dw HTTP/1.1
Authorization: Bearer cNw…Nz
Content-Type: application/json
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{
    "sms_code":"953568"
}
```

Список 19: Ответ, если введен неправильный код подтверждения телефона

```
{
    "instructions": [
        {
            "email":"mail123@example.com",
            "exp":1655283696,
            "attemts":2,
            "name":"eml-try-again"},
        {
            "mobile":"79988984169",
            "exp":1655280756,
            "attemts":3,
            "name":"mbl-try-again"
        }
    ],
    "context":"kE6r…7g"
}
```

Список 20: Ответ, если истек срок действия

```
{
    "instructions": [
        {
            "mobile":"79988984169",
            "name":"mbl-expired"
        }
    ],
    "context":"kE6r…7g"
}
```

Список 21: Ответ, если превышено число попыток

```
{
    "instructions": [
        {
            "mobile":"79988984169",
            "name":"mbl-no-attempts"
        }
    ],
    "context":"kE6r…7g"
}
```

Список 22: Запрос для инициирования повторной отправки кода по SMS (в качестве значения параметра указать любой код)

```
POST /blitz/reg/api/v3/users/YNx9…Dw HTTP/1.1
Authorization: Bearer cNw…Nz
Content-Type: application/json

{
    "sms_code_resend":"123456"
```

(continues on next page)

(продолжение с предыдущей страницы)

```
}
```

Ответ №2

Если все контакты были подтверждены в процессе регистрации, то в результате вызова сервиса в Blitz Identity Provider будет зарегистрирована учетная запись пользователя с предоставленными атрибутами и паролем. Сервис вернет присвоенный учетной записи идентификатор пользователя (`subject`). Кроме того, вернется ряд сервисных атрибутов (`instructions`, `cookies` и `context`).

```
{
    "instanceId": "Yml…Yw",
    "subject": "BIP-1TZYWXQ",
    "context": "NIi…qQ",
    "cookies": [
        {
            "name": "css",
            "value": "t8_…84"
        }
    ],
    "instructions": []
}
```

Ошибка

Регистрация может завершиться ошибкой. Тогда в теле ответа будет пояснение проблемы. В частности, если в Blitz Identity Provider нарушена уникальность атрибута, то сообщение будет содержать перечень полей, по которым нарушена уникальность.

```
{
    "errors": [
        {
            "errMsg": "Такой пользователь уже зарегистрирован…",
            "field": "email"
        },
        {
            "errMsg": "Такой пользователь уже зарегистрирован…",
            "field": "phone_number"
        }
    ],
    "context": ""
}
```

Регистрация с подтвержденными email и телефоном с передачей динамического client_id

Список 23: Запрос

```
PUT /blitz/reg/api/v3/users HTTP/1.1
Authorization: Bearer cNw…Nz
Content-Type: application/json

{
    "user": {
        "attrs": {
            "sub": "BIP-9TZYWXQ",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
            "family_name": "Иванов",
            "given_name": "Иван",
            "middle_name": "Иванович",
            "email": {
                "value": "ivan.ivanov@example.com",
                "verified": true
            },
            "phone_number": {
                "value": "79991234567",
                "verified": true
            }
        },
        "credentials": {
            "password": "Qwerty_123"
        }
    },
    "actions": [
        {
            "type": "bindDynClient",
            "client_id": "dyn~test-app~c84f26f3-10f3-4b85-a6ee-a4ca12c41d26"
        }
    ]
}
```

Регистрация на английском языке

Список 24: Запрос

```
curl -v --location --request PUT 'https://demo.identityblitz.com/blitz/reg/api/v3/
↪users' \
--header 'Content-Type: application/json' \
--header 'Accept-Language: en' \
--header 'Authorization: Bearer ...' \
--data-raw '{
    "user": {
        "attrs": {
            "sub": "username",
            "phone_number": {
                "value": "89101234567",
                "verified": false
            }
        },
        "credentials": {
            "password": "Qwerty_123"
        }
    }

}'
```

Поиск

Метод `GET https://login.company.com/blitz/api/v3/users`

Поиск учетной записи.

URL-параметры В query передается поисковый запрос в формате Resource Query Language[115] (RQL). Операции:

- `and` – одновременное выполнение поисковых условий;
- `or` – альтернативное выполнение поисковых условий (например, поиск по разным атрибутам);
- `eq` – проверка условия равенства.

При выполнении поиска по атрибуту, имеющему строковое значение, рекомендуется явно специфицировать тип значения. Например, `string:02142527602`.

Внимание: Если поисковый атрибут является строкой, содержащей специальные символы, такие как `&|()=<>,`, то необходимо придерживаться следующего алгоритма экранирования и кодирования параметров:

1. Выполнить кодирование всех значений атрибутов – экранировать присутствующие в параметрах специальные символы. Например, если выполняется поиск по телефону `+7(999)1234567`, то значение параметра должно быть преобразовано к значению `+7%28999%291234567`.
2. Собрать общую строку для передачи в качестве параметра query в запрос. Например, `phone_number=+7%28999%291234567`.
3. Выполнить URL-Encode значения параметра. Например, получится такое значение параметра – `phone_number%3D%2B7%2528999%25291234567`.

Примеры

Простой поисковый запрос

Запрос

```
GET /blitz/api/v3/users?query=eq(phone_number.string:79991234567) HTTP/1.1
Authorization: Basic YXBwX2lkOmFwcF9zZWNyZXQ=
```

Ответ

```
[
   {
      "instanceId":"Mzg5…nU",
      "attrs":{
         "sub":"854436f6-af58-4a3f-8cb7-c2c441eb4a76",
         "family_name":"Иванов",
         "given_name":"Иван",
         "middle_name":"Иванович",
         "phone_number":"79991234567",
      }
   }
]
```

[115] https://github.com/kriszyp/rql

Сложный поисковый запрос

Список 25: Запрос

```
GET /blitz/api/v3/users?query=or(eq(phone_number,string:79991234567),eq(phone_
↪number,string:79991112233)) HTTP/1.1
Authorization: Basic YXBwX2lkOmFwcF9zZWNyZXQ=
```

Поиск по строке, содержащей специальные символы

Список 26: Запрос

```
GET /blitz/api/v3/users?query=phone_number%3D%2B7%2528999%25291234567 HTTP/1.1
Authorization: Basic YXBwX2lkOmFwcF9zZWNyZXQ=
```

Атрибуты

Получение атрибутов

Метод `GET https://login.company.com/blitz/api/v3/users/{subjectId}`

Получение атрибутов любого пользователя по его идентификатору.

Необходимые разрешения: `blitz_api_user` или `blitz_api_sys_users`.

Возвращает JSON, содержащий атрибуты пользователя. В блоке `meta` передаются метаданные учетной записи.

Важно: Атрибут `instanceId` метаданных нужен для возможности вызова в последующем сервисов *изменения атрибутов учетной записи* (страница 450) и *изменения пароля* (страница 458).

Пример

Запрос

```
GET /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a HTTP/1.1
Authorization: Bearer cNw…Nz
```

Ответ

```
{
    "family_name": "Иванов",
    "sub": "d2580c98 e584 4aad a591 97a8cf45cd2a",
    "given_name": "Иван",
    "locked": false,
    "meta": {
        "instanceId": "Mzg…J1",
        "unmodifiable": [
            "sub"
        ]
    }
}
```

Изменение атрибута

Метод `POST https://login.company.com/blitz/api/v3/users/{instanceId}`

Изменение атрибутов пользователя по `instanceId`. Чтобы узнать значение `instanceId`, необходимо предварительно вызвать методом GET сервис *получения атрибутов* (страница 449) пользователя.

Необходимые разрешения: `blitz_api_user_chg` или `blitz_api_sys_users_chg`.

Тело запроса Значения изменяемых атрибутов пользователя.

Возвращает JSON, содержащий атрибуты пользователя.

Если переданные значения атрибутов не прошли проверку, вернется ошибка `HTTP 400 Bad Request` и вложенный JSON, включающий:

- тип ошибки (type) – `input_error` для случаев, когда запрос содержит некорректное или недопустимое значение;
- код ошибки (error);
- текстовое описание ошибки.

Примечание: Коды ошибок и тексты ошибок могут быть определены специфично для различных атрибутов и определяться реализованной для атрибутов логикой валидаторов.

Пример

Запрос

```
POST /blitz/api/v3/users/Mzg…J1 HTTP/1.1
Authorization: Bearer cNw…Nz
Content-Type: application/json

{
   "family_name":"Петров"
}
```

Ответ

```
{
    "family_name": "Петров",
    "given_name": "Иван",
    "locked": false,
    "sub": "5cffd68f-2cb8-4f7a-b0f3-9fa69a1fbbcd",
    "meta": {
        "instanceId": "Mzg…J1",
        "unmodifiable": [
            "sub"
        ]
    }
}
```

Ошибка

```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "contact_use_violation",
            "desc": "Validation mobile:79988887812 is failed.",
            "pos": "mobile"
        }
    ]
}
```

Изменение номера телефона

Метод Частный случай *изменения атрибута* (страница 450).

Режимы:

- изменение телефона сразу на подтвержденный,
- изменение телефона с прохождением подтверждения.

Тело запроса

- `phone_number` – мобильный телефон, в виде составного объекта с атрибутами:
  - `value` – номер телефона в формате `7XXXXXXXXXX`;
  - `vrf` – признак, что телефон подтвержден – true.

Примеры

Изменение номера на подтвержденный

Запрос

```
POST /blitz/api/v3/users/Mzg…J1 HTTP/1.1
Authorization: Bearer wzb…Tw
Content-Type: application/json

{
    "phone_number":
        {
            "value":"79991234567",
            "vrf":true
        }
}
```

Ответ


```
{
    "given_name": "Иван",
    "family_name": "Иванов",
    "meta": {
        "instanceId": "Mzg5L…2M",
        "unmodifiable": [
            "uid"
        ]
    },
    "email": {
        "value": "aivanov+2@gmail.com",
        "vrf": true
    },
    "sub": "BIP-LIR6BO33XBBDHANE6DZPUTYVME",
    "phone_number": {
        "value": "+7(999)1234567",
        "vrf": true
    }
}
```


Изменение номера с прохождением подтверждения

Запрос

```
POST /blitz/api/v3/users/Mzg…J1 HTTP/1.1
Content-Type: application/json
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5) AppleWebKit/537.36␣
↪(KHTML, like Gecko) Chrome/83.0.4103.106 Safari/537.36
Authorization: Bearer wzb…Tw

{
    "phone_number":{"value":"+79999999998","vrf":false}
}
```

Ответ №1

Промежуточный ответ содержит указание на необходимость подтверждения нового номера телефона. Код подтверждения отправляется пользователю на новый номер.


```
{
    "given_name": "Иван",
    "family_name": "Иванов",
    "meta": {
        "instanceId": "Mzg5L…2M",
        "unmodifiable": [
            "sub"
        ]
    },
    "email": {
        "value": "aivanov+2@gmail.com",
        "vrf": true
    },
    "sub": "BIP-LIR6BO33XBBDHANE6DZPUTYVME",
    "notes": {
```


(continues on next page)

(продолжение с предыдущей страницы)

```
        "actions": {
            "state": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
            "exp": 300,
            "status": "code_waiting",
            "from": "+7(964)1234567",
            "attr": "mobile",
            "attempts_left": 3,
            "value": "+7(999)9999998",
            "action": "validate_mobile",
            "created": 1598446512
        }
    },
    "phone_number": {
        "value": "+7(964)1234567",
        "vrf": true
    }
}
```

Код подтверждения

Нужно получить от пользователя код подтверждения нового номера телефона и отправить его в Blitz Identity Provider в запросе. В URL данного запроса использовать значение параметра `actions: state` из ответа №1:

```
POST /blitz/api/v3/users/notes/validate_mobile/ch_El…yQ HTTP/1.1
Content-Type: application/json
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)…
Authorization: Bearer wzb…Tw

{
    "cmd": "code",
    "value": "123456"
}
```

Ответ №2

Список 27: Успешное изменение номера телефона

```
{
    "given_name": "Иван",
    "family_name": "Иванов",
    "meta": {
        "instanceId": "Mzg5L…2M",
        "unmodifiable": [
            "sub"
        ]
    },
    "email": {
        "value": "aivanov+2@gmail.com",
        "vrf": true
    },
    "sub": "BIP-LIR6BO33XBBDHANE6DZPUTYVME",
    "phone_number": {
        "value": "+7(999)9999998",
        "vrf": true
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    }
}
```

Ошибка

Список 28: Неверный код

```
{
    "state": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
    "exp": 2592000,
    "from": "+7(964)1234567",
    "attr": "phone_number",
    "msg": "wrong_code",
    "attempts_left": 2,
    "created": 1649695409,
    "value": "+7(999)9999998",
    "action": "validate_mobile"
}
```

Список 29: Превышено количество попыток ввести верный код

```
{
    "state": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
    "id": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
    "attr": "phone_number",
    "cause": "no_attempts_left",
    "from": "+7(964)1234567",
    "value": "+7(999)9999998",
    "action": "validate_mobile"
}
```

Список 30: Код просрочен

```
{
    "state": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
    "id": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
    "attr": "phone_number",
    "cause": "code_expired",
    "from": "+7(964)1234567",
    "value": "+7(999)9999998",
    "action": "validate_mobile"
}
```

Изменение адреса электронной почты

Метод Частный случай *изменения атрибута* (страница 450).

Режимы:

- изменение email сразу на подтвержденный,
- изменение email с прохождением подтверждения.

Тело запроса

- `email` – адрес электронной почты:
  - `value` – адрес электронной почты;

– `vrf` – признак, что адрес подтвержден – `true`;

Примеры

Изменение адреса на подтвержденный

Запрос

```
POST /blitz/api/v3/users/Mzg…J1 HTTP/1.1
Authorization: Bearer wzb…Tw
Content-Type: application/json

{
    "email":
        {
            "value":"mail@example.com",
            "vrf":true
        }
}
```

Ответ

```
{
    "given_name": "Иван",
    "family_name": "Иванов",
    "meta": {
        "instanceId": "Mzg5LW…2M",
        "unmodifiable": [
            "sub"
        ]
    },
    "mail": {
        "value": "mail@example.com",
        "vrf": true
    },
    "sub": "BIP-LIR6BO33XBBDHANE6DZPUTYVME",
    "phone_number": {
        "value": "+7(999)1234567",
        "vrf": true
    }
}
```

Изменение адреса с прохождением подтверждения

Запрос

```
POST /blitz/api/v3/users/Mzg…J1 HTTP/1.1
Content-Type: application/json
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5) AppleWebKit/537.36␣
↪(KHTML, like Gecko) Chrome/83.0.4103.106 Safari/537.36
Authorization: Bearer wzb…Tw

{
```

(continues on next page)

(продолжение с предыдущей страницы)

```
 "email":{"value":"mail@example.com","vrf":false}
}
```

Ответ №1

Промежуточный ответ содержит указание на необходимость подтверждения нового адреса электронной почты. Код подтверждения отправляется пользователю на новый адрес.

```
{
    "given_name": "Иван",
    "family_name": "Иванов",
    "meta": {
        "instanceId": "Mzg5L…2M",
        "unmodifiable": [
            "sub"
        ]
    },
    "email": {
        "value": "aivanov+2@gmail.com",
        "vrf": true
    },
    "sub": "BIP-LIR6BO33XBBDHANE6DZPUTYVME",
    "notes": {
        "actions": {
            "state": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
            "exp": 86400,
            "status": "code_waiting",
            "from": "aivanov+2@gmail.com",
            "attr": "mail",
            "attempts_left": 3,
            "value": "mail@example.com",
            "action": "validate_mail",
            "created": 1598446512
        }
    },
    "phone_number": {
        "value": "+7(964)1234567",
        "vrf": true
    }
}
```

Код подтверждения

Нужно получить от пользователя код подтверждения нового адреса электронной почты и отправить его в Blitz Identity Provider в запросе. В URL данного запроса использовать значение параметра `actions:state` из ответа №1:

```
POST /blitz/api/v3/users/notes/validate_email/ch_El…yQ HTTP/1.1
Content-Type: application/json
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)…
Authorization: Bearer wzb…Tw

{
    "cmd": "code",
    "value": "123456"
}
```

Ответ №2

Список 31: Успешное изменение адреса электронной почты


```
{
    "given_name": "Иван",
    "family_name": "Иванов",
    "meta": {
        "instanceId": "Mzg5L…2M",
        "unmodifiable": [
            "sub"
        ]
    },
    "email": {
        "value": "mail@example.com",
        "vrf": true
    },
    "sub": "BIP-LIR6BO33XBBDHANE6DZPUTYVME",
    "phone_number": {
        "value": "+7(999)9999998",
        "vrf": true
    }
}
```


Ошибка

Список 32: Неверный код


```
{
    "state": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
    "exp": 2592000,
    "from": "aivanov+2@gmail.com",
    "attr": "email",
    "msg": "wrong_code",
    "attempts_left": 2,
    "created": 1649695409,
    "value": "mail@example.com",
    "action": "validate_email"
}
```


Список 33: Превышено количество попыток ввести верный код


```
{
    "state": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
    "id": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
    "attr": "email",
    "cause": "no_attempts_left",
    "from": "aivanov+2@gmail.com",
    "value": "mail@example.com",
    "action": "validate_email"
}
```


Список 34: Код просрочен


```
{
    "state": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
    "id": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
```


(continues on next page)

(продолжение с предыдущей страницы)

```
    "attr": "email",
    "cause": "code_expired",
    "from": "aivanov+2@gmail.com",
    "value": "mail@example.com",
    "action": "validate_email"
}
```

Пароли

Изменение пароля

Метод `POST https://login.company.com/blitz/api/v3/users/{instanceId}/pswd`

Изменение пароля. Чтобы узнать значение `instanceId` для пользователя, необходимо предварительно вызвать методом GET сервис *получения атрибутов* (страница 449) пользователя.

Необходимые разрешения: `blitz_api_usec_chg` или `blitz_api_sys_usec_chg`.

Заголовки

- При смене пароля в пользовательском режиме необходимо передать заголовки с IP-адресом пользователя и `User-Agent`.
- В сценарии самостоятельной смены пользователем пароля в Личном кабинете возможен сброс сессий пользователя. При этом может быть нежелательно, чтобы произошел выход пользователя с текущего устройства/браузера. Для того, чтобы указать Blitz Identity Provider, что определенное устройство необходимо сохранить по результатам успешной смены пароля (не делать с него логаут), необходимо в вызов сервиса смены пароля передать от приложения заголовок `IB-CI-UA-ID` с идентификатором текущего устройства пользователя.

  Совет: Идентификатор текущего устройства пользователя можно получить из *маркера идентификации* (страница 400).

- Для отправки письма на английском языке укажите заголовок `Accept-Language: en` (актуально только для версии `v3`).

Тело запроса

- `current` – текущий пароль пользователя (только при смене пароля в пользовательском режиме – обязательно передается).
- `password` – новый пароль пользователя (необязательный параметр). Если параметр не задан, то Blitz Identity Provider самостоятельно сгенерирует новый пароль.
- `resetSessions` – в случае если параметр не указан или указан в значении `true`, то при смене пароля будут аннулированы все сессии пользователя и удалены запомненные устройства. Если необходимо только сменить пароль без сброса сессий, то необходимо явно указать параметр в значении `false`.
- `sendPswdToAttr` – имя атрибута с телефонным номером для отправки пользователю пароля (необязательный параметр). Если параметр задан, то пользователю на телефон из указанного атрибута будет отправлена SMS с паролем.

Возвращает

- В случае успешного вызова Blitz Identity Provider - `HTTP 204 No Content`.
- Если смена пароля завершилась ошибкой - сообщение об ошибке:
  - `HTTP 401 Unauthorized` в случае ошибки контроля доступа - неправильный маркер доступа или неправильный текущий пароль пользователя.

– `HTTP 400 Bad Request` - новый пароль не удовлетворяет требованиям парольной политики.

Примеры

Запрос

Список 35: Пользовательский режим смены пароля

```
POST /blitz/api/v3/users/Mzg…J1/pswd HTTP/1.1
Content-Type: application/json
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)…
Authorization: Bearer wzb…Tw
IB-CI-UA-ID: {SHA256}rVWFmwgRKWeW_flH4CA4yuW7OhKZ32Da94m0kzwWsVs

{
    "current": "QWErty123",
    "password": "P@$$w0rd",
    "resetSessions": false
}
```

Список 36: Режим смены пароля системой

```
POST /blitz/api/v3/users/Mzg…J1/pswd HTTP/1.1
Content-Type: application/json
Authorization: Bearer qwa…Ez

{
    "password": "P@$$w0rd",
    "resetSessions": true
}
```

Список 37: Отправка нового пароля по SMS с автоматической генерацией пароля

```
POST /blitz/api/v3/users/Mzg…J1/pswd HTTP/1.1
Content-Type: application/json
Authorization: Bearer qwa…Ez

{
    "sendPswdToAttr": "phone_number"
}
```

Список 38: Запрос смены пароля на английском языке

```
curl -v --location --request POST 'https://demo.identityblitz.com/blitz/api/v3/
↪users/YnVpbHQtaW46a2dhdnJpbG92QGlkYmxpdHoucnU6MTcxMDU5ODgyODY3MjU0ODg2NA/pswd' \
--header 'Content-Type: application/json' \
--header 'Accept-Language: en' \
--header 'Authorization: Bearer ...' \
--data-raw '{"password": "nN2L98Nu1234"}'
```

Ошибки

Список 39: Неправильный текущий пароль

```
{
    "type": "security_error",
    "error": "invalid_credential",
    "desc": "Wrong subject identifier or current password"
}
```

Список 40: Неправильный маркер доступа

```
{
    "type": "security_error",
    "error": "bad_access_token",
    "desc": "BEARER_AUTH: CRID does not match"
}
```

Список 41: Новый пароль не соответствует парольной политике: слишком короткий

```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "password_policy_violated",
            "desc": "Password's length must be greater than 6",
            "pos": "password",
            "params": {
                "rule": "to_short",
                "low": 6
            }
        }
    ]
}
```

Список 42: Новый пароль не соответствует парольной политике, установленной в LDAP-каталоге

```
{
    "type": "input_error",
    "error": "password_policy_violated",
    "desc": "Failed to update password\n",
    "pos": "password",
    "params": {
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        "rule": "id_store"
    }
}
```

Список 43: Новый пароль не соответствует парольной политике: не содержит требуемых групп символов

```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "password_policy_violated",
            "desc": "Password doesn't match enough symbols groups",
            "pos": "password",
            "params": {
                "rule": "not_enough_groups",
                "no_matched_groups": [
                    {
                     "desc": "password.policy.desc.digits",
                     "min_number_symbols": 1
                    },
                    {
                        "desc": "password.policy.desc.capital",
                        "min_number_symbols": 1
                    },
                    {
                        "desc": "password.policy.desc.special",
                        "min_number_symbols": 1
                    }
                ]
            }
        }
    ]
}
```

Список 44: Новый пароль не соответствует парольной политике: пароль ранее использовался

```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "password_policy_violated",
            "desc": "Password found in previous used ones",
            "pos": "password",
            "params": {
                "rule": "in_password_history"
            }
        }
    ]
}
```

Список 45: Новый пароль не соответствует парольной политике: новый пароль совпадает с текущим


```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "password_policy_violated",
            "desc": "A new password can't be the same as the current",
            "pos": "password",
            "params": {
                "rule": "eq_current"
            }
        }
    ]
}
```


Список 46: Новый пароль не соответствует парольной политике: в новом пароле недостаточное число символов отличается от прежнего


```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "password_policy_violated",
            "desc": "There are not enough new characters in a new password",
            "pos": "password",
            "params": {
                "rule": "not_enough_new_chars",
                "minNew": 5
            }
        }
    ]
}
```


Список 47: Новый пароль не соответствует парольной политике: пароль включает вхождение из словаря запрещенных паролей


```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "password_policy_violated",
            "desc": "Password contains a word from the stop dictionary",
            "pos": "password",
            "params": {
                "rule": "in_stop_dic",
                "stop_word": "qwerty"
            }
        }
    ]
}
```

Список 48: Новый пароль не соответствует парольной политике: пароль совпадает со словарным


```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "password_policy_violated",
            "desc": "Password found in a password dictionary",
            "pos": "password",
            "params": {
                "rule": "in_password_dic"
            }
        }
    ]
}
```


Список 49: Новый пароль не соответствует парольной политике: пароль изменен ранее разрешенного срока


```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "password_policy_violated",
            "desc": "Password is too young",
            "pos": "password",
            "params": {
                "rule": "too_young",
                "minAgeInSec": 86400
            }
        }
    ]
}
```


Список 50: Переданный атрибут для отправки пароля не существует


```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "wrong_value",
            "desc": "Wrong mobile attribute 'phone_number_wrong'",
            "pos": "sendPswdToAttr"
        }
    ]
}
```


Список 51: У пользователя не задан атрибут с телефоном для отправки пароля на телефон


```
{
```


(continues on next page)

(продолжение с предыдущей страницы)

```
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "wrong_value",
            "desc": "User not contains mobile attribute 'phone_number'",
            "pos": "sendPswdToAttr"
        }
    ]
}
```

Изменение пароля ведомого аккаунта

Метод `POST https://login.company.com/blitz/api/v2/users/{subjectId}/password`

Изменение пароля ведомой учетной записи пользователя с помощью ведущей учетной записи пользователя. `subjectId` – идентификатор (`sub`) ведомой учетной записи.

Заголовки В запрос должен быть добавлен заголовок с маркером доступа на разрешение с именем `blitz_change_password`, полученным ведущей учетной записью. Ведущий пользователь может вызвать смену пароля ведомого только в том случае, если ранее ведущему пользователю *было дано* (страница 506) право на изменение пароля `change_password`.

Тело запроса Атрибут `value` со значением нового пароля, которое должно соответствовать требованиям настроенной парольной политики.

Возвращает

- При успешной смене пароля - статус `HTTP 200 (OK)`.
- При наличии ошибки - описание полученной ошибки.

Пример

Запрос

```
POST /blitz/api/v2/users/c574a512-3704-4576-bc3a-3fe28b636e85/password HTTP/1.1
Authorization: Bearer cNwIX…Tg
Content-Type: application/json

{"value":"QWErty1234"}
```

Ошибка

```
{
    "errors": [
        {
            "code": "access_denied",
            "desc": "Not enough rights: change_password",
            "params": {}
        }
    ]
}
```

Режимы аутентификации

Проверка состояния

Метод `GET https://login.company.com/blitz/api/v3/users/{subjectId}/auth`

Проверка состояния следующих режимов аутентификации учетной записи `subjectId`:

- наличие включенной двухфакторной аутентификации;
- наличие установленного признака необходимости смены пароля;
- наличие временного запрета по входу с использованием определенного метода входа.

Необходимые разрешения: `blitz_api_usec` или `blitz_api_sys_usec`.

Возвращает

- `requiredFactor` признак включенной двухфакторной аутентификации. Может принимать следующие значения:
  - отсутствует, `0` или `1` - выключен,
  - `2` - включен (требуется 2-й фактор аутентификации);
- `needPasswordChange` признак необходимости смены пароля при входе;
- `methodsLocked` список заблокированных методов аутентификации. Пользователь не может использовать данные методы входа, но может использовать остальные.

Пример

Запрос

```
GET /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/auth HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Cache-Control: no-cache
```

Ответ

```
{
    "requiredFactor": 2,
    "needPasswordChange": true,
    "methodsLocked": ["password"]
}
```

Изменение режимов аутентификации

`POST https://login.company.com/blitz/api/v3/users/{subjectId}/auth`

Изменения режимов аутентификации пользователя.

Необходимые разрешения: `blitz_api_usec_chg` или `blitz_api_sys_usec_chg`.

Заголовки В пользовательском режиме необходимо передать заголовки с IP-адресом пользователя и `User-Agent`.

Тело запроса Может содержать параметры:

- `requiredFactor` признак включенной двухфакторной аутентификации. Значения:
  - `null` выключен,

  - `2` включен (требуется 2-й фактор аутентификации);

- `needPasswordChange` признак необходимости смены пароля при входе – допустима только передача значения `true`;
- `methodsLocked` список заблокированных методов аутентификации. Пользователь не может использовать данные методы входа, но может использовать остальные. В настоящий момент Blitz Identity Provider поддерживает только блокирование использования парольного входа (`password`).

Пример

Запрос

```
POST /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/auth HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)…
Content-Type: application/json

{
    "requiredFactor": 2,
    "needPasswordChange": true,
    "methodsLocked": ["password"]
}
```

Ответ

```
{
    "requiredFactor": 2,
    "needPasswordChange": true,
    "methodsLocked": ["password"]
}
```

Ошибка

Список 52: HTTP 400 Bad Request: у пользователя не настроен ни один метод для второго фактора аутентификации

```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "has_not_sf_methods",
            "desc": "User 'd2580c98-e584-4aad-a591-97a8cf45cd2a' has not any␣
↪second factor method",
            "pos": "requiredFactor"
        }
    ]
}
```

Свойства пользователя

Получение свойств

Метод `GET https://login.company.com/blitz/api/v3/users/{subjectId}/props`

Получение свойств любого пользователя по его идентификатору.

Необходимые разрешения: `blitz_api_user` или `blitz_api_sys_users`.

Возвращает HTTP 200 и JSON, содержащий свойства пользователя.

Пример

Запрос

```
GET /blitz/api/v3/users/854436f6-af58-4a3f-8cb7-c2c441eb4a76/props HTTP/1.1
Content-Type: application/json
Authorization: Bearer cNw…Nz
```

Ответ


```
{
        "pipes.info.fed.readOn":1706530413,
        "fcOn":1707814866,
        "pipes.info.adv-totp.readOn":1696236815,
        "pipes.addKey.mobile.Android.disagreedOn":1701099042,
        "pipes.act.mobile.skippedOn":1695649488,
        "wak.failedOn":1689864670,
        "pipes.act.mobile.outdatedOn":1695649486,
        "last2fa":"x509",
        "pipes.addKey.pc.Windows.disagreedOn":1706100800,
        "pipes.act.mail.skippedOn":1689764346
}
```


Добавление, изменение и удаление свойств

Метод `POST https://login.company.com/blitz/api/v3/users/{subjectId}/props`

Добавление, изменение и удаление свойств пользователя по его идентификатору.

Необходимые разрешения: `blitz_api_user` или `blitz_api_sys_users`.

Тело запроса JSON с перечнем свойств для добавления и удаления. Для изменения значения нужно отправить новое значение свойства в секции `add`. Для удаления можно только указать удаляемое свойство.

Возвращает HTTP 200 и JSON, содержащий актуальные свойства.

Пример

Запрос

Список 53: Удаление свойства last2fa и добавление testBool

```
POST /blitz/api/v3/users/854436f6-af58-4a3f-8cb7-c2c441eb4a76/props HTTP/1.1
Content-Type: application/json
Authorization: Bearer cNw…Nz

{
  "remove" : ["last2fa"],
  "add" : {
    "testBool" : true
  }
}
```

Список 54: Изменение свойства testBool

```
POST /blitz/api/v3/users/854436f6-af58-4a3f-8cb7-c2c441eb4a76/props HTTP/1.1
Content-Type: application/json
Authorization: Bearer cNw…Nz

{
  "add" : {
    "testBool" : false
  }
}
```

Список 55: Удаление свойства testBool

```
POST /blitz/api/v3/users/854436f6-af58-4a3f-8cb7-c2c441eb4a76/props HTTP/1.1
Content-Type: application/json
Authorization: Bearer cNw…Nz

{
  "remove" : ["testBool"]
}
```

Ответ

Список 56: Удаление свойства last2fa и добавление testBool

```
{
        "pipes.act.mobile.skippedOn":1695649488,
        "pipes.act.mobile.outdatedOn":1695649486,
        "testBool":true,
        "pipes.addKey.mobile.Android.disagreedOn":1701099042,
        "pipes.info.adv-totp.readOn":1696236815,
        "wak.failedOn":1689864670,
        "pipes.info.fed.readOn":1706530413,
        "pipes.act.mail.skippedOn":1689764346,
        "fcOn":1707814866,
        "pipes.addKey.pc.Windows.disagreedOn":1706100800
}
```

Список 57: Изменение свойства testBool


```
{
        "pipes.act.mobile.skippedOn":1695649488,
        "pipes.act.mobile.outdatedOn":1695649486,
        "testBool":false,
        "pipes.addKey.mobile.Android.disagreedOn":1701099042,
        "pipes.info.adv-totp.readOn":1696236815,
        "wak.failedOn":1689864670,
        "pipes.info.fed.readOn":1706530413,
        "pipes.act.mail.skippedOn":1689764346,
        "fcOn":1707814866,
        "pipes.addKey.pc.Windows.disagreedOn":1706100800
}
```


Список 58: Удаление свойства testBool


```
{
        "pipes.act.mobile.skippedOn":1695649488,
        "pipes.act.mobile.outdatedOn":1695649486,
        "pipes.addKey.mobile.Android.disagreedOn":1701099042,
        "pipes.info.adv-totp.readOn":1696236815,
        "wak.failedOn":1689864670,
        "pipes.info.fed.readOn":1706530413,
        "pipes.act.mail.skippedOn":1689764346,
        "fcOn":1707814866,
        "pipes.addKey.pc.Windows.disagreedOn":1706100800
}
```


TOTP

Совет: См. RFC 6238 TOTP: Time-Based One-Time Password Algorithm[116].

Проверка наличия TOTP

Метод `GET https://login.company.com/blitz/api/v3/users/{subjectId}/totps`

Проверка наличия у пользователя настроенного TOTP-генератор кодов подтверждения.

Необходимые разрешения: `blitz_api_usec` или `blitz_api_sys_usec`.

Возвращает Если TOTP настроен, то в ответ будут получены его настройки.

[116] https://tools.ietf.org/html/rfc6238

Пример

Запрос

```
GET /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/totps HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Cache-Control: no-cache
```

Ответ

```
[
    {
        "id": "SW_TOTP_1_d2580c98-e584-4aad-a591-97a8cf45cd2a",
        "len": 6,
        "name": "Google Authenticator"
    }
]
```

Привязка TOTP

Привязка к учетной записи пользователя TOTP-генератора осуществляется в два этапа.

Этап №1

Метод `GET https://login.company.com/blitz/api/v3/users/{subjectId}/totps/attach/qr`

Запрос в Blitz Identity Provider QR-кода и строки привязки.

Необходимые разрешения: `blitz_api_usec_chg` или `blitz_api_sys_usec_chg`.

Заголовки В пользовательском режиме необходимо передать заголовки с IP-адресом пользователя и `User-Agent`.

Возвращает Атрибуты:

- `base64QRCode` – QR-код привязки генератора, который нужно отобразить пользователю;
- `base32Secret` – секретная строка привязки генератора, которую нужно отобразить пользователю, если ему неудобно будет фотографировать QR-код, и он предпочтет ввести код привязки в генератор вручную.

Пример

Запрос

```
GET /blitz/api/v3/users/d25..2a/totps/attach/qr HTTP/1.1
Authorization: Bearer cN..z
Cache-Control: no-cache
```

Ответ

```
{
    "base64QRCode": "iVB…g==",
    "base32Secret": "W247OHVTPPTIAOXMGKK6Z7BZ3DEYWO74"
}
```

Этап №2

Метод `POST https://login.company.com/blitz/api/v3/users/{subjectId}/totps/attach/qr`

Подтверждение регистрации привязки.

Необходимые разрешения: `blitz_api_usec_chg` или `blitz_api_sys_usec_chg`.

Тело запроса

- `base32Secret` – секретная строка инициализации TOTP-генератора;
- `otpCode` – код подтверждения, выработанный генератором по алгоритму TOTP от строки secret и текущего временного слота;
- `name` – отображаемое имя TOTP-генератора (необязательно).

Возвращает

- В случае успешного выполнения - `HTTP 204 No Content`.
- В случае ошибки сервис - `HTTP 400 Bad Request`.

Пример

Запрос

```
POST /blitz/api/v3/users/d2580c98..cd2a/totps/SW_TOTP_1_d2580c98..cd2a HTTP/1.1
Content-Type: application/json
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)

{
    "base32Secret": "W247OHVTPPTIAOXMGKK6Z7BZ3DEYWO74",
    "name": "Google Authenticator",
    "otpCode": "123456"
}
```

Ответ

```
{
    "base64QRCode": "iVB…g==",
    "base32Secret": "W247OHVTPPTIAOXMGKK6Z7BZ3DEYWO74"
}
```

Ошибка

Список 59: Передан неправильный код

```
{
    "type": "process_error",
    "error": "wrong_otp_code"
}
```

Удаление привязки

Метод `DELETE https://login.company.com/blitz/api/v3/users/{subjectId}/totps/{id}`

Удаление привязки TOTP-генератора к учетной записи пользователя.

Необходимые разрешения: `blitz_api_usec_chg` или `blitz_api_sys_usec_chg`.

URL-параметры В качестве `id` указывается *полученный* (страница 469) идентификатор привязки.

Заголовки В пользовательском режиме необходимо передать заголовки с IP-адресом пользователя и `User-Agent`.

Возвращает При успешном выполнении сервис вернет `HTTP 204 No Content`.

Пример

Список 60: Запрос

```
DELETE /blitz/api/v3/users/d..2a/totps/SW_TOTP_1_d..2a HTTP/1.1
Authorization: Bearer cN..z
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)...
```

Состояние учетной записи

Проверка состояния учетной записи

Метод `GET https://login.company.com/blitz/api/v3/users/{subjectId}/state`

Проверка состояния учетной записи:

- наличие блокировки по причине неактивности;
- наличие запрета на блокировку по причине неактивности.

Необходимые разрешения: `blitz_api_usec` или `blitz_api_sys_usec`.

Примеры

Запрос

```
GET /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/state HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Cache-Control: no-cache
```

Ответы

Список 61: Состояние учетной записи еще не инициализировано (учетная запись только создана или еще не использовалась до входа с момента появления функции)

```
{
  "name": "initial"
}
```

Список 62: Учетная запись активна

```
{
  "name": "active",
  "checkedOn": 1688106755
}
```

Примечание: В параметре `checkedOn` хранится временная отметка последней проверки состояния.

Список 63: Учетная запись заблокирована по причине длительной неактивности

```
{
  "name": "inactivityLock",
  "on": 1688106646
}
```

Примечание: В параметре `on` хранится время блокировки.

Список 64: Учетная запись находится в списке исключений и не может быть заблокирована по причине неактивности до наступления даты из параметра `till`

```
{
  "name": "untouchable",
  "till": 1689106755
}
```

Примечание: Если параметр `till` отсутствует, то учетная запись не может быть вообще заблокирована по причине неактивности.

Изменение состояния учетной записи

Метод `POST https://login.company.com/blitz/api/v3/users/{subjectId}/state`

Изменение состояния учетной записи пользователя.

Необходимые разрешения: `blitz_api_sys_usec_chg`.

Тело запроса Возможные параметры:

- `name` - назначаемое состояние. Можно назначить только состояние `untouchable`;
- `till` - необязательный параметр, в котором можно указать время, до которого учетной записи назначается состояние `untouchable`. Для отмены состояния `untouchable` можно назначить параметру `till` текущее время.

Возвращает В случае успешного вызова `HTTP 204 No Content`.

Пример

Список 65: Запрос

```
POST /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/state HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json

{
    "name": "untouchable",
    "till": 1689106755
}
```

Внешние поставщики

Список внешних поставщиков

Метод `GET /api/v3/users/{subjectId}/fa`

Получение списка привязок учетных записей внешних поставщиков идентификации к учетной записи пользователя.

Необходимые разрешения: `blitz_api_ufa` или `blitz_api_sys_ufa`.

Возвращает Тип и имя привязки (`fpType` и `fpName`), идентификатор привязки (`sid`) и имя пользователя (`userName`).

Пример

Запрос

```
GET /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/fa HTTP/1.1
Authorization: Bearer m9tuVBNUnizkuwFnq95IXQm1XTplXLUFD1O5TUmGij4
Cache-Control: no-cache
```

Ответ

```
[
    {
        "sid": "1000347601",
        "fpType": "esia",
        "fpName": "esia_1",
        "userName": "user.name@esia.ru"
    },
    {
        "sid": "1234",
        "fpType": "tcs",
        "fpName": "tcs_1"
        "userName": "Олег"
    }
]
```

Привязка поставщика по идентификатору

Метод `POST /api/v3/users/{subjectId}/fa/{fpType}/{fpName}/{sid}`

Привязка учетной записи внешнего поставщика идентификации к учетной записи пользователя, если вход через внешний поставщик идентификации произведен ранее иными средствами и известен идентификатор (`sid`) учетной записи во внешнем поставщике идентификации.

Необходимые разрешения: `blitz_api_ufa_chg` или `blitz_api_sys_ufa_chg`.

URL-параметры `guid` пользователя (`subjectId`), тип внешнего поставщика (`fpType`), имя внешнего поставщика (`fpName`) и идентификатор учетной записи во внешнем поставщике (`sid`).

Тело запроса JSON:

- `federatedAccountName`: имя внешней учетной записи, которую необходимо привязать (опционально). Если параметр не передается, то используется прежнее имя.

Возвращает В случае успешного вызова `204 No Content`.

Пример

Список 66: Запрос

```
POST /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/fa/tcs/tcs_1/1234␣
↪HTTP/1.1
Authorization: Bearer m9tuVBNUnizkuwFnq95IXQm1XTplXLUFD1O5TUmGij4

{
    "federatedAccountName": "Elle Woods"
}
```

Привязка поставщика

Привязка к учетной записи внешнего поставщика при неизвестном идентификатору учетной записи во внешнем поставщике осуществляется в два этапа:

- Запрос инструкции привязки.
- Выполнение привязки пользователем в браузере.

Метод `POST /api/v2/users/current/fa/bind`

Запрос инструкции привязки.

Тело запроса

- `fp` – идентификатор поставщика, связь с профилем которого должна быть установлена;
- `callback` – адрес, на который должен быть возвращен пользователь после успешной привязки аккаунта соцсети;
- `isPopup` – требуется ли открытие страницы поставщика идентификации в popup-окне (опционально).

Возвращает Параметр `redirectTo` с ссылкой, на которую необходимо направить пользователя в браузере для выполнения второго этапа и создания привязки учетной записи пользователя к внешнему поставщику идентификации.

Пример

Запрос

```
POST /blitz/api/v2/users/current/fa/bind HTTP/1.1
Authorization: Basic ZG5ldm5pay10ZXN0Lm1vcy5ydTphUU56S0JuY2VBQVQwelg
Content-Type: application/json

{
    "fp": "vk:vk_1",
    "callback": "https://app.company.com/callback"
}
```

Ответ

```
200 OK
{
"redirectTo": "https://login.company.com/blitz/api/v2/users/current/fa/bind/auth/
↪fc111c86-5193-42a2-862a-d819a4f45a86"
}
```

Удаление привязки поставщика

Метод `DELETE /api/v2/users/{subjectId}/fa/{fpType}/{fpName}/{sid}`

Удаление привязки внешнего поставщика к пользователю.

URL-параметры `guid` пользователя (`subjectId`), тип внешнего поставщика (`fpType`), имя внешнего поставщика (`fpName`) и *идентификатор учетной записи во внешнем поставщике* (страница 474) (`sid`).

Пример

Список 67: Заголовок

```
DELETE /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/fa/tcs/tcs_1/1234␣
↪HTTP/1.1
Authorization: Bearer m9tuVBNUnizkuwFnq95IXQm1XTplXLUFD1O5TUmGij4
```

Получение маркера доступа пользователя

Метод `GET /api/v3/users/${subjectId}/fedToken/${fedPointType}/${fedPointName}`

Получение актуального маркера доступа пользователя во внешнем поставщике идентификации с типом `${fedPointType}` и именем `${fedPointName}`. Маркер доступа считается актуальным, если время жизни больше минимально допустимого (по умолчанию 30 секунд). Если маркер доступа неактуален, но вместе с ним был сохранен маркер обновления, то происходит попытка обновления маркера доступа. В случае удачной попытки данный метод выдает новый маркер доступа.

Важно: Получение маркера возможно только для тех поставщиков, у которых *включена настройка* (страница 143) `Запоминать маркеры`.

Необходимые разрешения: `fed_tkn_any` или `fed_tkn_${fedPointType}_${fedPointName}`.

> **Примечание:** Для того чтобы приложение могло запросить маркер доступа, для него также должны быть *указаны* (страница 238) данные разрешения.

Возвращает

- `HTTP 404`: маркер доступа не найден.
- `HTTP 200` и JSON с информацией по маркерам доступа пользователя в случае успеха. Для каждого маркера передается ключ `sid`, значение маркера `token` и период действия `expiresOn` в формате Unix-time.
- `HTTP 401`: нет необходимого разрешения или неверный поставщик.

Пример

Запрос

```
GET /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/fedToken/tcs/tcs_1␣
↪HTTP/1.1
Authorization: Bearer m9tuVBNUnizkuwFnq95IXQm1XTplXLUFD1O5TUmGij4
Content-Type: application/json
```

Ответ

Список 68: Успех

```
{
    "da0c69c5-aef8-41e4-a37f-89c6d30abdfa": {
        "expiresOn": 1711125311,
        "token": "t.eFgoMik6regKsLjxfds1V0PlNEv_smx-W_x"
    },
    "00000000-1111-41e4-a37f-89c6d30abdfa": {
        "expiresOn": 1711125344,
        "token": "t.ddddddddddLjxfds1V0PlNEv_smx-W_x"
    }
}
```

Список 69: Нет необходимого разрешения

```
{
    "type":"security_error",
    "error":"bad_access_token",
    "desc":"No enough scopes or wrong subject Id"
}
```

События аудита

Метод `GET https://login.company.com/blitz/api/v3/users/{subjectId}/audit`

Получение списка событий безопасности, зарегистрированных на учетную запись пользователя.

Необходимые разрешения: `blitz_api_uaud` или `blitz_api_sys_uaud`.

URL-параметры

- `rql` – запрос фильтрации выводимых сведений в формате Resource Query Language[117] (RQL). Поддерживается фильтрация по атрибуту `ts` (время события).

  Операции:

  - `and` – одновременное выполнение поисковых условий;
  - `le` – проверка условия «меньше или равно»;
  - `ge` – проверка условия «больше или равно»;
  - `limit` – ограничение числа возвращаемых записей.

- `ua` - требуемый вид вывода сведений о `UserAgent` (атрибут `ua`). Варианты:

  - `none` – не возвращать UserAgent;

[117] https://github.com/kriszyp/rql

– `parsed` – возвращать UserAgent в разобранном виде (отдельно браузер и операционная система с указанием их версий);

Если параметр `ua` не указывать, то `UserAgent` (атрибут `ua`) вернется просто в виде строки.

Возвращает JSON, содержащий перечень событий аудита учетной записи за указанный период времени.

Примеры

Без парсинга сведений о UserAgent

Запрос

```
GET /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/audit?rql=and(ge(ts,
↪1637230238),le(ts,1637250238),limit(2)) HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Cache-Control: no-cache
```

Ответ


```
[
    {
        "sbj": "af583e70-fe39-407d-a87e-06cd0ec1830c",
        "ua": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) …",
        "ts": 1637250238015,
        "cAthM": "Basic",
        "ipCt": "Москва",
        "ipRad": 20,
        "cId": "test_app",
        "ip": 1406987879,
        "obj": "af583e70-fe39-407d-a87e-06cd0ec1830c",
        "ipSt": "Москва",
        "lpId": "test_app",
        "pid": "ddeebaba-2dc3-41bb-b539-7f0e472414a3",
        "ipLat": 55.7483,
        "prms": {
            "used_login": "test@yandex.ru",
            "auth_methods": "password",
            "authnDone": "true",
            "id_store": "389-ds"
        },
        "type": "login",
        "ipCtr": "Россия",
        "proc": "profile",
        "ipLng": 37.6171,
        "sid": "54914ac3-0d39-40d3-9617-92e0e7fe07ab"
    }
]
```

С парсингом сведений о UserAgent

Запрос

```
GET /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/audit?rql=and(ge(ts,
↪1637230238),le(ts,1637250238),limit(2))&ua=parsed HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Cache-Control: no-cache
```

Ответ


```
[
    {
        "sbj": "af583e70-fe39-407d-a87e-06cd0ec1830c",
        "ua": {
            "broName": "Chrome",
            "broVer": "109",
            "deviceType": "pc",
            "raw": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) …",
            "osName": "macOS",
            "osVer": "10.15.7"
        },
        "ts": 1637250238015,
        "cAthM": "Basic",
        "ipCt": "Москва",
        "ipRad": 20,
        "cId": "test_app",
        "ip": 1406987879,
        "obj": "af583e70-fe39-407d-a87e-06cd0ec1830c",
        "ipSt": "Москва",
        "lpId": "test_app",
        "pid": "ddeebaba-2dc3-41bb-b539-7f0e472414a3",
        "ipLat": 55.7483,
        "prms": {
            "used_login": "test@yandex.ru",
            "auth_methods": "password",
            "authnDone": "true",
            "id_store": "389-ds"
        },
        "type": "login",
        "ipCtr": "Россия",
        "proc": "profile",
        "ipLng": 37.6171,
        "sid": "54914ac3-0d39-40d3-9617-92e0e7fe07ab"
    }
]
```

Известные устройства и сессии

Список известных устройств

Метод `GET https://login.company.com/blitz/api/v3/users/{subjectId}/uas`

Получение списка устройств пользователя.

Необходимые разрешения: `blitz_api_uapps` или `blitz_api_sys_uapps`.

Возвращает JSON, содержащий перечень устройств пользователя.

Пример

Запрос

```
GET /blitz/api/v3/users/af583e70-fe39-407d-a87e-06cd0ec1830c/uas HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Cache-Control: no-cache
```

Ответ


```
[
    {
        "name": "Chrome 96",
        "lastUsed": 1637249978,
        "tp": "Browser",
        "os": "macOS 10.15.7",
        "newlyCreated": false,
        "deviceType": "pc",
        "latestIp": "172.25.0.1",
        "subjectId": "af583e70-fe39-407d-a87e-06cd0ec1830c",
        "id": "SHA256_Z0x284K3qv313WViRuPfV5rglhDuYqSn4ztdxVKMBec",
        "trusted": false,
        "cls": true,
        "deviceId": "738f5ce91f912ddd4a0cc5fefa9e8c63",
        "device": "PC"
    }
]
```


Удаление устройства из списка

Метод `DELETE https://login.company.com/blitz/api/v3/users/{subjectId}/uas/{id}`

Удаление устройства из числа запомненных. В качестве `id` нужно передать *полученный* (страница 481) идентификатор устройства.

Необходимые разрешения: `blitz_api_uapps_chg` или `blitz_api_sys_uapps_chg`.

Заголовки В пользовательском режиме необходимо передать заголовки с IP-адресом пользователя и `User-Agent`.

Пример

Список 70: Запрос

```
DELETE /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/uas/SHA256_
↪Z0x284K3qv313WViRuPfV5rglhDuYqSn4ztdxVKMBec HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)
```

Сброс сессий пользователя

Метод `POST https://login.company.com/blitz/api/v3/users/{subjectId}/sessions/reset`

Сброс сессий пользователя.

Необходимые разрешения: `blitz_api_usec_chg` или `blitz_api_sys_usec_chg`.

Заголовки

- В пользовательском режиме необходимо передать заголовки с IP-адресом пользователя и `User-Agent`.
- Если выход пользователя с текущего устройства/браузера является нежелательным, необходимо передать от приложения заголовок `IB-CI-UA-ID` с идентификатором текущего устройства, чтобы сохранить на нем сессию.

Совет: Идентификатор текущего устройства пользователя можно получить из *маркера идентификации* (страница 400).

Возвращает В случае успешного вызова - код `HTTP 204 No Content`.

Внимание: Сброс сессий приведет к аннулированию ранее полученных маркеров доступа и маркеров обновления текущего пользователя.

Примеры запросов

Список 71: Пользовательский режим

```
POST /blitz/api/v3/users/c574a512-3704-4576-bc3a-3fe28b636e85/sessions/reset HTTP/
↪1.1
Content-Type: application/json
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)…
Authorization: Bearer wzb…Tw
IB-CI-UA-ID: {SHA256}rVWFmwgRKWeW_flH4CA4yuW7OhKZ32Da94m0kzwWsVs
```

Список 72: Режим вызова сервиса системой

```
POST /blitz/api/v3/users/c574a512-3704-4576-bc3a-3fe28b636e85/sessions/reset HTTP/
↪1.1
Content-Type: application/json
Authorization: Bearer qwa…Ez
```

Контрольные вопросы

Проверка наличия вопроса

Метод `GET https://login.company.com/blitz/api/v3/users/{subjectId}/secQsn`

Проверка наличия у пользователя контрольного вопроса.

Необходимые разрешения: `blitz_api_usec` или `blitz_api_sys_usec`.

Возвращает

- Если контрольный вопрос задан - текст контрольного вопроса.
- Если контрольный вопрос не задан - `404 Not Found`.

Пример

Запрос

```
GET /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/secQsn HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Cache-Control: no-cache
```

Ответ

```
{
    "question": "Как звали вашего первого питомца"
}
```

Проверка ответа

Метод `POST https://login.company.com/blitz/api/v3/users/{subjectId}/secQsn/check`

Проверка правильности ответа на контрольный вопрос.

Необходимые разрешения: `blitz_api_usec` или `blitz_api_sys_usec`.

Тело запроса Контрольный вопрос (`question`) и ответ на него (`answer`).

Возвращает

- В случае успешной проверки вопроса и ответа - `204 No Content`.
- В противном случае - `400 Bad request`.

Пример

Запрос

```
POST /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/secQsn/check HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)…
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{
    "question": "Как звали вашего первого питомца",
    "answer": "Тигр"
}
```

Ошибка

Список 73: Не совпал контрольный вопрос

```
{
    "type": "process_error",
    "error": "wrong_security_answer",
    "desc": "security question not match"
}
```

Список 74: Не совпал ответ на контрольный вопрос

```
{
    "type": "process_error",
    "error": "wrong_security_answer",
    "desc": "security answer not match"
}
```

Список 75: Контрольный вопрос у пользователя не установлен

```
{
    "type": "process_error",
    "error": "wrong_security_answer",
    "desc": "security question not found"
}
```

Установка или изменение вопроса

Метод `POST https://login.company.com/blitz/api/v3/users/{subjectId}/secQsn`

Установка или изменение контрольного вопроса пользователя.

Необходимые разрешения: `blitz_api_sys_usec_chg` или `blitz_api_sys_usec_chg`.

Тело запроса Контрольный вопрос (`question`) и ответ на него (`answer`).

Возвращает В случае успешной установки контрольного вопроса - `204 No Content`.

Список 76: Пример запроса

```
POST /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/secQsn HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)…

{
    "question": "Как звали вашего первого питомца",
    "answer": "Тигр"
}
```

Удаление вопроса

Метод `DELETE https://login.company.com/blitz/api/v3/users/{subjectId}/secQsn`

Удаление контрольного вопроса из учетной записи пользователя.

Необходимые разрешения: `blitz_api_usec_chg` или `blitz_api_sys_usec_chg`.

Возвращает При успешном выполнении - `204 No Content`.

Список 77: Пример запроса

```
DELETE /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/secQsn HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)…
```

Выданные пользователем разрешения

Список разрешений

Метод `GET https://login.company.com/blitz/api/v3/users/{subjectId}/acls`

Получение списка выданных пользователем разрешений.

Необходимые разрешения: `blitz_api_usec` или `blitz_api_sys_usec`.

Возвращает JSON, содержащий перечень выданных пользователем разрешений.

Пример

Запрос

```
GET /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/acls HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Cache-Control: no-cache
```

Ответ

```
[
    {
        "id": "d2580c98 e584 4aad a591 97a8cf45cd2a_app1",
        "updated": 1552896932780,
        "client_id": "app1",
        "scopes": [
            "openid",
            "profile",
        ]
    }
]
```

Отзыв разрешения

Метод `DELETE https://login.company.com/blitz/api/v3/users/{subjectId}/acls/{acl_id}`

Отзыв выданного разрешения.

Необходимые разрешения: `blitz_api_usec_chg` или `blitz_api_sys_usec_chg`.

URL-параметры В качестве `acl_id` передается *полученный* (страница 485) идентификатор (`id`) разрешения.

Заголовки В пользовательском режиме необходимо передать заголовки с IP-адресом пользователя и `User-Agent`.

Пример

Список 78: Запрос

```
DELETE /blitz/api/v3/users/d25..2a/acls/d25..2a_app1 HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
```

Мобильные приложения

Список мобильных приложений

Метод `GET https://login.company.com/blitz/api/v3/users/{subjectId}/apps`

Получение списка привязанных мобильных приложений.

Необходимые разрешения: `blitz_api_uapps` или `blitz_api_sys_uapps`.

Возвращает JSON, содержащий перечень привязанных мобильных приложений.

Пример

Запрос

```
GET /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/apps HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Cache-Control: no-cache
```

Ответ

```
[
    {
        "id": "dyn~test_app~afae0cab-2649-482d-9832-5f73816afb59",
        "name": {
            "_default_": "Тестовое приложение (test_app)"
        },
        "availableScopes": [
            "openid",
            "profile"
        ],
        "softwareId": "test_app"
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    }
]
```

Отвязка от аккаунта мобильного приложения

`DELETE https://login.company.com/blitz/api/v3/users/{subjectId}/apps/{app_id}`

Отзыв выданного разрешения.

Необходимые разрешения: `blitz_api_uapps_chg` или `blitz_api_sys_uapps_chg`.

URL-параметры В качестве `app_id` передается *полученный* (страница 486) идентификатор (`id`) привязки приложения.

Заголовки В пользовательском режиме необходимо передать заголовки с IP-адресом пользователя и `User-Agent`.

Пример

Список 79: Запрос

```
DELETE /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/apps/d2580c98-e584-
↪4aad-a591-97a8cf45cd2a_app1 HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)...
```

Удаление учетной записи

Метод `DELETE https://login.company.com/blitz/api/v2/users/{subjectId}?instanceId={instanceId}`

Удаление учетной записи пользователя.

В `subjectId` передается идентификатор удаляемой учетной записи, в параметре `instanceId` - ссылка на удаляемую учетную запись. Чтобы узнать значение `instanceId` для пользователя, необходимо предварительно вызвать методом GET *сервис получения атрибутов* (страница 449) пользователя.

Пример

Список 80: Запрос

```
DELETE /blitz/api/v2/users/d..2a?instanceId=M…U HTTP/1.1
Authorization: Basic YXBwX2lkOmFwcF9zZWNyZXQ=
```

3.4.3 Группы пользователей

Внимание: Для вызова сервисов система должна получить маркер доступа на *системное разрешение* (страница 432) `blitz_groups` и включать его во все вызываемые сервисы.

Группы в Blitz Identity Provider описываются следующими атрибутами:

- `id` – идентификатор группы в Blitz Identity Provider;
- `name` – наименование группы пользователей.

Получение атрибутов группы по id

Метод `GET https://login.company.com/blitz/api/v2/grps/{id}`

Получение атрибутов группы, если известен `id` группы.

URL-параметры

- `profile` – имя профиля групп пользователей (например, `orgs`);
- `expand` – значение `true`, указывающее, что необходимо вернуть все атрибуты группы.

Пример

Запрос

```
GET /blitz/api/v2/grps/14339e8e-a665-4556-92f1-5c348eff6696?profile=orgs HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Cache-Control: no-cache
```

Ответ

```
{
    "instanceId": "Mzg…nU",
    "id": "14339e8e-a665-4556-92f1-5c348eff6696",
    "OGRN": "1234567890329",
    "INN": "7743151614",
    "name": "ООО Тестовая компания",
    "profile": "orgs"
}
```

Поиск группы по атрибуту

Метод `GET https://login.company.com/blitz/api/v2/grps`

Поиск группы по атрибуту и получение всех ее атрибутов, если неизвестен `id` группы.

URL-параметры

- `profile` – имя профиля групп пользователей;
- `rql` – поисковый запрос по атрибутам группы в формате Resource Query Language[118] (RQL).

  Операции:

  - `and` – одновременное выполнение поисковых условий;
  - `or` – альтернативное выполнение поисковых условий (например, поиск по разным атрибутам);
  - `eq` – проверка условия равенства;
  - `limit` – ограничение числа возвращаемых записей.
- `expand` (необязательный параметр):
  - `true`: включать в полученный ответ атрибуты групп;
  - `false`: вернуть только идентификаторы найденных групп.

Возвращает JSON, содержащий перечень групп, удовлетворяющих заданным поисковым условиям, с указанием их идентификатора (`id`), а также значения остальных атрибутов групп (в случае `expand=true`).

Пример

Запрос

Список 81: Поиск группы по ОГРН или ИНН

```
GET /blitz/api/v2/grps?profile=orgs&expand=true&rql=or(eq(OGRN,
↪string:1230123456789),eq(INN,string:7743151614)) HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Cache-Control: no-cache
```

Ответ

```
[
    {
        "instanceId": "Mzg5L…nU",
        "id": "14339e8e-a665-4556-92f1-5c348eff6696",
        "OGRN": "1234567890329",
        "INN": "7743151614",
        "name": "ООО Тестовая компания",
        "profile": "orgs"
    }
]
```

[118] https://github.com/kriszyp/rql

Создание группы

Метод `POST https://login.company.com/blitz/api/v2/grps`

Создание группы пользователей.

Тело запроса

- `profile` – имя профиля групп пользователей;
- `id` – уникальный идентификатор группы;
- остальные атрибуты группы и их значения.

Пример

Запрос

```
POST /blitz/api/v2/grps HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json

{
    "id":"95339e8e-a665-4556-92f1-5c348eff6696",
    "OGRN":"9876543210321",
    "INN":"5012345678",
    "name":"ООО Тестовая компания 2",
    "profile":"orgs"
}
```

Ответ

```
{
    "instanceId": "b3Jnc…dQ",
    "name": "ООО Тестовая компания 2",
    "OGRN": "9876543210321",
    "id": "95339e8e-a665-4556-92f1-5c348eff6696",
    "profile": "orgs",
    "INN": "5012345678"
}
```

Изменение атрибутов группы

Метод `POST https://login.company.com/blitz/api/v2/grps/{id}?profile=orgs`

Изменение атрибутов группы.

Тело запроса Новый набор атрибутов:

- `profile` – имя профиля групп (должно быть передано и в составе URL, и в теле запроса);
- `id` – идентификатор группы;
- остальные атрибуты группы и их значения.

Пример

Запрос

```
POST /blitz/api/v2/grps/5f7b0580-cd2e-4146-8fc5-6eb5a95c7b42?profile=orgs HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json

{
    "id": "5f7b0580-cd2e-4146-8fc5-6eb5a95c7b42",
    "OGRN": "1147746651733",
    "INN": "7715434658",
    "name": "Новое название",
    "profile": "orgs"
}
```

Ответ

```
{
    "instanceId": "Mzg5L…nU",
    "id": "5f7b0580-cd2e-4146-8fc5-6eb5a95c7b42",
    "OGRN": "1147746651733",
    "INN": "7715434658",
    "name": "Новое название",
    "profile": "orgs"
}
```

Ошибка

Список 82: Организация не существует

```
{
    "errors": [
        {
            "code": "group_not_found",
            "desc": "Group with '95339e8e-…97' id not found in '389-ds' LDAP group␣
↪store",
            "params": {}
        }
    ]
}
```

Удаление группы

Метод `DELETE https://login.company.com/blitz/api/v2/grps/{id}?profile=orgs`

Удаление группы.

Пример

Список 83: Запрос

```
DELETE /blitz/api/v2/grps/5f7b0580-cd2e-4146-8fc5-6eb5a95c7b42?profile=orgs HTTP/1.
↪1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
```

Получение списка пользователей в группе

Метод `GET https://login.company.com/blitz/api/v2/grps/{id}/members`

Получение списка пользователей из группы.

URL-параметры

- `profile` – имя профиля групп пользователей;
- `expand` (необязательный параметр):
  - `true`: включать в полученный ответ ФИО пользователя;
  - `false`: вернуть только идентификаторы пользователей.

Пример

Запрос

Список 84: expand=false

```
GET /blitz/api/v2/grps/14339e8e-a665-4556-92f1-5c348eff6696/members?profile=orgs&
↪expand=false HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Cache-Control: no-cache
```

Список 85: expand=true

```
GET /blitz/api/v2/grps/14339e8e-a665-4556-92f1-5c348eff6696/members?profile=orgs&
↪expand=true HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Cache-Control: no-cache
```

Ответ

Список 86: expand=false

```
[
    {
        "instanceId": "Mzg5L…J1",
        "subjectId": "d434b7d4-9816-460a-83aa-0a994226cbe7"
    },
    {
        "instanceId": "Mzg5L…J1",
        "subjectId": "2cafa5f4-bc84-4f6f-91aa-080da47975f0"
    }
]
```

Список 87: expand=true

```
[
    {
        "instanceId": "Mzg5L…J1",
        "family_name": "Иванов",
        "middle_name": "Иванович",
        "given_name": "Иван",
        "subjectId": "d434b7d4-9816-460a-83aa-0a994226cbe7"
    },
    {
        "instanceId": "Mzg5L…J1",
        "family_name": "Сергеев",
        "middle_name": "Сергеевич",
        "given_name": "Сергей",
        "subjectId": "2cafa5f4-bc84-4f6f-91aa-080da47975f0"
    }
]
```

Добавление пользователей

Метод `POST https://.../blitz/api/v2/grps/{id}/members/add?profile=orgs`

Добавление пользователей в группу.

Тело запроса Список добавляемых в группу пользователей с указанием их идентификаторов (`sub`) в атрибуте `subjectId`.

Запрос

```
POST /blitz/api/v2/grps/5f7b0580-cd2e-4146-8fc5-6eb5a95c7b42/members/add?
↪profile=orgs HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json

[
    {
        "subjectId": "45ff69f2-6c40-418f-a21d-cbe6f07b88c9"
    },
    {
        "subjectId": "cc8c4589-b2f8-40b8-b351-36d643808943"
    }
]
```

Ответ

```
[
    {
        "instanceId": "Mzg5L…J1",
        "storeId": "tam",
        "subjectId": "45ff69f2-6c40-418f-a21d-cbe6f07b88c9"
    },
    {
        "instanceId": "Nzg5L…J1",
        "storeId": "tam",
        "subjectId": "cc8c4589-b2f8-40b8-b351-36d643808943"
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    }
]
```

Ошибка

Список 88: Попытка добавить несуществующего пользователя

```
{
    "errors": [
        {
            "code": "user_not_found",
            "desc": "User with subjectId 'd2580c98-e584-4aad-a591-97a8cf45cd2q'␣
↪not found",
            "params": {}
        }
    ]
}
```

Список 89: Попытка добавить пользователя, который уже есть в группе

```
{
    "errors": [
        {
            "code": "some_members_already_in_group",
            "desc": "Some of adding members are already included in group",
            "params": {}
        }
    ]
}
```

Исключение пользователей

Метод `POST https://.../blitz/api/v2/grps/{id}/members/rm?profile=orgs`

Исключение пользователей из группы.

Тело запроса Список исключаемых из организации доверенных лиц с указанием их идентификаторов (`sub`) в атрибуте `subjectId`.

Запрос

```
POST /blitz/api/v2/grps/5f7b0580-cd2e-4146-8fc5-6eb5a95c7b42/members/rm?
↪profile=orgs HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json

[
    {
        "subjectId": "d2580c98-e584-4aad-a591-97a8cf45cd2a"
    }
]
```

Ответ


```
[
    {
        "instanceId": "Mzg5L…J1",
        "storeId": "389-ds",
        "subjectId": "d2580c98-e584-4aad-a591-97a8cf45cd2a"
    }
]
```


Ошибка

Список 90: Попытка удалить из группы пользователя, которого в ней уже нет


```
{
    "errors": [
        {
            "code": "some_members_not_in_group",
            "desc": "Some of removing members are not included in group",
            "params": {}
        }
    ]
}
```


Список 91: Попытка удалить несуществующего пользователя


```
{
    "errors": [
        {
            "code": "user_not_found",
            "desc": "User with subjectId 'd2580c98-e584-4aad-a591-97a8cf45cd2b'␣
↪not found",
            "params": {}
        }
    ]
}
```


3.4.4 Права доступа

Внимание: Для выполнения запросов по просмотру, назначению, отзыву прав доступа приложение должно получить маркер доступа с системным разрешением `blitz_rights_full_access`.

Совет: Для просмотра прав доступа пользователя, где он является субъектом, также можно использовать маркер доступа с пользовательским разрешением `blitz_user_rights`.

Право доступа назначается от субъекта доступа к объекту доступа.

Субъекты доступа:

- пользователи,
- приложения (префикс `its`).

Объекты доступа:

- пользователи,
- группы пользователей (префикс `grps`),
- приложения (префикс `its`).

Перечень прав пользователя

Метод `GET https://login.company.com/blitz/api/v3/rights/of/<sub>`

Получение прав доступа по субъекту доступа, являющемуся пользователем.

Примеры

Запрос

```
GET /blitz/api/v3/rights/of/BIP-1SEQ41A HTTP/1.1
Authorization: Bearer cNwIX…Nz
```

Ответ

Список 92: Пользователь `BIP-1SEQ41A` имеет право `ORG_ADMIN` к группе пользователей `1147746651733`, право `APP_ADMIN` к приложению `test_app2`, право `change_password` к учетной записи пользователя `BIP-3SGR7TA`

```
{
    "grps|1147746651733|orgs": {
        "ORG_ADMIN": [
            "set_from_api",
            "another_one_tag"
        ]
    },
    "its|test_app2": {
        "APP_ADMIN": [
            "set_from_api"
        ]
    },
    "BIP-3SGR7TA": {
        "change_password": [
            "parent"
        ]
    }
}
```

Перечень прав приложения

Метод `GET https://login.company.com/blitz/api/v3/rights/of/its/<app_id>`

Получение прав доступа по субъекту доступа, являющемуся приложением.

Примеры

Запрос

```
GET /blitz/api/v3/rights/of/its/test_app HTTP/1.1
Authorization: Bearer cNwIX…Nz
```

Ответ

Список 93: Приложение `test_app` имеет право `SYS_MON` к приложению `test_app2`, право `change_password` к учетной записи пользователя `BIP-3SGR7TA`, право `ORG_ADMIN` к группе пользователей `1147746651733`

```
{
    "its|test_app2": {
        "SYS_MON": [
            "set_from_api"
        ]
    },
    "BIP-3SGR7TA": {
        "change_password": [
            "set_from_api"
        ]
    },
    "grps|1147746651733|orgs": {
        "ORG_ADMIN": [
            "set_from_api"
        ]
    }
}
```

Права в отношении пользователя

Метод `GET https://login.company.com/blitz/api/v3/rights/on/<sub>`

Получение прав доступа по объекту доступа, являющемуся пользователем.

Примеры

Запрос

```
GET /blitz/api/v3/rights/on/BIP-3SGR7TA HTTP/1.1
Authorization: Bearer cNwIX…Nz
```

Ответ

Список 94: На учетную запись BIP-3SGR7TA у пользователя BIP-1SEQ41A и у приложения test_app есть право change_password

```
{
    "BIP 1SEQ41A": [
        "change_password"
    ],
    "its|test_app": [
        "change_password"
    ]
}
```

Права в отношении группы пользователей

Метод GET https://.../blitz/api/v3/rights/on/grps/<grp_id>?objectExt=<profile>

Получение прав доступа по объекту доступа, являющемуся группой.

Примеры

Запрос

```
GET /blitz/api/v3/rights/on/grps/1147746651733?objectExt=orgs HTTP/1.1
Authorization: Bearer cNwIX…Nz
```

Ответ

Список 95: На учетную запись группы `1147746651733` из профиля `orgs` у пользователя `BIP-1SEQ41A`, и у приложения `test_app` есть право `ORG_ADMIN`

```
{
    "BIP 1SEQ41A": [
        "ORG_ADMIN"
    ],
    "its|test_app": [
        "ORG_ADMIN"
    ]
}
```

Права в отношении приложения

Метод `GET https://login.company.com/blitz/api/v3/rights/on/its/<app_id>`

Получение прав доступа по объекту доступа, являющемуся приложением.

Примеры

Запрос

```
GET /blitz/api/v3/rights/on/its/test_app2 HTTP/1.1
Authorization: Bearer cNwIX…Nz
```

Ответ

Список 96: На учетную запись приложения `test_app2` у пользователя `BIP-1SEQ41A` есть право `APP_ADMIN`, и у приложения `test_app` есть право `SYS_MON`

```
{
    "BIP 1SEQ41A": [
        "APP_ADMIN"
    ],
    "its|test_app": [
        "SYS_MON"
    ]
}
```

Ошибка

Список 97: В случае если маркер доступа просрочен, сервис вернет ошибку `HTTP 401 Unauthorized` и JSON

```
{
    "type": "security_error",
    "error": "bad_access_token",
    "desc": "expired_access_token"
}
```

Назначение прав

Метод `PUT https://login.company.com/blitz/api/v3/rights`

Назначение прав доступа.

Тело запроса

- `subject` – идентификатор субъекта, которому назначается право (идентификатор пользователя или приложения);
- `subjectType` – тип субъекта. Параметр указывается только в случае назначения права приложению. В этом случае используется значение `its`;
- `object` – идентификатор объекта, на который назначается право (идентификатор пользователя, группы пользователей или приложения);
- `objectType` – тип объекта. Параметр указывается только в случае назначения права на группу пользователей (значение `grps`) или на приложение (значение `its`);
- `rights` – массив со списком назначаемых прав субъекту на объект;
- `tags` – массив со списком тэгов назначаемых прав.

Возвращает

- В случае успешного назначения права доступа - `HTTP 204 No Content`.
- Если маркер доступа просрочен - `HTTP 401 Unauthorized`.
- Если субъекта или объекта не существует - `HTTP 400 Bad Request`

Примеры

Запрос

Список 98: Назначение права доступа пользователю на другого пользователя

```
PUT /blitz/api/v3/rights HTTP/1.1
Authorization: Bearer cNwIXNz
Content-Type: application/json

{
    "subject": "BIP-1SEQ41A",
    "object": "BIP-3SGR7TA",
    "rights": ["change_password"],
    "tags": ["set_from_api"]
}
```

Список 99: Назначение права доступа пользователю на группу

```
PUT /blitz/api/v3/rights HTTP/1.1
Authorization: Bearer cNwIXNz
Content-Type: application/json

{
    "subject": "BIP-1SEQ41A",
    "object": "1147746651733",
    "objectType": "grps",
    "rights": ["ORG_ADMIN"],
    "tags": ["set_from_api"]
}
```

Список 100: Назначение права доступа пользователю на приложение

```
PUT /blitz/api/v3/rights HTTP/1.1
Authorization: Bearer cNwIXNz
Content-Type: application/json

{
    "subject": "BIP-1SEQ41A",
    "object": "test_app2",
    "objectType": "its",
    "rights": ["APP_ADMIN"],
    "tags": ["set_from_api"]
}
```

Список 101: Назначение права доступа приложению на пользователя

```
PUT /blitz/api/v3/rights HTTP/1.1
Authorization: Bearer cNwIXNz
Content-Type: application/json

{
    "subject": "test_app",
    "subjectType": "its",
    "object": "BIP-3SGR7TA",
    "rights": ["change_password"],
    "tags": ["set_from_api"]
}
```

Список 102: Назначение права доступа приложению на группу

```
PUT /blitz/api/v3/rights HTTP/1.1
Authorization: Bearer cNwIXNz
Content-Type: application/json

{
    "subject": "test_app",
    "subjectType": "its",
    "object": "1147746651733",
    "objectType": "grps",
    "rights": ["ORG_ADMIN"],
    "tags": ["set_from_api"]
}
```

Список 103: Назначение права доступа приложению на другое приложение

```
PUT /blitz/api/v3/rights HTTP/1.1
Authorization: Bearer cNwIXNz
Content-Type: application/json

{
    "subject": "test_app",
    "subjectType": "its",
    "object": "test_app2",
    "objectType": "its",
    "rights": ["SYS_MON"],
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "tags": ["set_from_api"]
}
```

Ошибка

Список 104: Маркер доступа просрочен

```
{
    "type": "security_error",
    "error": "bad_access_token",
    "desc": "expired_access_token"
}
```

Список 105: Назначаемого права не существует

```
{
    "type": "process_error",
    "error": "unknown_right",
    "desc": "The specified right is unknown",
    "params": {
        "right": "change_password1"
    }
}
```

Список 106: Указанного в качестве субъекта или объекта пользователя не существует

```
{
    "type": "process_error",
    "error": "unknown_user",
    "desc": "The specified user is unknown",
    "params": {
        "userId": "ivanov1"
    }
}
```

Список 107: Указанной в качестве объекта группы не существует

```
{
    "type": "process_error",
    "error": "unknown_group",
    "desc": "The specified group is unknown",
    "params": {
        "grpId": "1147746651734"
    }
}
```

Список 108: Указанного в качестве субъекта или объекта приложения не существует

```
{
    "type": "process_error",
    "error": "unknown_rp",
    "desc": "The specified relying party is unknown",
    "params": {
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        "rpId": "test_app3"
    }
}
```

Отзыв прав

Метод `DELETE https://login.company.com/blitz/api/v3/rights`

Отзыв права доступа.

Тело запроса

- `subject` – идентификатор субъекта, у которого отзывается право (идентификатор пользователя или приложения);
- `subjectType` – тип субъекта. Параметр указывается только в случае отзыва права у приложения. В этом случае используется значение `its`;
- `object` – идентификатор объекта, на который отзывается право (идентификатор пользователя, группы пользователей или приложения);
- `objectType` – тип объекта. Параметр указывается только в случае отзыва права на группу пользователей (значение `grps`) или на приложение (значение `its`);
- `rights` – массив со списком отзываемых прав субъекта на объект;
- `tags` – массив со списком тэгов отзываемых прав.

> **Предупреждение:** Если право доступа было назначено субъекту доступа на объект доступа с указанием нескольких тэгов, то для отзыва права доступа также необходимо указать все тэги. Если отзыв права доступа вызывается не с полным указанием тэгов, то при отзыве будут удалены только отзываемые тэги, а право доступа у субъекта доступа к объекту доступа останется, пока остается хотя бы один из тэгов.

Возвращает

- В случае успешного отзыва права доступа сервис вернет `HTTP 204 No Content`.
- Если маркер доступа просрочен - `HTTP 401 Unauthorized`.
- Если отзываемого права, субъекта или объекта не существует - `HTTP 400 Bad Request`

Примеры

Запрос

Список 109: Отзыв права доступа пользователю на другого пользователя

```
DELETE /blitz/api/v3/rights HTTP/1.1
Authorization: Bearer cNwIXNz
Content-Type: application/json

{
    "subject": "BIP-1SEQ41A",
    "object": "BIP-3SGR7TA",
    "rights": ["change_password"],
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "tags": ["set_from_api"]
}
```

Список 110: Отзыв права доступа пользователю на группу

```
DELETE /blitz/api/v3/rights HTTP/1.1
Authorization: Bearer cNwIXNz
Content-Type: application/json

{
    "subject": "BIP-1SEQ41A",
    "object": "1147746651733",
    "objectType": "grps",
    "rights": ["ORG_ADMIN"],
    "tags": ["set_from_api"]
}
```

Список 111: Отзыв права доступа пользователю на приложение

```
DELETE /blitz/api/v3/rights HTTP/1.1
Authorization: Bearer cNwIXNz
Content-Type: application/json

{
    "subject": "BIP-1SEQ41A",
    "object": "test_app2",
    "objectType": "its",
    "rights": ["APP_ADMIN"],
    "tags": ["set_from_api"]
}
```

Список 112: Отзыв права доступа приложению на пользователя

```
DELETE /blitz/api/v3/rights HTTP/1.1
Authorization: Bearer cNwIXNz
Content-Type: application/json

{
    "subject": "test_app",
    "subjectType": "its",
    "object": "BIP-3SGR7TA",
    "rights": ["change_password"],
    "tags": ["set_from_api"]
}
```

Список 113: Отзыв права доступа приложению на группу

```
DELETE /blitz/api/v3/rights HTTP/1.1
Authorization: Bearer cNwIXNz
Content-Type: application/json

{
    "subject": "test_app",
    "subjectType": "its",
    "object": "1147746651733",
    "objectType": "grps",
    "rights": ["ORG_ADMIN"],
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "tags": ["set_from_api"]
}
```

Список 114: Отзыв права доступа приложению на другое приложение

```
DELETE /blitz/api/v3/rights HTTP/1.1
Authorization: Bearer cNwIXNz
Content-Type: application/json

{
    "subject": "test_app",
    "subjectType": "its",
    "object": "test_app2",
    "objectType": "its",
    "rights": ["SYS_MON"],
    "tags": ["set_from_api"]
}
```

Ответ

Список 115: Маркер доступа просрочен

```
{
    "type": "security_error",
    "error": "bad_access_token",
    "desc": "expired_access_token"
}
```

Список 116: Отзываемого права не существует

```
{
    "type": "process_error",
    "error": "unknown_right",
    "desc": "The specified right is unknown",
    "params": {
        "right": "change_password1"
    }
}
```

Список 117: Указанного в качестве субъекта или объекта пользователя не существует

```
{
    "type": "process_error",
    "error": "unknown_user",
    "desc": "The specified user is unknown",
    "params": {
        "userId": "ivanov1"
    }
}
```

Список 118: Указанной в качестве объекта группы не существует

```
{
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "type": "process_error",
    "error": "unknown_group",
    "desc": "The specified group is unknown",
    "params": {
        "grpId": "1147746651734"
    }
}
```

Список 119: Указанного в качестве субъекта или объекта приложения не существует

```
{
    "type": "process_error",
    "error": "unknown_rp",
    "desc": "The specified relying party is unknown",
    "params": {
        "rpId": "test_app3"
    }
}
```

Права ведущего пользователя в отношении ведомого

Метод `POST https://login.company.com/blitz/api/v3/users/rights/change`

Назначение и отзыв права ведущего пользователя в отношении ведомого пользователя.

Внимание: Запрос на отзыв прав может быть выполнен приложением не только с использованием пользовательского маркера доступа, полученного на разрешение с именем `blitz_user_rights`, но и с использованием системного маркера доступа, полученного на разрешение с именем `blitz_rm_rights`. В этом случае запрос на отзыв может включать `subject` любых пользователей (для отзыва у пользователя права не потребуется, чтобы именно этот пользователь осуществлял вход в систему и получал маркер доступа – система может отзывать права любого пользователя).

Заголовки В запрос должен быть добавлен заголовок с маркером доступа на разрешение с именем `blitz_user_rights`, полученным учетной записью ведущего пользователя.

Тело запроса

Назначение прав

Заполненный блок `update` с перечнем прав, которые должны быть добавлены в результате выполнения операции.

Каждое право описывается параметрами:

- `subject` – идентификатор (`sub`) учетной записи ведущего пользователя;
- `object` – идентификатор (`sub`) учетной записи ведомого пользователя;
- `rights` – перечень прав в виде массива, который получает учетная запись ведущего пользователя в отношении учетной записи ведомого пользователя. Например, для права менять пароль от учетной записи должно быть указано право `change_password` (смена пароля), а для права менять атрибуты должно быть указано право `change_attrs` (смена атрибутов);
- `tags` – перечень тегов, указывающих на основания, по которым данный пользователь получил права.

Отзыв прав

Заполненный блок `delete` с перечнем прав, которые должны быть отозваны в результате выполнения операции.

Каждое право описывается параметрами:

- `subject` – идентификатор (`sub`) учетной записи ведущего пользователя;
- `object` – идентификатор (`sub`) учетной записи ведомого пользователя;
- `rights` – перечень прав в виде массива, которые отзываются у ведущей учетной записи в отношении ведомой учетной записи;
- `tags` – перечень тегов, указывающих на основания, по которым данный пользователь получил права.

Если при выполнении запроса права не назначаются или не отзываются, то в теле запроса должен соответственно присутствовать или пустой блок `update`, или пустой блок `delete`. В одном запросе может быть указано сразу несколько назначаемых/отзываемых прав, но в качестве субъекта (`subject`) должен быть указан только тот пользователь, на которого был получен маркер доступа, используемый для вызова сервиса.

Примеры

Запрос

Список 120: Назначение прав

```
POST /blitz/api/v3/users/rights/change HTTP/1.1
Authorization: Bearer cNwIXTg
Content-Type: application/json

{
   "update":[
      {
         "subject":"6561d0d9-5583-4bb5-a681-b591358e5fcd",
         "object":"5cffd68f-2cb8-4f7a-b0f3-9fa69a1fbbcd",
         "rights":[
            "change_password"
         ],
         "tags":[
            "parent"
         ]
      },
      {
         "subject":"6561d0d9-5583-4bb5-a681-b591358e5fcd",
         "object":"b855957d-bf24-48d4-bb63-cce4f5064590d",
         "rights":[
            "change_password"
         ],
         "tags":[
            "parent"
         ]
      }
   ],
   "delete":[
   ]
}
```

Список 121: Отзыв прав

```
POST /blitz/api/v3/users/rights/change HTTP/1.1
Authorization: Bearer cNwIXTg
Content-Type: application/json

{
   "update":[
   ],
   "delete":[
      {
         "subject":"b855957d-bf24-48d4-bb63-cce4f5064590d",
         "object":"5cffd68f-2cb8-4f7a-b0f3-9fa69a1fbbcd",
         "rights":[
            "change_password"
         ],
         "tags":[
            "parent"
         ]
      }
   ]
}
```

Ошибка

Список 122: В случае ошибки запрос отклоняется целиком и возвращается перечень возникших ошибок

```
{
   "errors" : [
      {
         "code" : "validation_error",
         "params" : {},
         "desc" : "(For subject 'dea75b73-a2ba-4b60-a41c-bb640968826b') Incorrect␣
↪right '' to object '5cffd68f-2cb8-4f7a-b0f3-9fa69a1fbbcd'"
      },
      {
         "params" : {},
         "code" : "validation_error",
         "desc" : "(For subject 'dea75b73-a2ba-4b60-a41c-bb640968826b') Incorrect␣
↪tag '' for right 'write' to object '5cffd68f-2cb8-4f7a-b0f3-9fa69a1fbbcd'"
      },
      {
         "desc" : "(For subject 'dea75b73-a2ba-4b60-a41c-bb640968826b') Incorrect␣
↪object ''",
         "code" : "validation_error",
         "params" : {}
      },
      {
         "desc" : "Incorrect subject ''",
         "code" : "validation_error",
         "params" : {}
      }
   ]
}
```

3.5 Расширенные возможности

3.5.1 Дополнительный метод аутентификации

Blitz Identity Provider позволяет подключить собственный разработанный метод аутентификации. Для этого система, выступающая в качестве поставщика такого метода аутентификации, должна:

- предоставить обработчик запроса на аутентификацию;
- передать в Blitz Identity Provider результат аутентификации;
- предоставить сервис проверки применимости метода аутентификации Опционально.

В Blitz Identity Provider разработанный метод аутентификации нужно зарегистрировать как *внешний метод аутентификации* (страница 141).

Сервис обработчика запроса

Взаимодействие Blitz Identity Provider с сервисом обработчика запроса на аутентификацию выполняется следующим образом:

1. Сервис обработчика представляет собой URL для приема HTTP-запросов от Blitz Identity Provider. При запросе на аутентификацию Blitz Identity Provider будет делать запрос методом POST по данному адресу.

   В теле запроса Blitz Identity Provider в формате JSON передаст следующие данные:

   - идентификатор запроса (`id`);
   - утверждения, характеризующие пользователя (`claims`) – опционально, только при вызове в качестве второго фактора;
   - идентификатор системы, запросивший вход (`rpId`);
   - идентификатор контекста аутентификации (`loginContextId`);
   - данные о запросе (`request`), включающий в себя заголовки (`headers`), IP-адрес пользователя (`remoteAddress`), адрес метода (`uri`), перечень `cookie` (`cookies`) и `User Agent` пользователя (`userAgent`).

Список 123: Пример тела запроса

```
{
    "id": "a9692091-4613-41aa-91d2-9a71a3fc2e07",
    "claims": {},
    "rpId": "_blitz_profile",
    "loginContextId": "4502aa51-f28c-4a64-951c-5ab1e77b1294",
    "request": {
      "headers": {},
      "remoteAddress": "172.25.0.1",
      "uri": "/blitz/login/methods2/outside_test",
      "cookies": {},
      "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:80.0)…"
    }
}
```

2. На стороне поставщика внешнего метода необходимо предусмотреть обработку запроса Blitz Identity Provider. В результате внешний метод должен вернуть:

   - Если аутентификация возможна - HTTP-ответ для выполнения в браузере пользователя, который, например, содержит код HTML-страницы или инициирует редирект браузера на необходимую страницу внешнего метода.
   - Ошибку в случае невозможности провести аутентификацию пользователя.

Требования к обработке запроса Blitz Identity Provider

HTTP-ответ

- ответ должен включать в себя установку cookie (на общий домен Blitz Identity Provider и внешнего метода);
- название cookie должно быть предварительно зарегистрировано в Blitz Identity Provider;
- в качестве значения cookie должен быть использован идентификатор сессии, сгенерированный внешним методом.

Список 124: Пример HTTP-ответа с редиректом и установкой cookie

```
HTTP/1.1 302 Found
Location: https://login.company.com/blitz/begin?id=a9692091-4613-41aa-91d2
Set-Cookie: Bmr=YTk2OTIwOTEtNDYxMy00MWFhLTkxZDItOWE3MWEzZmMyZTA3;␣
↪Domain=company.com; path=/blitz; Secure; HttpOnly
```

Важно: При прохождении внешнего метода поставщик должен проверить, что значение `cookie` для данного запроса не было изменено.

Ошибка

Рекомендуемые коды возврата:

Код ответа HTTP	Значение ответа	Описание ответа
`200`	`OK`	Инициирование внешнего метода посредством отображения контента страницы
`302`	`Found`	Инициирование внешнего метода посредством редиректа
`400`	`Bad Request`	Отсутствуют обязательные параметры запроса
`500`	`Internal Server Error`	Внутренняя ошибка обработки входящего запроса

Передача результата аутентификации

После прохождения внешнего метода поставщик должен выполнить следующие действия:

1. Серверная часть поставщика должна вызвать Blitz Identity Provider методом POST по адресу:

```
https://login.company.com/blitz/login/methods/outside/save?methodName=outside\_
↪{name}
```

В данном запросе `name` – это имя внешнего метода, присвоенное ему в Blitz Identity Provider при регистрации.

Тело запроса

Успешная аутентификация

В случае успеха аутентификации в теле запроса должны быть указаны:

- идентификатор запроса (`id`);
- `extSessionId` – идентификатор сессии, сгенерированный внешним методом. Идентификатор должен совпадать со значением, переданным в исходном запросе в cookie;
- `claims` – перечень утверждений, которыми нужно обогатить сессию пользователя. Перечень может быть пустым;
- `subjectId` – идентификатор пользователя (только для первого фактора; при вызове внешнего метода в качестве второго фактора нельзя передавать идентификатор пользователя);
- `loginContextId` – идентификатор контекста аутентификации, соответствующий исходному запросу.

Список 125: Пример запроса

```
POST /blitz/login/methods/outside/save?methodName=outside_test HTTP/1.1
Content-Type: application/json

{
   "id": "426b5139-e4f7-41e6-a206-9503de6f34dd",
   "extSessionId": "YTk2OTIwOTEtNDYxMy00MWFhLTkxZDItOWE3MWEzZmMyZTA3",
   "claims": {},
   "loginContextId": "3ca4d1f0-654a-4665-be98-d105ab6ec35d",
   "subjectId": "2db787c7-6e37-4018-abe9-2bea1011c047"
}
```

Ошибка аутентификации

В случае ошибки в теле запроса должны быть указаны:

- `id` – идентификатор запроса;
- `extSessionId` – идентификатор сессии, сгенерированный внешним методом. Идентификатор должен совпадать со значением, переданным в исходном запросе в cookie;
- `error` – код ошибки;
- `msg` – текстовое описание ошибки (опционально).

Список 126: Пример запроса

```
POST /blitz/login/methods/outside/save?methodName=outside_test HTTP/1.1
Content-Type: application/json

{
   "id": "426b5139-e4f7-41e6-a206-9503de6f34dd",
   "extSessionId": "YTk2OTIwOTEtNDYxMy00MWFhLTkxZDItOWE3MWEzZmMyZTA3",
   "error": "not_found",
   "msg": "User not found"
}
```

В случае сохранения результатов аутентификации (как успешной, так и неуспешной) Blitz Identity Provider возвращает ответ `HTTP 200 OK`.

2. Браузерная часть поставщика должна обеспечить перенаправление пользователя обратно в Blitz Identity Provider. Для этого необходимо перенаправить браузер по адресу:

```
https://login.company.com/blitz/login/methods/outside/callback?
↪methodName=outside\_{name}
```

В данном запросе `name` – это имя внешнего метода, присвоенное ему в Blitz Identity Provider при регистрации.

Сервис проверки метода

Сервис проверки применимости метода аутентификации представляет собой URL для приема HTTP-запросов от Blitz Identity Provider. До запроса на аутентификацию Blitz Identity Provider будет делать запрос методом POST по данному адресу, передав в теле в формате JSON те же данные, что и в запросе на аутентификацию.

В качестве ответа внешний метод должен вернуть JSON со следующими атрибутами:

- идентификатор запроса (`id`);
- результат проверки применимости (`result`), принимающий значение либо `true` (метод применим) или `false` (метод неприменим);
- идентификатор контекста аутентификации (`loginContextId`), соответствующий запросу.

Если сервис вернет `false` в качестве результата проверки применимости, то далее Blitz Identity Provider не будет выполнять запрос на аутентификацию для данного пользователя.

3.5.2 Вызов вспомогательного приложения в момент входа

В момент входа Blitz Identity Provider может вызвать вспомогательное приложение, которое выполнит дополнительные операции (например, покажет пользователю информационное сообщение или запросит актуализацию сведений), после чего вернет пользователя в Blitz Identity Provider для последующего входа в целевое приложение.

С технической точки зрения вспомогательное приложение должно выполнять следующие действия:

- обработка запроса на открытие вспомогательного приложения,
- возвращение пользователя в Blitz Identity Provider после окончания обработки.

Запрос об открытии приложения

Прием запроса о вызове вспомогательного приложения происходит следующим образом:

1. Переход во вспомогательное приложение происходит посредством перенаправления пользователя на предоставленную приложением ссылку. Ссылка в качестве параметра будет содержать код авторизации (`code`).

Список 127: Пример ссылки для инициирования запроса

```
https://<app_hostname>/?lang=ru&theme=default&code=0Tj…qw
```

2. Приложение должно обменять код авторизации на маркер доступа согласно спецификации OAuth 2.0. Маркер доступа будет использован для получения идентификатора сессии, чтобы вернуть пользователя в Blitz Identity Provider, а также данных пользователя при необходимости.

Пример

Запрос

```
curl -k -d "grant_type=authorization_code&redirect_uri=https%3A%2F%2Fapp.
↪company.com%2F&client_id=app&client_secret=EW…l0&code=0Tj…qw" -X POST https:/
↪/login.company.com/blitz/oauth/te
```

Полученный маркер доступа

```
{
    "access_token": "ey…J9.ey…n0.Wa…Pw",
    "token_type": "Bearer",
    "expires_in": 3600,
    "scope": "profile"
}
```

Важно: Вспомогательное приложение должно быть предварительно зарегистрировано в Blitz Identity Provider с учетом следующих особенностей:

- должен быть указан предопределенный URL возврата, именно он далее должен быть использован для получения токена;
- должны быть настроены разрешения по умолчания (`scope`), именно они определяют объем данных, получаемых вспомогательным приложением.

Возврат пользователя в Blitz Identity Provider

Возврат пользователя в Blitz Identity Provider производится следующим образом:

1. Выполнив необходимые действия (например, показав пользователю информационное сообщение), вспомогательное приложение должно вернуть пользователя в Blitz Identity Provider. Для этого необходимо декодировать полученный маркер доступа, полученный в формате JWT, и извлечь из него утверждение с сессией пользователя (`sessionId`).

Список 128: Пример тела декодированного `access_token`

```
{
    "scope": "blitz_api_user blitz_api_user_chg blitz_api_usec_chg",
    "jti": "kfP…jA",
    "client_id": "app",
    "exp": 1631026605,
    "sessionId": "ce9f3109-ac79-46b4-b277-099ff1aa1ff0",
    "iat": 1631023005,
    "sub": "8b970179-e141-43b9-b9d5-25997be99261",
    "aud": [
        "app"
    ],
    "crid": "u9th2LzMXZdwb3rRmI3Paw",
    "iss": "https://login.company.com/blitz"
}
```

2. После декодирования маркера доступа вспомогательное приложение должно сделать POST-запрос на URL обработчика завершения аутентификации Blitz Identity Provider `/login/pipe/save/<sessionId>`. В теле запроса может быть указан набор утверждений (`claims`), которые следует добавить в сессию пользователя, либо информация об ошибке (`error`).

Список 129: Пример запроса

```
curl -v --location --request POST 'https://login.company.com/blitz/login/pipe/
↪save/ce9f3109-ac79-46b4-b277-099ff1aa1ff0' \
--header 'Content-Type: application/json' \
--header 'Authorization: Basic Z2…ww' \
--data-raw '{"claims":{"org_id":"12345678"}}'
```

3. В случае успеха Blitz Identity Provider вернет `HTTP 204 No Content`. Получив его, вспомогательное приложение должно вернуть браузер пользователя по адресу `/login/pipe/callback`, чтобы пользователь завершил вход в целевое приложение.

Список 130: Пример ссылки для перенаправления

```
https://login.company.com/blitz/login/pipe/callback
```

3.5.3 API администрирования

Администрировать Blitz Identity Provider можно с помощью:

- консоли управления;
- конфигурационных файлов;
- административные REST-сервисов.

Административные REST-сервисы в Blitz Identity Provider в текущей версии позволяют выполнять следующие действия:

- регистрация приложений;
- получение настроек приложений;
- изменение настроек приложений;
- удаление приложений.

Административные REST-сервисы доступны по адресу:

`https://login.company.com/blitz/admin/api/v3/....`

Для включения административных сервисов предварительно должны быть сделаны настройки на веб-сервере, используемом Blitz Identity Provider. Не рекомендуется публиковать административные REST-сервисы в сети Интернет.

Пример блока location в настройках веб-сервера nginx для включения доступности административных REST-сервисов:

```
location /blitz/admin/api {
    proxy_intercept_errors off;
    proxy_pass  http://blitz-console/blitz/admin/api;
}
```

Доступ к административным REST-сервисам регулируется с помощью разрешений (scope), приведенных в таблице:

Разрешения (scope) для административных REST API

№	Разрешение	Название	Описание
1.	blitz_api_sys_app	Разрешение на чтение настроек приложений	Для использования сервиса GET /blitz/admin/api/v3/app/{appId}
2.	blitz_api_sys_app_chg	Разрешение на внесение изменений в настройки приложений	Для использования сервисов: PUT /blitz/admin/api/v3/app/{appId} POST /blitz/admin/api/v3/app/{appId} DELETE /blitz/admin/api/v3/app/{appId}

Чтобы получить маркер доступа на системное разрешение, приложение должно выполнить запрос методом POST на URL для получения маркера (`https://login.company.com/blitz/oauth/te`). Запрос должен содержать заголовок `Authorization` со значением `Basic {secret}`, где `secret` – это `client_id:client_secret` (например, `app:topsecret`) в формате Base64.

Пример заголовка:

```
Authorization: Basic YWlzOm…XQ=
```

Тело запроса должно содержать следующие параметры:

- `grant_type` – принимает значение `client_credentials`;
- `scope` – запрашиваемое системное разрешение.

Пример запроса:

```
POST blitz/oauth/te HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Authorization: Basic ZG5ld…lg

grant_type=client_credentials&scope=blitz_api_sys_app+blitz_api_sys_app_chg
```

В ответ приложение получит маркер доступа (`access_token`), время его жизни (`expires_in`) и тип маркера (`token_type`). Возможные ошибки при вызове `/oauth/te` соответствуют RFC 6749[119].

Пример ответа с успешным выполнением запроса:

```
{
   "access_token": "QFiJ9mPgERPuusd36mQvD4mfzYolH_CmuddAJ3YKTOI",
   "expires_in": 3600,
   "scope": "blitz_api_sys_app blitz_api_sys_app_chg",
   "token_type": "Bearer"
}
```

Рекомендуется, чтобы приложение кэшировало полученный маркер доступа для многократного использования на время, немного меньшее, чем параметр `expires_in`, после чего осуществляло получение нового маркера доступа для обновления в кэше.

Если приложение попытается вызвать с просроченным маркером доступа соответствующий ему REST-сервис, то получит ошибку `HTTP 401 Unauthorized`.

[119] https://tools.ietf.org/html/rfc6749#section-5.2

Получение настроек приложений

Для получения настроек приложения по его идентификатору необходимо методом GET вызвать сервис по адресу `https://login.company.com/blitz/admin/api/v3/app/{appId}`.

Необходимые разрешения: `blitz_api_sys_app`.

В результате выполнения запроса Blitz Identity Provider вернет JSON, содержащий настройки приложения.

Пример запроса:

```
GET /blitz/admin/api/v3/app/test-app HTTP/1.1
Authorization: Bearer cNw…Nz
```

Пример ответа:

```
HTTP/2 200
…
content-type: application/json
etag: 96_1658847045000

{
    "name":"…",
    "tags": [
        "tag1",
        "tag2"
    ],
    "domain":"…",
    "startPageUrl":"…",
    "oauth": {
        "clientSecret":"…",
        "redirectUriPrefixes":["…"],
        "predefinedRedirectUri":"…",
        "availableScopes":["…","…"],
        "defaultScopes":["…"],
        "enabled":true,
        "autoConsent":true,
        "idToken":{"claims":["…"]},
        "accessTokenTtl":3600,
        "defaultAccessType":"online",
        "refreshTokenTtl":86400,
        "dynReg":{
            "isAllow":true,
            "allowedPlainJsonClaims":["device_type"]
        },
        "pixyMandatory":true,
        "deviceGrant": {
            "userCodeFormat":"[0-9]{3,3}-[0-9]{3,3}-[0-9]{3,3}",
            "userCodeTtl":120,
            "verificationUrl":"…",
            "useCompleteUri":true
        },
        "teAuthMethod":"client_secret_basic",
        "grantTypes":["authorization_code","client_credentials"],
        "responseTypes":["code"],
        "extraClientSecret":"…",
        "accessTokenFormat":"jwt",
        "logout": {
            "logoutAutoConsent":false,
            "logoutUriPrefixes":["…"],
            "predefinedLogoutUri":"…",
            "frontchannelLogoutUri":"…",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
            "frontchannelLogoutSessionRequired":true,
            "backchannelLogoutUri":"…"
        }
    },
    "simple": {
        "ssl":true,
        "formSelector":"…",
        "loginSelector":"…",
        "logoutUrl":"…",
        "postLogoutUrl":"…"
    },
    "rest": {
        "Basic":{"pswd":"…"},
        "TLS":[]
    },
    "theme":"default",
    "saml": {
        "spMetadata":"…",
        "spAttributeFilterPolicy": {
            "id":"test-app",
            "attributeRules":[{"attr":"…","isPermitted":true}]
        },
        "saml2SSOProfile": {
            "signAssertions":"always",
            "encryptAssertions":"always",
            "encryptNameIds":"always",
            "includeAttributeStatement":true
        }
    }
}
```

Содержимое ответа может отличаться в зависимости от заданных для приложения настроек и сконфигурированных протоколов подключения. Блоки `saml`, `oauth`, `simple`, `rest` могут отсутствовать, если соответствующие протоколы для приложения не настроены.

В ответе сервиса присутствует заголовок `etag`. Значение из этого заголовка следует использовать в заголовке `If-Match`, если планируется после получения настроек приложения вызывать сервисы регистрации приложения, редактирования настроек приложения или удаления приложения. С помощью `etag` Blitz Identity Provider проверяет, что между последним получением `etag` и вызовом операции изменения настроек с `If-Match` не выполнялись какие-либо еще изменения в конфигурационном файле на сервере в параллельных сеансах (оптимистичное блокирование).

При использовании SAML в настройке `spMetadata` будет находиться закодированный в Base64URL файл метаданных для приложения (Service Provider Metadata).

Имена возвращаемых сервисом настроек соответствуют именам в конфигурационном файле `blitz.conf`.

Если настройки приложения по переданному `appId` не будут найдены, то сервер Blitz Identity Provider вернет ошибку `HTTP 404 Not found`.

Регистрация приложения

Для регистрации приложения необходимо выполнить запрос методом PUT по адресу `https://login.company.com/blitz/admin/api/v3/app/{appId}`.

Необходимые разрешения: `blitz_api_sys_app_chg`.

В запрос может быть (опционально) добавлен заголовок `If-Match`, содержащий последнее полученное от сервера значение `etag`.

Тело запроса должно содержать значения настроек регистрируемого приложения.

Пример запроса:

```
PUT /blitz/admin/api/v3/app/test-app2 HTTP/1.1
Authorization: Bearer cNw…Nz
Content-Type: application/json
If-Match: 98_1658857264000

{
    "name":"…",
    "tags": [
        "tag1",
        "tag2"
    ],
    "domain":"…",
    "startPageUrl":"…",
    "oauth": {
        "clientSecret":"…",
        "redirectUriPrefixes":["…"],
        "predefinedRedirectUri":"…",
        "availableScopes":["…","…"],
        "defaultScopes":["…"],
        "enabled":true,
        "autoConsent":true,
        "idToken":{"claims":["…"]},
        "accessTokenTtl":3600,
        "defaultAccessType":"online",
        "refreshTokenTtl":86400,
        "dynReg":{
            "isAllow":true,
            "allowedPlainJsonClaims":["device_type"]
        },
        "pixyMandatory":true,
        "deviceGrant": {
            "userCodeFormat":"[0-9]{3,3}-[0-9]{3,3}-[0-9]{3,3}",
            "userCodeTtl":120,
            "verificationUrl":"…",
            "useCompleteUri":true
        },
        "teAuthMethod":"client_secret_basic",
        "grantTypes":["authorization_code","client_credentials"],
        "responseTypes":["code"],
        "extraClientSecret":"…",
        "accessTokenFormat":"jwt",
        "logout": {
            "logoutAutoConsent":false,
            "logoutUriPrefixes":["…"],
            "predefinedLogoutUri":"…",
            "frontchannelLogoutUri":"…",
            "frontchannelLogoutSessionRequired":true,
            "backchannelLogoutUri":"…"
        }
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    },
    "simple": {
        "ssl":true,
        "formSelector":"…",
        "loginSelector":"…",
        "logoutUrl":"…",
        "postLogoutUrl":"…"
    },
    "rest": {
        "Basic":{"pswd":"…"},
        "TLS":[]
    },
    "theme":"default",
    "saml": {
        "spMetadata":"…",
        "spAttributeFilterPolicy": {
            "id":"…",
            "attributeRules":[{"attr":"…","isPermitted":true}]
        },
        "saml2SSOProfile": {
            "signAssertions":"always",
            "encryptAssertions":"always",
            "encryptNameIds":"always",
            "includeAttributeStatement":true
        }
    }
}
```

При регистрации приложения, работающего по SAML, нужно учесть следующие особенности:

- в `spMetadata` нужно передавать содержимое метаданных приложения, закодированное в формате Base64URL.
- в настройку `id` в `spAttributeFilterPolicy` необходимо передать тот же `id`, что передан в URL в качестве `appId`.

Если регистрация успешна, то сервер вернет `HTTP 200`, актуальные данные приложения и актуальное значение `etag`.

Пример ответа:

```
HTTP/2 200
…
content-type: application/json
etag: 99_1658857631000

{
    "id":"test-app2",
    "name":"…",
    …
    "oauth": {
        …
    },
    …
}
```

Если при регистрации приложения будет обнаружено, что данные в конфигурационном файле на сервере были изменены между получением etag и вызовом регистрации, то сервер вернет ответ с кодом `HTTP 412 Precondition Failed` и телом ошибки:

```
{
   "type":"process_error",
   "error":"cas_mismatch",
   "desc":"cas_mismatch"
}
```

Если при регистрации приложения возникла ошибка, то сервер вернет ответ с кодом `HTTP 400 Bad Request` с описанием ошибки.

Пример ответа с ошибкой регистрации:

```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "json.error.mandatory.field",
            "desc": "json.error.expected.array",
            "pos": "oauth.redirectUriPrefixes"
        },
        …
    ]
}
```

Изменение настроек приложения

Для изменения настроек приложения необходимо выполнить запрос методом POST по адресу `https://login.company.com/blitz/admin/api/v3/app/{appId}`.

Необходимые разрешения: `blitz_api_sys_app_chg`.

В запрос должен быть добавлен заголовок `If-Match`, содержащий последнее полученное от сервера значение `etag`.

Тело запроса должно содержать значения изменяемых настроек приложения после редактирования. Должна быть передана вся ветка с изменяемым параметром. Например, если параметр находится на третьем уровне, то нужно также прислать его родительские параметры на первом и втором уровнях. Для того чтобы удалить параметр, необходимо прислать всю ветку со значением `null` для этого параметра.

Пример запроса изменения метки приложения:

```
POST /blitz/admin/api/v3/app/test-app HTTP/1.1
Authorization: Bearer cNw…Nz
Content-Type: application/json
If-Match: 98_1658857264000

{
    "tags": [
        "default",
        "2F"
    ]
}
```

Если изменение успешно, то сервер вернет HTTP 200, актуальные значения настроек приложения и новый `etag`.

Пример ответа:

```
HTTP/2 200
…
content-type: application/json
etag: 99_1658857631000

{
    "name": "",
    "tags": [
        "default",
        "2F"
    ],
    "domain": "test.app1.ru",
    "id": "app1",
    "simple": {
        "formSelector": "select",
        "postLogoutUrl": "http://localhost",
        "ssl": true,
        "loginSelector": "select",
        "js": "dyMw==",
        "logoutUrl": "https://localhost"
    },
    "disabled": false
}
```

Пример запроса удаления меток приложения:

```
POST /blitz/admin/api/v3/app/test-app HTTP/1.1
Authorization: Bearer cNw…Nz
Content-Type: application/json
If-Match: 98_1658857264000

{
    "tags": null
}
```

Пример ответа:

```
HTTP/2 200
…
content-type: application/json
etag: 99_1658857631000

{
    "name": "",
    "domain": "test.app1.ru",
    "id": "app1",
    "simple": {
        "formSelector": "select",
        "postLogoutUrl": "http://localhost",
        "ssl": true,
        "loginSelector": "select",
        "js": "dyMw==",
        "logoutUrl": "https://localhost"
    },
    "disabled": false
}
```

Если при редактировании приложения будет обнаружено, что данные в конфигурационном файле на сервере были изменены между получением etag и вызовом редактирования, то сервер вернет ответ с кодом `HTTP 412 Precondition Failed` и телом ошибки:

```
{
   "type":"process_error",
   "error":"cas_mismatch",
   "desc":"cas_mismatch"
}
```

Если при редактировании приложения возникла ошибка, что переданы неправильные данные, то сервер вернет ответ с кодом `HTTP 400 Bad Request` с описанием ошибок.

Пример ответа с ошибкой:

```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "json.error.mandatory.field",
            "desc": "json.error.expected.array",
            "pos": "oauth.redirectUriPrefixes"
        },
        …
    ]
}
```

Удаление приложения

Для удаления приложения необходимо выполнить запрос методом DELETE по адресу `https://login.company.com/blitz/admin/api/v3/app/{appId}`.

Необходимые разрешения: `blitz_api_sys_app_chg`.

В запрос должен быть добавлен заголовок `If-Match`, содержащий последнее полученное от сервера значение `etag`.

Пример запроса:

```
DELETE /blitz/admin/api/v3/app/test-app HTTP/1.1
Authorization: Bearer cNw…Nz
If-Match: 99_1658857631000
```

Если приложение успешно удалено, то сервер вернет HTTP 204.

Если при удалении приложения будет обнаружено, что данные в конфигурационном файле на сервере были изменены между получением etag и вызовом удаления, то сервер вернет ответ с кодом `HTTP 412 Precondition Failed` и телом ошибки:

```
{
   "type":"process_error",
   "error":"cas_mismatch",
   "desc":"cas_mismatch"
}
```

3.5.4 Вызов стороннего приложения регистрации пользователей

В Blitz Identity Provider можно настроить использование стороннего приложения регистрации пользователей. В этом случае Blitz Identity Provider сможет вызвать приложение регистрации пользователей со страницы входа (при переходе по ссылке *Зарегистрироваться*) или в результате первого входа пользователя через внешний поставщик идентификации. При этом доступны следующие возможности:

- В случае если регистрация запущена в результате первого входа через внешний поставщик идентификации, то Blitz Identity Provider передаст приложению регистрации полученные из внешнего поставщика идентификации атрибуты. Приложение сможет их использовать для предзаполнения формы регистрации.
- Если пользователь успешно пройдет регистрацию, то он сможет продолжить процесс входа. Например, можно обеспечить автоматический вход зарегистрированного пользователя в приложение аналогично тому, как это происходит при использовании встроенного в Blitz Identity Provider приложения регистрации.

Для подключения к Blitz Identity Provider стороннего приложения регистрации необходимо на стороне веб-приложения регистрации поддержать сервисы в соответствии с описанными в последующих разделах требованиями.

Сервис инициирования регистрации

Стороннее приложение регистрации должно предоставить HTTP POST сервис инициирования регистрации.

> **Примечание:** Адрес сервиса задается в настройках Blitz Identity Provider (см. *Администрирование* (страница 11)).

Сервис должен принимать следующие параметры (в виде JSON):

- `id` – идентификатор заявки на регистрацию;
- `entryPoint` – сведения о точке входа. Возможны следующие значения:
  - `SOCIAL` – регистрация вызвана вследствие входа нового пользователя через внешний поставщик идентификации;
  - `WEB` – пользователь самостоятельно инициировал регистрацию (выбрал «Зарегистрироваться» на странице входа).
- `appId` – идентификатор приложения, в которое изначально хотел войти пользователь, в результате чего запустился процесс регистрации;
- `expires` – время окончания действия заявки на регистрацию. Указывается в Unix time, в секундах;
- `source` – источник сведений о пользователе (в случае получения сведений из внешнего поставщика входа). Содержит идентификатор внешнего поставщика входа;
- перечень атрибутов, полученных из внешнего поставщика идентификации. Передаются атрибуты из настроек связывания учетных записей соответствующего внешнего поставщика идентификации.
- `hints` – подсказки, переданные в вызов формы входа. Например, тут может быть передан логин пользователя, в случае если пользователь инициировал самостоятельную регистрацию с формы входа, которая в свою очередь была открыта с параметром `login_hint`;
- `lang` – текущий язык интерфейса пользователя на странице входа.

Пример запроса (при вызове в режиме входа через ЕСИА):

```
POST /reg/url HTTP/1.1
Content-Type: application/json
```

(continues on next page)

(продолжение с предыдущей страницы)


```
{
   "id":"6DXDHyyiZ2hByUN-sCRUEdvAoQun7WwQ",
   "entryPoint":"SOCIAL",
   "appId":"portal",
   "expires":1608129702,
   "source":"esia:esia_1",
   "hints": {},
   "attrs":[
      {
         "esia_family_name":"Петров",
         "esia_given_name":"Иван",
         "esia_middle_name":"Сергеевич",
         "esia_passport": "{\"issueDate\":\"01.01.2016\",\"stateFacts\":[\
↪"EntityRoot\"],\"eTag\":\"452E4EEA3A9FBCD244766D6549B8E7E616478BD2\",\"vrfStu\":\
↪"VERIFIED\",\"type\":\"RF_PASSPORT\",\"issueId\":\"111001\",\"number\":\"123456\
↪",\"series\":\"4567\",\"issuedBy\":\"РУВД г.Москвы\",\"id\":38226}",
         "esia_trusted":true,
         "esia_id":"1000334562",
         "esia_gender": "M",
         "esia_birthdate": "01.01.1999",
         "esia_birthplace": "Москва",
         "esia_email": "johndoe@company.ru",
         "esia_snils": "123-456-789 12",
         "esia_inn": "123456789012",
         "esia_phone_number": "+7(999)1234567",
         "esia_liv_address": {\"stateFacts\":[\"Identifiable\"],\"id\":24243131,\
↪"type\":\"PRG\",\"addressStr\":\"г Москва, ул Онежская\",\"fiasCode\":\"06690b31-
↪d4ae-463d-ad12-cf3963e0d7ed\",\"flat\":\"56\",\"countryId\":\"RUS\",\"house\":\
↪"16\",\"zipCode\":\"125414\",\"street\":\"Онежская\",\"region\":\"Москва\",\
↪"vrfDdt\":\"0,10,0\",\"eTag\":\"0C7C02CA3BC3623B2628A7603DA342792D5CE491\"}",
         "esia_reg_address": {\"stateFacts\":[\"Identifiable\"],\"id\":24343142,\
↪"type\":\"PRG\",\"addressStr\":\"г Москва, ул Онежская\",\"fiasCode\":\"06690b31-
↪d4ae-463d-ad12-cf3963e0d7ed\",\"flat\":\"56\",\"countryId\":\"RUS\",\"house\":\
↪"16\",\"zipCode\":\"125414\",\"street\":\"Онежская\",\"region\":\"Москва\",\
↪"vrfDdt\":\"0,10,0\",\"eTag\":\"0C7C02CA3BC3623B2628A7603DA342792D5CE591\"}"
      }
   ],
   "lang": "ru"
}
```


Пример запроса (при нажатии пользователем «Зарегистрироваться» на странице входа):

```
POST /reg/url HTTP/1.1
Content-Type: application/json

{
   "id":"6DXDHyyiZ2hByUN-sCRUEdvAoQun7WwQ",
   "entryPoint":"WEB",
   "appId":"portal",
   "expires":1608129702,
   "hints": {},
   "attrs": {},
   "lang": "ru"
}
```

В ответ сервис инициирования регистрации должен вернуть либо HTTP-ответ для выполнения в браузере пользователя (например, код HTML-страницы или инициировать перенаправление пользователя в браузере на страницу регистрации), либо сообщение об ошибке.

Пример ответа:

```
HTTP/1.1 302 Found
Location: https://www.company.ru/register/
```

В результате пользователь будет перенаправлен из Blitz Identity Provider в стороннее приложение регистрации.

Сервис завершения регистрации

Когда пользователь в стороннем приложении регистрации ввел все данные, необходимые для регистрации учетной записи, стороннее приложение регистрации должно вызвать в Blitz Identity Provider сервис завершения регистрации учетной записи пользователя. Сервис вызывается методом POST по адресу `https://login.company.com/blitz/reg/api/v1/users/{id}`, где в качестве `id` в URL сервиса передается идентификатор заявки на регистрацию, ранее полученный от Blitz Identity Provider.

В запрос должен быть добавлен следующий заголовок, где `secret` – это присвоенные приложению при регистрации в Blitz Identity Provider `client_id:rest_secret` в формате Base64:

```
Authorization: Basic <secret>
```

Внимание: Список атрибутов приведен в качестве образца. Содержание списка необходимо скорректировать в зависимости от конкретных настроек, сделанных при внедрении Blitz Identity Provider. См. *Администрирование* (страница 11).

Тело запроса должно содержать атрибуты регистрируемой учетной записи:

- `first_name` – фамилия;
- `name` – имя;
- `middle_name` – отчество;
- `phone_number` – номер мобильного телефона в виде составного объекта с атрибутами:
  - `value` – номер телефона в формате 7XXXXXXXXXX;
  - `verified` – признак, что телефон подтвержден – `true` или `false`;
- `email` – адрес электронной почты в виде составного объекта с атрибутами:
  - `value` – адрес электронной почты;
  - `verified` – признак, что адрес подтвержден – `true` или `false`;
- `password` – пароль для создаваемой учетной записи пользователя (должен соответствовать настроенной парольной политике).

Пример запроса (регистрация с подтвержденными email и телефоном):

```
POST /blitz/reg/api/v1/users/6DXDHyyiZ2hByUN-sCRUEdvAoQun7WwQ HTTP/1.1
Authorization: Basic YXBwX2lkOmFwcF9zZWNyZXQ=
Content-Type: application/json

{
    "first_name":"Иванов",
    "name":"Иван",
    "middle_name":"Иванович",
    "phone_number": {
        "value": "79991234567",
        "verified": true
    },
    "email": {
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        "value":"mail@example.com",
        "verified": true
    },
    "password":"QWErty$123"
}
```

В ответ Blitz Identity Provider в случае успешного завершения регистрации вернет JSON со следующими данными:

- `subject` – идентификатор зарегистрированного пользователя;
- `origin` – ссылку, на которую необходимо направить браузер пользователя;
- `cookies` – куки, которые нужно установить при перенаправлении браузера пользователя на общем с Blitz Identity Provider домене;
- `instanceId`, `instructions` – прочие технологические сведения, которые нужно проигнорировать.

Пример ответа:

```
{
   "instanceId": "amRiY2lkcG9zdGdyZXM6YzhjMGExYzEtYzdmYS00ZDg3LWFiYmMtZTNiYzg1YTk4
↪",
   "subject":"5cffd68f-2cb8-4f7a-b0f3-9fa69a1fbbcd",
   "context":"6DXDHyyiZ2hByUN-sCRUEdvAoQun7WwQ",
   "cookies": [{
      "name": "css",
      "value": "TSQA-AruOjUNphGZ984eLgzT_ROebNiBsjyjEg4n-nL-PdsiXqq"
    }],
    "origin": "/blitz/profile?",
    "instructions": []
}
```

После перенаправления сторонним приложением регистрации браузера пользователя по ссылке, указанной в `origin`, и с указанными `cookies` Blitz Identity Provider создаст сессию и обеспечит вход пользователя в приложение, для входа в которое пользователь осуществил регистрацию учетной записи.

3.5.5 API аутентификации

Стандартно при необходимости провести идентификацию и аутентификацию пользователя веб-сайт или мобильное приложение взаимодействует с Blitz Identity Provider по любому из доступных протоколов (см. *Выбор протокола взаимодействия* (страница 383)). При этом непосредственно аутентификацией приложение не занимается. Приложение перенаправляет пользователя в Blitz Identity Provider на страницу входа. Далее Blitz Identity Provider самостоятельно предлагает пользователю различные методы аутентификации, осуществляет взаимодействие с пользователем в процессе входа.

В некоторых случаях может быть желательно предоставить пользователю возможность пройти идентификацию и аутентификацию без перенаправления на страницу входа Blitz Identity Provider. Такие возможности ограничены (не все методы входа и подтверждения входа доступны без перенаправления), требуют большого объема доработок на стороне приложения (так как в приложении необходимо поддержать обработку различных сценариев, связанных с аутентификацией).

Blitz Identity Provider предоставляет HTTP API, позволяющее встроить в веб-страницу приложения идентификацию и аутентификацию пользователей без перенаправления пользователя на отдельную страницу входа. Данное HTTP API создано для веб-приложений. При использовании API обеспечивается Web Single Sign-On, а именно при последующем входе в той же веб-сессии пользователя в другое подключенное к Blitz Identity Provider приложение, у него не будет повторно запрашиваться вход.

Настройки для использования API

Приложение должно быть зарегистрировано в Blitz Identity Provider. Приложению в Blitz Identity Provider должны быть присвоены `client_id` и `client_secret`, и в Blitz Identity Provider должны быть зарегистрированы URL возврата приложения.

Взаимодействие страницы приложения и Blitz Identity Provider основано на выполнении серии AJAX-взаимодействий. Для возможности такого взаимодействия на веб-сервере приложения и на веб-сервере Blitz Identity Provider должны быть сделаны следующие настройки CORS (`Cross-origin resource sharing`):

1. На сервере Blitz Identity Provider для обработчика `/blitz/oauth/ae` нужно настроить CORS-разрешение, добавив следующие HTTP Headers (нужно указать `origin` для ПРОД-сайта и необходимые `origin` для нужных тестовых сред):

   ```
   "Access-Control-Allow-Origin" -> "https://{app-domain}",
   "Access-Control-Allow-Credentials" -> "true"
   ```

   В этом заголовке `{app-domain}` – это домен приложения.

2. На сервере портала для callback-обработчика (см. *Схема взаимодействия* (страница 527)) ответа от Blitz Identity Provider нужно настроить следующее CORS-разрешение (разрешение на `null`, так как после редиректа браузер сбрасывает `origin`):

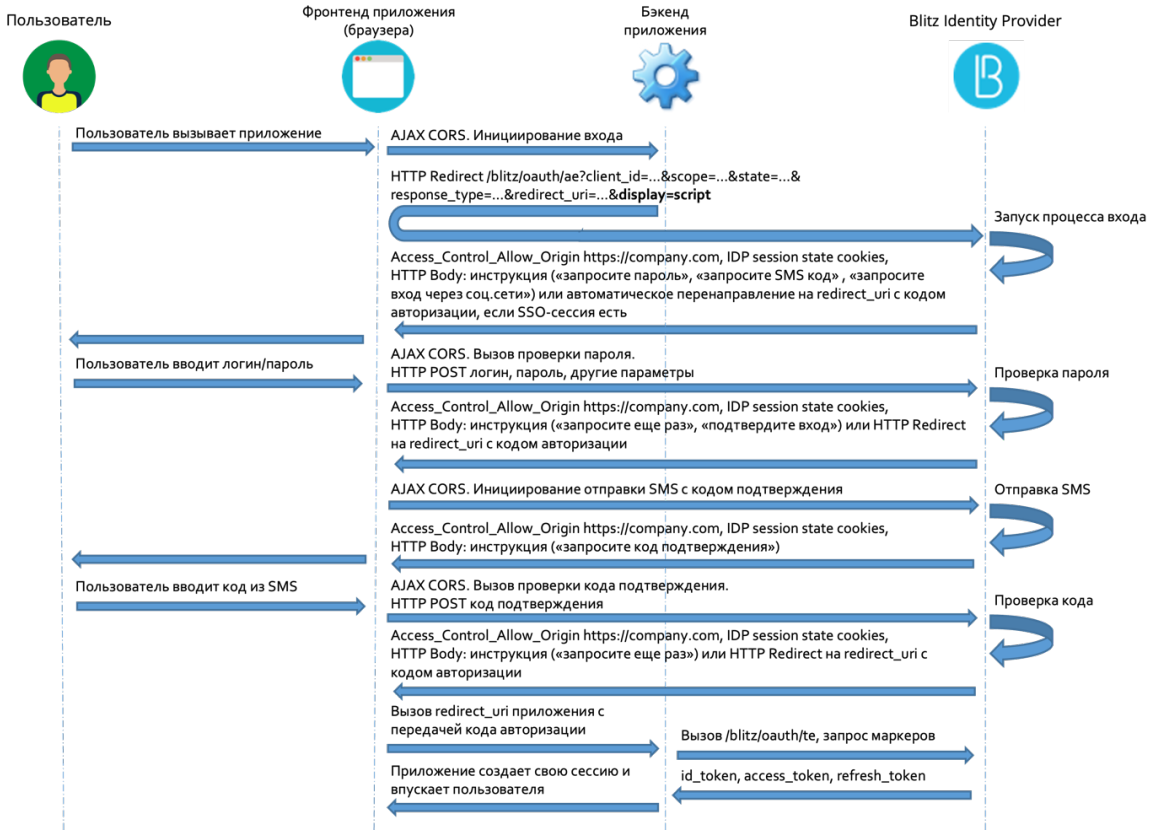
   ```
   "Access-Control-Allow-Origin" -> null,
   "Access-Control-Allow-Credentials" -> "true"
   ```

Схема взаимодействия

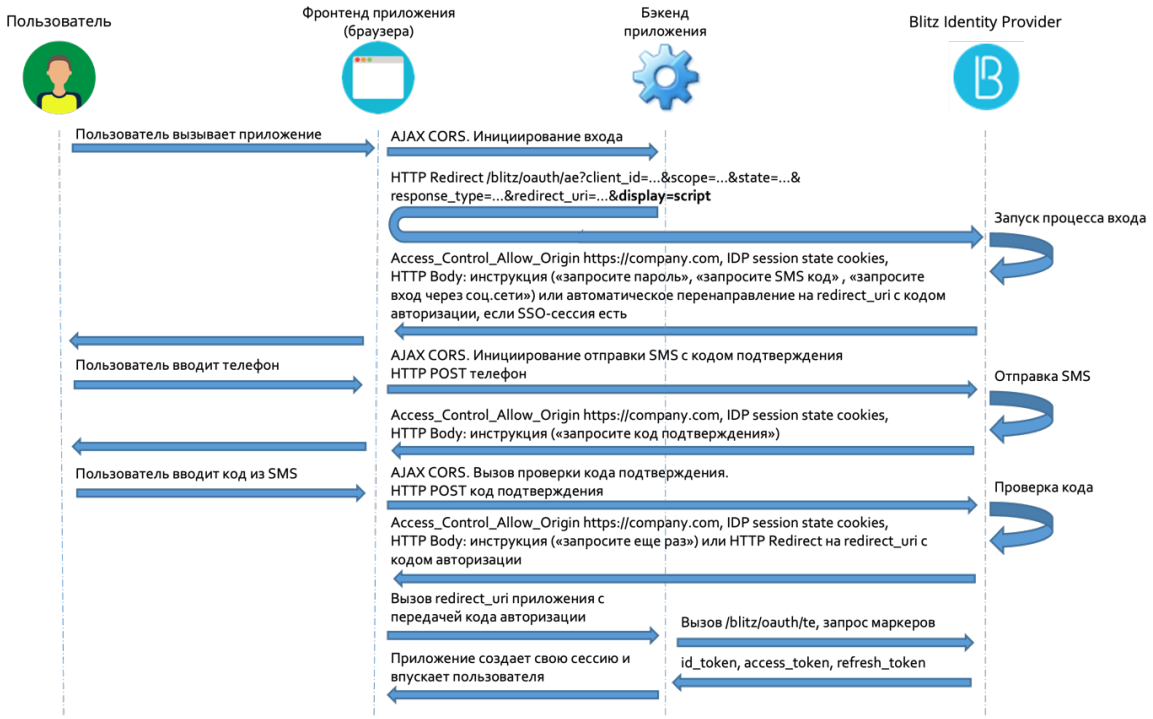
HTTP API аутентификации позволяет:

- Проверить наличие SSO-сессии. В случае отсутствия SSO-сессии получить список доступных пользователю методов аутентификации.
- Провести идентификацию и аутентификацию с использованием логина и пароля.
- Провести идентификацию и аутентификацию с использованием логина (телефона) и кода подтверждения, отправляемого по SMS.
- Провести идентификацию и аутентификацию по QR-коду;
- Провести подтверждение входа с использованием кода подтверждения, отправляемого по SMS.

На рисунке ниже приведена схема взаимодействия при входе по логину и паролю с последующим подтверждением входа с использованием кода подтверждения, отправляемого по SMS.



На следующем рисунке приведена схема взаимодействия при входе по телефону и коду подтверждения, отправляемому по SMS.



Веб-приложение взаимодействует с Blitz Identity Provider, выполняя серию из AJAX-запросов.

Примечание: Запросы должны делаться обязательно с сохранением и передачей cookie – необходимо использовать `withCredentials: true`

В последующих разделах приводятся описания вызываемых запросов, возможных ответов и рекомендаций по их обработке. Примеры запросов и ответов приводятся в виде вызовов cURL.

Запуск процесса входа

Чтобы запустить процесс входа, приложение должно направить в AJAX к Blitz Identity Provider запрос HTTP GET (обязательно с `withCredentials: true`) на обычный обработчик Authorization Endpoint (`/blitz/oauth/ae`, см. *Получение кода авторизации* (страница 390)), добавив к запросу специальный параметр `display=script`.

Пример запроса:

```
curl -v -b cookies.txt -c cookies.txt \
--request GET 'https://login.company.com/blitz/oauth/ae?response_type=code&client_
↪id=ais&scope=openid&state=…&display=script&redirect_uri=https%3A%2F%2Fapp.
↪company.com%2Fre'
```

Если SSO-сессия уже существует, то Blitz Identity Provider автоматически перенаправит пользователя на адрес обработчика `redirect_uri`, добавив к запросу код авторизации и параметр `state`. Используя полученный код авторизации приложение продолжит стандартное взаимодействие по OpenID Connect для получения маркеров безопасности и данных учетной записи.

Пример ответа с перенаправлением, если сессия SSO-сессия уже существует:

```
…
< HTTP/2 302
…
< location: https://…?code=…&state=…
…
```

Пример ответа, если требуется аутентификация:

```
{
    "inquire":"choose_one",
    "items":[
        {
            "inquire":"login_with_password"
        },
        {
            "inquire":"request_auth_with_fed_point",
            "fp":"esia:esia_1"
        },
        …
        {
            "inquire":"request_auth_with_fed_point",
            "fp":"yandex:yandex_1"
        },
        {
            "inquire":"login_to_send_sms"
        },
        {
            "inquire":"show_qr_code",
            "link":"https://…?code=dde087f0-8f4a-478e-886b-5354b0283362",
            "expires":1660905165,
            "logo":"https:/…"
        }
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    ]
}
```

Если требуется аутентификация, то Blitz Identity Provider возвращает приложению одну из возможных инструкций:

- `login_with_password` – войти по логину и паролю;
- `request_auth_with_fed_point` – войти с помощью внешнего поставщика идентификации (социальной сети);
- `login_to_send_sms` – войти с помощью логина и кода подтверждения, отправленного по SMS;
- `show_qr_code` – отобразить QR-код, позволяющий осуществить вход.

Если какие-то из методов аутентификации не сконфигурированы в Blitz Identity Provider или являются недоступными для входа в запрашивающее приложение (например, в результате настроек «процедуры входа» для соответствующего приложения), то и инструкции по ним будут отсутствовать в ответе сервиса.

В зависимости от включенных в Blitz Identity Provider режимов защиты инструкция `login_with_password` может содержать дополнительные параметры:

- Если в Blitz Identity Provider настроен режим необходимости использования CAPTCHA при входе, то в инструкции будет параметр `captchaId`, который необходимо использовать приложению для теста CAPTCHA:

```
{
    "inquire": "choose_one",
    "items": [
        {
            "inquire": "login_with_password",
            "captchaId": "9cf48a75-6be1-4008-b34e-8906220c472f"
        },
        …
    ]
}
```

- Если в Blitz Identity Provider настроен режим защиты от подбора пароля, требующий решения от приложения длительной вычислительной задачи (Proof of Work), то в инструкции будет параметр `proofOfWork`:

```
{
    "inquire": "choose_one",
    "items": [
        {
            "inquire": "login_with_password",
            "captchaId": "9cf48a75-6be1-4008-b34e-8906220c472f",
            "proofOfWork": "1:15:220313184752:abe…539::Ekf…w==:"
        }
    ]
}
```

- В случае получения параметра `proofOfWork` рекомендуется асинхронно сразу запустить алгоритм нахождения решения, не дожидаясь, пока пользователь выберет режим входа по логину и паролю и введет данные. Это позволит скрыть от пользователя время задержки на решение задачи (может составлять несколько секунд в зависимости от сложности задачи). В настоящий момент используется алгоритм Hashcash[120].

Важно: Необходимо дополнить параметр proofOfWork таким значением, чтобы вычисленный от

[120] http://www.hashcash.org

него по алгоритму SHA-1 хэш содержал в начале столько нулевых бит, сколько задано условием задачи (число после первого символа `:` в параметре `proofOfWork`).

Например, решением для `1:15:yyyy03Su212003:BlitzIdp::McMybZIhxKXu57jd:0` будет строка `1:15:yyyy03Su212003:BlitzIdp::McMybZIhxKXu57jd:3/g`

В зависимости от выбранного способа аутентификации приложение вызывает в Blitz Identity Provider вход одним из следующих способов:

- *Вход по логину и паролю* (страница 531).
- *Вход по телефону и коду подтверждения в SMS* (страница 536).
- *Вход по QR-коду* (страница 540).
- Вход через внешний поставщик идентификации – такой способ входа возможен только через браузер с перенаправлением пользователя на страницу входа внешнего поставщика идентификации. Нужно повторить вызов `Authorization Endpoint` (см. *Получение кода авторизации* (страница 390)), использовать в вызове необходимое значение параметра `bip_action_hint`, соответствующее выбранному пользователем внешнему поставщику входа (например, `bip_action_hint=externalIdps:esia:esia_1`).

Пример запроса:

```
https://login.company.com/blitz/oauth/ae?response_type=code&client_id=portal.ru&
↪scope=openid+profile&redirect_uri=https://apitest.company.com/success&
↪state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f& bip_action_hint=used_
↪externalIdps:esia:esia_1
```

Завершение процесса входа в этом случае будет происходить стандартным образом в соответствии с OpenID Connect – Blitz Identity Provider вернет код авторизации на `redirect_uri` обработчик приложения.

Вход по логину и паролю

Если в Blitz Identity Provider настроено использование CAPTCHA, то до вызова проверки логина и пароля приложение должно выполнить вызовы по получению и проверке CAPTCHA. Запросы на проверку должны формироваться через специализированные proxy-сервисы Blitz Identity Provider, а не напрямую к сервисам CAPTCHA.

При использовании reCAPTCHA v3 необходимо выполнить инициализацию reCAPTCHA v3 согласно документации[121].

- Загрузить на странице приложения скрипт, используя такой же reCAPTCHA v3 `sitekey` как зарегистрирован в Blitz Identity Provider:

```
<script src="https://www.google.com/recaptcha/api.js?render=reCAPTCHA_site_key"></
↪script>
```

- Вызвать grecaptcha.execute на нажатие кнопки входа:

```
<script>
    function onClick(e) {
        e.preventDefault();
        grecaptcha.ready(function() {
            grecaptcha.execute('reCAPTCHA_site_key', {action: 'submit'}).
↪then(function(token) {
                // Add your logic to submit to your backend server here.
            });
```

(continues on next page)

[121] https://developers.google.com/recaptcha/docs/v3#programmatically_invoke_the_challenge

(продолжение с предыдущей страницы)

```
        });
    }
</script>
```

Сразу после вызова со страницы входа сервисов reCAPTCHA необходимо вызвать с сервера приложений операцию проверки (verify). Вызов должен быть произведен не напрямую на сервера Google, а через специальный proxy-сервис в Blitz Identity Provider.

Пример запроса на проверки (операция `verify`):

```
POST /blitz/login/captcha/verify
Content-Type: 'text/json'
{
    "ctx": {
         // captchaId
        "id": "9cf48a75-6be1-4008-b34e-8906220c472f",
        "method": "password"
    },
    "params": {
        // token для проверки капчи, полученный при регистрации в Google
        "response": "03…sA"
    }
}


Ответ ``HTTP 200 OK``:
```

```
{
    "action": "submit",
    "challenge_ts": "2021-03-16T11:18:41Z",
    "success": true,
    "hostname": "company.com",
    "score": 0.9
}
```

Также если в Blitz Identity Provider включена защита Proof of Work, то нужно вычислить значение параметра `proofOfWork` (см. *Запуск процесса входа* (страница 529)).

Для проверки логина и пароля приложение должно направить в AJAX к Blitz Identity Provider запрос HTTP POST (обязательно с `withCredentials: true`) на URL `https://login.company.com/blitz/login/methods/headless/password` с `Content-Type x-www-form-urlencoded` и Body, содержащим параметры `login` и `password`, а также вычисленный `proofOfWork` (если этот параметр был получен от Blitz Identity Provider при запуске процесса входа).

Пример запроса:

```
curl -v -b cookies.txt -c cookies.txt \
--request POST 'https://login.company.com/blitz/login/methods/headless/password' \
--header "Content-Type: application/x-www-form-urlencoded" \
--data 'login=логин&password=пароль&proofOfWork=решение'
```

Blitz Identity Provider при получении запроса выполняет необходимые проверки безопасности (пройдена ли CAPTCHA, решен ли `ProofOfWork`, не заблокирована ли учетная запись). Если проверки безопасности пройдены, то Blitz Identity Provider проверяет переданные логин и пароль.

Если проверки логина и пароля успешные и если пройденной аутентификации достаточно, то Blitz Identity Provider автоматически перенаправит пользователя на адрес обработчика `redirect_uri`, добавив к запросу код авторизации и параметр `state`. Используя полученный код авторизации приложение продолжит стандартное взаимодействие по OpenID Connect для получения маркеров безопасности и данных учетной записи.

Пример ответа с перенаправлением, если сессия SSO-сессия уже существует:

```
…
< HTTP/2 302
…
< location: https://…?code=…&state=…
…
```

Если какие-либо проверки завершились ошибкой или если необходимы дальнейшие действия от пользователя, то Blitz Identity Provider возвращает одну из инструкций.

Пример ответа в случае ошибки проверки логина и пароля:

```
{
    "inquire": "login_with_password",
    "errors": [
        {
            "code": "invalid_credentials",
            "params": {}
        }
    ]
}
```

При получении такого ответа приложение может отобразить текст ошибки и предложить пользователю ввести еще логин и пароль, после чего можно повторить проверку логина и пароля.

Если пользователь ввел пароль, который ранее был в учетной записи, или если учетная запись заблокирована, то ошибка будет иметь вид:

```
{
    "inquire": "login_with_password",
    "captchaId": "9cf48a75-6be1-4008-b34e-8906220c472f",
    "proofOfWork": "1:15:220313184752:abe…539::Ekf…w==:",
    "errors": [
        {
            "code": "invalid_credentials",
            "params": {
                "_cause": "used_old_password"
            }
        }
    ]
}
```

Пример получения ошибки, что не прошла проверка CAPTCHA:

```
{
    "inquire": "login_with_password",
    "captchaId": "9cf48a75-6be1-4008-b34e-8906220c472f",
    "errors": [
        {
            "code": "invalid_captcha",
            "params": {}
        }
    ]
}
```

Пример ошибки, что не прошла проверка решения Proof of Work:

```
{
    "inquire": "handle_error",
    "errors": [
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        {
            "code": "doesNotMatch",
            "params": {}
        }
    ]
}
```

Если в Blitz Identity Provider включена специальная защита на задержку проверки логина и пароля, то при проверке логина и пароля можно получить от Blitz Identity Provider следующую инструкцию, что нужен повторный вызов проверки пароля спустя определенное число секунд:

```
{
   "inquire": "delayed_login_with_password",
   "delayedFor": 5
}
```

Повторный вызов должен быть сделан, когда пройдет требуемое время. В повторный вызов необходимо передать параметр `isDelayed=true`.

Пример повторного вызова проверки пароля в ответ на инструкцию `delayed_login_with_password`:

```
curl -v -b cookies.txt -c cookies.txt \
--request POST 'https://login.company.com/blitz/login/methods/headless/password' \
--header "Content-Type: application/x-www-form-urlencoded" \
--data 'login=логин&password=пароль&proofOfWork=решение&isDelayed=true'
```

Если в Blitz Identity Provider включена специальная защита от перебора пароля, то Blitz Identity Provider при проверке пароля по данной учетной записи может запросить дополнительно проверку CAPTCHA. Различаются две возможные ситуации:

- Пользователь передал неправильный пароль, после чего включилась защита, и CAPTCHA нужна для очередной попытки аутентификации.
- Защита от подбора пароля для учетной записи включалась ранее. Текущий переданный пароль не проверялся, так как не проводился тест CAPTCHA.

В первом случае нужно сообщить пользователю, что логин и пароль неправильный, и для новой попытки дополнительно к вводу пароля запросить пройти тест CAPTCHA.

Во втором случае нужно попросить пользователя пройти тест CAPTCHA, после чего направить на проверку ранее введенные логин и пароль.

Пример ответа для первого случая, что пароль неправильный и нужен тест CAPTCHA:

```
{
    "inquire":"login_with_password",
    "captchaId":"1c9e4047-c8c4-47ad-a447-cc1809bd3e6c",
    "errors":[
        {
            "code":"invalid_credentials",
            "params":{}
        }
    ]
}
```

Пример ответа для второго случая, что пароль не проверялся и нужен тест CAPTCHA:

```
{
    "inquire":"login_with_password",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "captchaId":"2f818f5d-3a89-428d-b424-cde38c19051e",
    "errors":[
        {
            "code":"bypass_captcha",
            "params":{}
        }
    ]
}
```

Пример ошибки, если учетная временно заблокирована:

```
{
    "inquire":"login_with_password",
    "errors": [
        {
            "code":"pswd_method_temp_locked",
            "params":{"0":"2"}
        }
    ]
}
```

Пример ошибки, если учетная заблокирована по причине длительной неактивности:

```
{
    "inquire": "handle_error",
    "errors": [
        {
            "code": "inactivity_lock",
            "params": {}
        }
    ]
}
```

Если пароль учетной записи не соответствует парольной политике, то может возникнуть необходимость сменить пароль при входе. В этом случае Blitz Identity Provider вернет инструкцию, что необходимо перенаправить пользователя на страницу с указанным адресом:

```
{
    "inquire":"go_to_web",
    "redirect_uri":
        "https://…/blitz/login/methods/password/change?f=false&c=password_policy_
↪violated"
}
```

Если логин и пароль успешны, но дополнительно требуется подтвердить вход, то вернется инструкция с возможными способами подтверждения:

```
{
    "inquire":"choose_one",
    "items": [
        {
            "inquire":"ask_to_send_sms"
        },
        {
            "inquire":"go_to_web",
            "redirect_uri":"https://login.company.com/blitz/login/methods2/sms"
        }
    ]
}
```

Можно или перенаправить пользователя на веб-страницу, чтобы он продолжил подтверждение входа на веб-странице Blitz Identity Provider, или продолжить использовать HTTP API для *подтверждения входа по коду из SMS* (страница 543).

Если в процедуре входа, установленной для приложения, настроен вызов дополнительного экрана после входа (например, см. *Вызов вспомогательного приложения в момент входа* (страница 512)). Вызов вспомогательного приложения в момент входа), то Blitz Identity Provider переадресует пользователя на этот экран.

Вход по телефону и коду подтверждения

Вход по телефону и коду подтверждения состоит из следующих шагов:

- Отправка пользователю кода подтверждения по SMS.
- Проверка введенного пользователем кода подтверждения.

Для отправки пользователю кода подтверждения по SMS приложение должно направить в AJAX к Blitz Identity Provider запрос HTTP POST (обязательно с `withCredentials: true`) на URL `https://login.company.com/blitz/login/methods/headless/sms/bind` с `Content-Type x-www-form-urlencoded` и Body, содержащим `login` пользователя. В качестве `login` рекомендуется передавать номер телефона, введенный пользователем.

Пример запроса:

```
curl -v -b cookies.txt -c cookies.txt \
--request POST 'https://login.company.com/blitz/login/methods/headless/sms/bind' \
--header "Content-Type: application/x-www-form-urlencoded" \
--data 'login=логин'
```

Если учетная запись с переданным логином не найдена, то Blitz Identity Provider вернет ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"no_subject_found",
            "params":{}
        }
    ]
}
```

Если учетная запись найдена, но по ней ранее был зафиксирован перебор кодов подтверждения, то Blitz Identity Provider вернет ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"method_temp_locked",
            "params":{}
        }
    ]
}
```

Если учетная запись найдена и для нее возможен вход данным способом, то Blitz Identity Provider отправит пользователю SMS с кодом подтверждения и вернет ответ:

```
{
    "inquire":"enter_sms_code",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "contact":"+79991234567",
    "ttl":300,
    "remain_attempts":3
}
```

В полученном ответе указано, сколько секунд у пользователя остается для отправки кода на проверку (ttl), сколько попыток ввести код у него есть (remain_attempts), на какой номер телефона ему был отправлен код (`contact`).

Для проверки введенного пользователем кода подтверждения приложение должно направить в AJAX к Blitz Identity Provider запрос HTTP POST (обязательно с `withCredentials: true`) на URL `https://login.company.com/blitz/login/methods/headless/sms/bind` с `Content-Type x-www-form-urlencoded` и Body, содержащим `sms-code` с кодом подтверждения.

Пример запроса:

```
curl -v -b cookies.txt -c cookies.txt \
--request POST 'https://login.company.com/blitz/login/methods/headless/sms/bind' \
--header "Content-Type: application/x-www-form-urlencoded" \
--data 'sms-code=123456'
```

Если код неправильный, то Blitz Identity Provider вернет ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"invalid_otp",
            "params":{}
        }
    ],
    "contact":"+79991234567",
    "remain_attempts":2,
    "ttl":276
}
```

Если количество попыток проверки кода закончилось, то Blitz Identity Provider вернет ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"no_attempts",
            "params":{}
        }
    ]
}
```

Если срок действия кода истек, то Blitz Identity Provider вернет ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"expired",
            "params":{}
        }
    ]
}
```

В случае этой ошибки можно запросить отправку нового кода подтверждения. Для этого приложение должно вызвать Blitz Identity Provider следующим образом:

```
curl -v -b cookies.txt -c cookies.txt \
--request POST 'https://login.company.com/blitz/login/methods/headless/sms/bind' \
--header "Content-Type: application/x-www-form-urlencoded" \
--data 'sms-send=sms'
```

Если запросить переотправку кода до истечения срока действия предыдущего, то вернется ошибка:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"code_not_expired",
            "params":{}
        }
    ]
}
```

Если общее количество попыток входа по коду подтверждения из SMS превышено, то Blitz Identity Provider осуществляет временное блокирование входа для учетной записи по коду подтверждения. В этом случае при очередной попытке ввода неправильного кода подтверждения Blitz Identity Provider может вернуть ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"method_temp_locked",
            "params":{}
        }
    ]
}
```

Если введенный код подтверждения правильный, и этого достаточно для завершения входа, то Blitz Identity Provider автоматически перенаправит пользователя на адрес обработчика `redirect_uri`, добавив к запросу код авторизации и параметр `state`. Используя полученный код авторизации приложение продолжит стандартное взаимодействие по OpenID Connect для получения маркеров безопасности и данных учетной записи.

Пример ответа в случае успешного входа:

```
…
< HTTP/2 302
…
< location: https://…?code=…&state=…
…
```

Если проверка кода подтверждения успешна, но дополнительно требуется подтвердить вход, то вернется инструкция с возможными способами подтверждения:

```
{
    "inquire":"choose_one",
    "items":[
        {
            "inquire":"go_to_web",
            "redirect_uri":"https://login.company.com/blitz/login/methods2/email"
        }
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    ]
}
```

Первичный вход по email

Первичный вход с помощью электронной почты состоит из следующих шагов:

- Отправка пользователю кода подтверждения по электронной почте.
- Проверка введенного пользователем кода подтверждения.

Для отправки пользователю кода подтверждения по email приложение должно направить в AJAX к Blitz Identity Provider запрос HTTP POST (обязательно с `withCredentials: true`) на URL `https://login.company.com/blitz/login/methods/headless/email/bind` с `Content-Type x-www-form-urlencoded` и Body, содержащим `login` пользователя. В качестве `login` нужно передавать адрес электронной почты, введенный пользователем.

Пример запроса:

```
curl --location --request POST 'https://login.company.com/blitz/login/methods/
↪headless/email/bind' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'login=<email>'
```

Пример ответа:

```
{
    "inquire": "enter_email_code",
    "contact": "user@gmail.com",
    "remain_attempts": 3,
    "ttl": 300
}
```

Проверка кода:

```
curl --location --request POST 'https://login.company.com/blitz/login/methods/
↪headless/email/bind' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'email-code=746234'
```

Варианты ответов, если проверка прошла неуспешно:

```
{
    "errors": [
        {
            "code": "invalid_otp",
            "params": {}
        }
    ],
    "contact": "user@gmail.com",
    "inquire": "handle_error",
    "remain_attempts": 2,
    "ttl": 257
}
```

```
{
    "inquire": "handle_error",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "errors": [
        {
            "code": "no_attempts",
            "params": {}
        }
    ]
}
```

Повторная отправка кода:

```
curl --location --request POST 'https://login.company.com/blitz/login/methods/
↪headless/email/bind' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Cookie: blc=Hc..; bua=7cd2c312-...; cTm=1:RGVm==; cTmTgs=1:c3Nv; oauth_
↪az=0MyeV-5v_...OnIE; portal_lang=ru' \
--data-urlencode 'email-send=email'
```

Ответ на повторную отправку кода:

```
{
    "inquire": "enter_email_code",
    "contact": "user@gmail.com",
    "remain_attempts": 1,
    "ttl": 288
}
```

Вход по QR-коду

Вход по QR-коду состоит из следующих шагов:

- Отображение пользователю QR-кода на компьютере, на котором выполняется вход;
- Периодическая проверка, выполнил ли пользователь сканирование QR-кода мобильным приложением;
- Периодическая проверка, подтвердил или отклонил пользователь в мобильном приложении запрос на вход по QR-коду;
- Обновление устаревшего QR-кода.

Приложение должно отобразить пользователю QR-код, закодировав в него строку, полученную от Blitz Identity Provider. Ниже показан фрагмент инструкции для входа по QR-коду (см. *Запуск процесса входа* (страница 529)).

```
{
    "inquire":"choose_one",
    "items":[
        …
        {
            "inquire":"show_qr_code",
            "link":"https://…?code=dde087f0-8f4a-478e-886b-5354b0283362",
            "expires":1660905165,
            "logo":"https:/…"
        }
    ]
}
```

Пояснения по полученным от Blitz Identity Provider параметрам:

- `inquire` – инструкция с доступным вариантом входа, в случае входа по QR-коду имеет значение `show_qr_code`;
- `link` – ссылка, которая должна быть закодирована в QR-коде, отображаемом пользователю;
- `expires` – время (в Unix Epoch), до которого действителен QR-код. По истечении срока действия рекомендуется отобразить пользователю, что QR-код просрочен;
- `logo` – если в Blitz Identity Provider настроено отображение маленького логотипа в центре поверх QR-кода, то в указанной настройке вернется URL-адрес логотипа.

Когда приложение отобразило пользователю QR-код, необходимо дождаться, чтобы пользователь прочитал QR-код специальным мобильным приложением. Интеграция мобильного приложения для встраивания функции входа по QR-коду описано в *Вход в приложение по QR-коду* (страница 417).

Веб-приложение может периодически выполнять проверку, был ли считан мобильным приложением QR-код. Для этого необходимо выполнить в AJAX к Blitz Identity Provider запрос HTTP GET (обязательно с `withCredentials: true`) на URL `https://login.company.com/blitz/login/methods/headless/qrCode/pull`.

Пример запроса:

```
curl -v -b cookies.txt -c cookies.txt \
--request GET 'https://login.company.com/blitz/login/methods/headless/qrCode/pull'
```

Если QR-код еще не был считан, то вернется ответ:

```
{
   "command":"showQRCode"
}
```

Если QR-код считан, то вернется ответ:

```
{
   "command":"askForConfirm"
}
```

В этом случае можно обновить пользователю веб-страницу и написать на ней, что ожидается подтверждение входа в мобильном приложении.

Если QR-код просрочен, то вернется ответ:

```
{
   "command":"needRefresh",
   "cause":"qr_code_expired"
}
```

Если пользователь отклонил в мобильном приложении запрос входа по QR-коду, то вернется ответ:

```
{
   "command":"needRefresh",
   "cause":"refused_login"
}
```

В случае если QR-код просрочен или пользователь отклонил вход по QR-коду, то можно предложить пользователю получить новый QR-код. Для этого выполнить в AJAX к Blitz Identity Provider запрос HTTP POST (обязательно с `withCredentials: true`) на URL `https://login.company.com/blitz/login/methods/headless/qrCode/refresh`.

Пример запроса:

```
curl -v -b cookies.txt -c cookies.txt \
--request POST 'https://login.company.com/blitz/login/methods/headless/qrCode/
↪refresh'
```

Пример ответа:

```
{
    "link":"https://…?code=4ddf1667-d57f-4f86-b8f2-3ee53b367dfe",
    "expires":1660922807,
    "logo":"https:/…"
}
```

Если пользователь подтвердил в мобильном приложении запрос входа по QR-коду, то сервис `https://login.company.com/blitz/login/methods/headless/qrCode/pull` вернется ответ:

```
{
   "command":"needComplete"
}
```

В ответ на этот запрос для завершения входа необходимо выполнить в AJAX к Blitz Identity Provider запрос HTTP POST (обязательно с `withCredentials: true`) на URL `https://login.company.com/blitz/login/methods/headless/qrCode/complete`.

Пример запроса:

```
curl -v -b cookies.txt -c cookies.txt \
--request POST 'https://login.company.com/blitz/login/methods/headless/qrCode/
↪complete'
```

Если пройденной аутентификации достаточно для завершения входа, то Blitz Identity Provider автоматически перенаправит пользователя на адрес обработчика `redirect_uri`, добавив к запросу код авторизации и параметр `state`. Используя полученный код авторизации приложение продолжит стандартное взаимодействие по OpenID Connect для получения маркеров безопасности и данных учетной записи.

Пример ответа в случае успешного входа:

```
…
< HTTP/2 302
…
< location: https://…?code=…&state=…
…
```

Если требуется пройти дополнительно подтверждение входа, то вернется инструкция с возможными способами подтверждения:

```
{
    "inquire":"choose_one",
    "items":[
        {
            "inquire":"go_to_web",
            "redirect_uri":"https://login.company.com/blitz/login/methods2/email"
        }
    ]
}
```

Подтверждение входа по коду подтверждения

Подтверждение входа с помощью кода подтверждения по SMS состоит из следующих шагов:

- Отправка пользователю кода подтверждения по SMS.
- Проверка введенного пользователем кода подтверждения.

Для отправки пользователю кода подтверждения по SMS приложение должно направить в AJAX к Blitz Identity Provider запрос HTTP POST (обязательно с `withCredentials: true`) на URL `https://login.company.com/blitz/login/methods/headless/sms/bind` с `Content-Type x-www-form-urlencoded` без Body:

Пример запроса:

```
curl -v -b cookies.txt -c cookies.txt \
--request POST 'https://login.company.com/blitz/login/methods/headless/sms/bind' \
--header "Content-Type: application/x-www-form-urlencoded"
```

Blitz Identity Provider отправит пользователю SMS с кодом подтверждения и вернет ответ:

```
{
   "inquire":"enter_sms_code",
   "contact":"+79991234567",
   "ttl":300,
   "remain_attempts":3
}
```

В полученном ответе указано, сколько секунд у пользователя остается для отправки кода на проверку (ttl), сколько попыток ввести код у него есть (remain_attempts), на какой номер телефона ему был отправлен код (`contact`).

Для проверки введенного пользователем кода подтверждения приложение должно направить в AJAX к Blitz Identity Provider запрос HTTP POST (обязательно с `withCredentials: true`) на URL `https://login.company.com/blitz/login/methods/headless/sms/bind` с `Content-Type x-www-form-urlencoded` и Body, содержащим `sms-code` с кодом подтверждения.

Пример запроса:

```
curl -v -b cookies.txt -c cookies.txt \
--request POST 'https://login.company.com/blitz/login/methods/headless/sms/bind' \
--header "Content-Type: application/x-www-form-urlencoded" \
--data 'sms-code=123456'
```

Если код неправильный, то Blitz Identity Provider вернет ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"invalid_otp",
            "params":{}
        }
    ],
    "contact":"+79991234567",
    "remain_attempts":2,
    "ttl":276
}
```

Если количество попыток проверки кода закончилось, то Blitz Identity Provider вернет ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"no_attempts",
            "params":{}
        }
    ]
}
```

Если срок действия кода истек, то Blitz Identity Provider вернет ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"expired",
            "params":{}
        }
    ]
}
```

В случае этой ошибки можно запросить отправку нового кода подтверждения. Для этого приложение должно вызвать Blitz Identity Provider следующим образом:

```
curl -v -b cookies.txt -c cookies.txt \
--request POST 'https://login.company.com/blitz/login/methods/headless/sms/bind' \
--header "Content-Type: application/x-www-form-urlencoded" \
--data 'sms-send=sms'
```

Если запросить переотправку кода до истечения срока действия предыдущего, то вернется ошибка:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"code_not_expired",
            "params":{}
        }
    ]
}
```

Если общее количество попыток подтверждения входа по коду подтверждения из SMS превышено, то Blitz Identity Provider осуществляет временное блокирование подтверждения входа для учетной записи по коду подтверждения. В этом случае при очередной попытке ввода неправильного кода подтверждения Blitz Identity Provider может вернуть ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"method_temp_locked",
            "params":{}
        }
    ]
}
```

Если введенный код подтверждения правильный, и этого достаточно для завершения входа, то Blitz Identity Provider автоматически перенаправит пользователя на адрес обработчика `redirect_uri`, добавив к запросу код авторизации и параметр `state`. Используя полученный код авторизации приложение продол-

жит стандартное взаимодействие по OpenID Connect для получения маркеров безопасности и данных учетной записи.

Пример ответа в случае успешного входа:

```
…
< HTTP/2 302
…
< location: https://…?code=…&state=…
…
```

Глава 4

Модули

В данном разделе вы найдете подробную информацию по дополнительным модулям Blitz Identity Provider.

4.1 Шлюз безопасности Blitz Keeper

4.1.1 О модуле Blitz Keeper

С помощью Blitz Identity Provider можно осуществлять контроль доступа при вызове приложениями защищаемых сервисов.

Обеспечение авторизации при вызове приложениями сервисов основано на спецификациях OAuth 2.0. Перед использованием сервисов приложение должно получить у Blitz Identity Provider маркер доступа (`access_token`). Для получения маркера доступа приложению доступны *различные способы взаимодействия* (страница 383). При этом маркер доступа может быть получен:

- в контексте входа пользователя – маркер будет включать информацию о пользователе и наборе согласий (разрешений), предоставленных пользователем приложению;
- на приложение вне контекста входа пользователя – маркер будет включать набор согласий (разрешений) из числа разрешенных приложению.

Далее с использованием полученного маркера доступа приложение может вызывать сервисы. При этом будут следующие сложности:

- внутри каждого сервиса необходимо будет реализовывать собственную логику авторизации – проверять предоставленный маркер доступа, извлекать из него информацию о пользователе и предоставленных согласиях (разрешениях) и анализировать, достаточно ли их для выполнения сервиса или нет. Осуществлять протоколирование принятого решения по доступу.
- приложение будет использовать единый маркер доступа для вызова различных сервисов. Маркер доступа в таком случае может содержать больше информации о пользователе и больший набор согласий (разрешений), чем нужно конкретному вызванному сервису. Это будет нарушать принцип наименьших привилегий – сервис получит больше прав доступа, чем ему необходимо для выполнения своей задачи.

Чтобы решить вышеописанные сложности в Blitz Identity Provider предусмотрено специальное приложение – шлюз безопасности (`blitz-keeper`). Это приложение представляет собой специализированный прокси-сервер, используемый при вызове защищаемых сервисов – приложение вызывает сервисы не напрямую, а через шлюз безопасности. При этом шлюз безопасности берет на себя выполнение следующих задач:

- Проверяет включенный в вызов сервиса заголовок авторизации, извлекает из заголовка маркер доступа и, во взаимодействии с сервисом авторизации (`blitz-idp`) выполняет проверку, действителен ли маркер доступа, а также, достаточно ли у пользователя и приложения прав для вызова защищаемого сервиса.

- Во взаимодействии с сервисом авторизации (`blitz-idp`) заменяет маркер доступа таким образом, чтобы передаваемый от шлюза безопасности к защищаемому сервису маркер безопасности содержал только тот набор сведений о пользователе и разрешений, который необходим для работы защищаемого сервиса. При этом из маркера безопасности могут быть как изъяты излишние разрешения и сведения о пользователе, так и наоборот, добавлены в маркер доступа дополнительные разрешения и сведения, если такое установлено политикой безопасности.
- Протоколирует в журнале событий безопасности Blitz Identity Provider события успешной и неуспешной проверки прав доступа.

Взаимодействие шлюза безопасности с сервисом авторизации осуществляется на основе спецификации OAuth 2.0 Token Exchange[122]. Иллюстрация взаимодействия приведена на схеме.

Настройка использования шлюза безопасности для защиты сервисов описана в последующих разделах.

4.1.2 Установка сервиса blitz-keeper

Важно: См. *системные требования* (страница 12).

Для установки сервиса `blitz-keeper` используется установщик `blitz-keeper-5.X.X.bin`.

При установке сертифицированной версии дополнительно используется файл `blitz-keeper-thirdparty-5.X.X.tar.gz`, содержащий архив с используемыми сторонними библиотеками.

Для установки `blitz-keeper` выполните следующие действия:

1. Скопируйте на предназначенный для установки шлюза безопасности сервер (например, в директорию `/tmp`) из дистрибутива Blitz Identity Provider файл `blitz-keeper-5.X.X.bin`.

   В случае установки сертифицированной версии необходимо также скопировать `blitz-keeper-thirdparty-5.X.X.tar.gz`.

2. Запустите установщик `blitz-keeper-5.X.X.bin`:

```
cd /tmp
chmod +x blitz-keeper-5.X.X.bin
./blitz-keeper-5.X.X.bin
```

[122] https://tools.ietf.org/html/rfc8693

В ответ на вопросы установщика задайте значение JAVA_HOME – директории, в которую на сервере установлен JDK.

Для сертифицированной версии также потребуется задать путь к файлу blitz-keeper-thirdparty-5.X.X.tar.gz.

Установка будет произведена в директорию /usr/share/identityblitz.

3. Добавьте сервис blitz-keeper в автозапуск и запустите его:

```
systemctl enable blitz-keeper
systemctl start blitz-keeper
```

4. Скорректируйте блок настроек балансировки в конфигурационном файле nginx (каталог /etc/nginx/conf.d):

```
upstream blitz-keeper {
    server [BLITZ-KPR-NODE-01]:9012 max_fails=3 fail_timeout=120;
    server [BLITZ-KPR-NODE-02]:9012 max_fails=3 fail_timeout=120;
}
```

Примечание: [BLITZ-%%%-NODE-XX] – имена (hostname) серверов с сервисом blitz-keeper.

4.1.3 Настройка Blitz Keeper

Настройка шлюза безопасности Blitz Keeper осуществляется путем редактирования конфигурационного файла blitz-keeper.conf, расположенного в каталоге /etc/blitz-keeper. Пример конфигурационного файла:

```
{
  "authenticators": {
    "prod-auth": {
      "type": "token-exchange",
      "te": "https://blitz-host/blitz/oauth/te",
    },
  },
  "services" : {
    "api-1":{
      "display-name" : "secured services",
      "host": "service-host.com",
      "locations": {
        "/api/service1/**": {
          "methods" : ["GET","POST"],
          "authenticator": "prod-auth",
          "required-scopes": ["scope1","scope2"]
        },
        "/path/api/user/*/getdata/**": {
          "methods" : ["GET","PUT"],
          "authenticator": "prod-auth",
          "required-scopes": ["scope3"]
        }
      }
    }
  }
}
```

В блоке authenticators нужно зарегистрировать все используемые сервисы авторизации blitz-idp. Обычно достаточно использовать один единственный сервис авторизации для защиты

сервисов, и тогда нужно заполнить только один блок как в примере (в примере зарегистрирован один сервис авторизации с именем `prod-auth`). Если в системе используется несколько раздельных установок Blitz Identity Provider (например, ПРОД- и ТЕСТ-среда или внутренний контур для сотрудников и внешний контур для клиентов), то можно использовать общий шлюз безопасности, который будет взаимодействовать с несколькими разными сервисами авторизации – тогда нужно в блоке `authenticators` задать настройки нескольких сервисов авторизации. Для каждого сервиса авторизации задается имя (в примере использован `prod-auth`, но можно задать любое имя). В блоке настроек сервиса авторизации задается тип взаимодействия (`type`) в значении token-exchange (пока это единственный поддерживаемый тип взаимодействия) и адрес (`te`) вызова обработчика Token Endpoint сервиса авторизации. Если blitz-keeper развернут на отдельных серверах, то рекомендуется задать адрес обработчика с https и доменным именем. Если приложение `blitz keeper` развернуто на том же сервере что сервис авторизации `blitz-idp`, то рекомендуется задать в `te` локальное имя, например, `http://localhost:9000/blitz/oauth/te`.

В блоке `services` нужно зарегистрировать защищаемые сервисы. Для всех защищаемых сервисов можно создать общий блок настроек или несколько отдельных блоков. Каждый блок имеет имя (в примере, `api-1`). Внутри блока задаются настройки:

- `display-name` – текстовое описание сервиса (любой комментарий или описание);
- `host` – адрес сервера защищаемого сервиса;
- `locations` – допустимые пути и операции вызова сервиса.

В блоке `locations` указываются настройки всех путей сервиса и разрешенных методов. В качестве имени каждого вложенного блока указывается адрес сервиса. Допустимо в адресе использовать звездочку (`*`), чтобы указать на пропуск отдельного компонента в адресе пути сервиса и допустимо использовать двойную звездочку (`**`), чтобы указать, что вся оставшаяся часть пути сервиса может быть любая. Внутри вложенного блока с адресом сервиса можно опционально перечислить разрешенные методы сервиса (настройка `methods`), указать имя используемого сервиса авторизации (настройка `authenticator`) и перечень разрешений (настройка `required-scopes`) для целевого маркера доступа, которые будут включены в маркер доступа, передаваемый в защищаемый сервис.

После изменения настроек в `blitz-keeper.conf` необходимо перезапустить шлюз безопасности.

4.1.4 Создание правил доступа к сервисам

См. *общее описание* (страница 367) создания списка правил доступа к защищаемым сервисам по Token Exchange.

4.1.5 Настройка обмена маркеров доступа

См. *общее описание* (страница 371) настройки обмена маркеров доступа по Token Exchange.

4.1.6 Просмотр логов

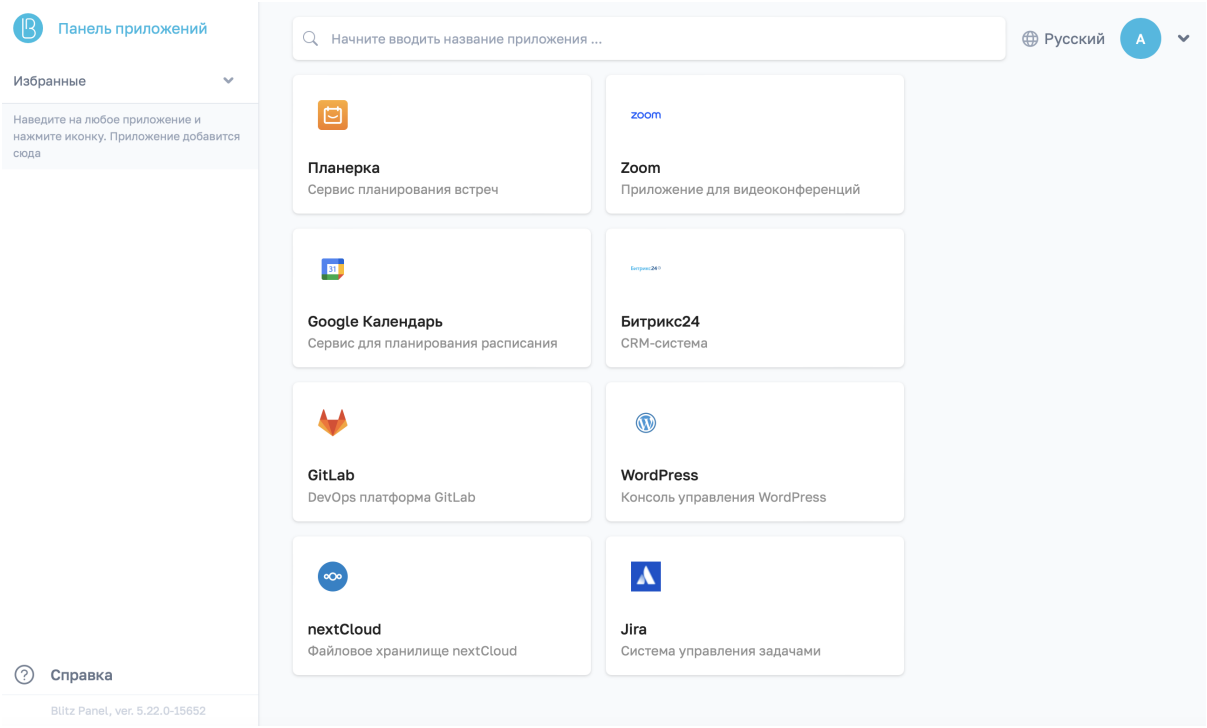
Работа сервиса `blitz-keeper` записывается в отдельный лог. Для просмотра лога откройте файл `blitz-keeper.log` в директории `/var/log/identityblitz/`.

```
sudo vim /var/log/identityblitz/blitz-keeper.log
```

4.2 Витрина с приложениями Blitz Panel

4.2.1 О модуле Blitz Panel

Модуль Blitz Panel предназначен для создания витрины, предоставляющей быстрый доступ пользователей к подключенным приложениям. На странице витрины пользователи также могут выбирать язык, добавлять часто используемые приложения в Избранные и переходить в Личный кабинет.



Администрирование модуля выполняется посредством сервиса `blitz-panel`.

4.2.2 Установка сервиса blitz-panel

Важно: См. *системные требования* (страница 12).

Для установки сервиса `blitz-panel` используется установщик `blitz-panel.bin`.

Важно: Сервис `blitz-panel` можно установить на любой сервер, где установлен сервер Blitz Identity Provider.

Для установки `blitz-panel` выполните следующие действия:

1. На предназначенный для установки сервер скопируйте (например, в директорию `/tmp`) из дистрибутива Blitz Panel файл `blitz-panel.bin`.
2. Запустите установщик `blitz-panel.bin`, указав в качестве параметра запуска `-j` значение `JAVA_HOME` – директории, в которую на сервере установлен JDK.

   Установка будет произведена в директорию `/usr/share/identityblitz/blitz-panel`.

```
cd /tmp
chmod +x blitz-panel.bin
./blitz-panel.bin -- -j <JAVA_HOME>
```

3. Создайте файл `panel.conf` с первичными настройками Blitz Panel:
   - `IDP_DOMAIN` – имя домена, на котором функционирует Blitz Identity Provider;
   - `CLIENT_ID` – идентификатор для подключения приложения Blitz Panel к Blitz Identity Provider по протоколу OAuth 2.0.

     Внимание: В `client_id` недопустимо использовать двоеточие и тильду.

   - `CLIENT_SECRET` – секретный ключ для подключения приложения Blitz Panel к Blitz Identity Provider по протоколу OAuth 2.0.
   - `PANEL_DOMAIN` – имя домена, на котором будет функционировать Blitz Panel.

   Список 1: Пример конфигурационного файла

   ```
   IDP_DOMAIN=mydomain.com
   CLIENT_ID=qwerty12345
   CLIENT_SECRET=54321ytrewq
   PANEL_DOMAIN=mydomain.com/panel
   ```

4. Запустите скрипт первоначальной настройки Blitz Panel, указав путь к файлу `panel.conf`.

   ```
   /usr/share/identityblitz/blitz-panel/bin/configure -f /tmp/panel.conf
   ```

   В результате выполнения скрипта будут настроены конфигурационные файлы Blitz Panel.

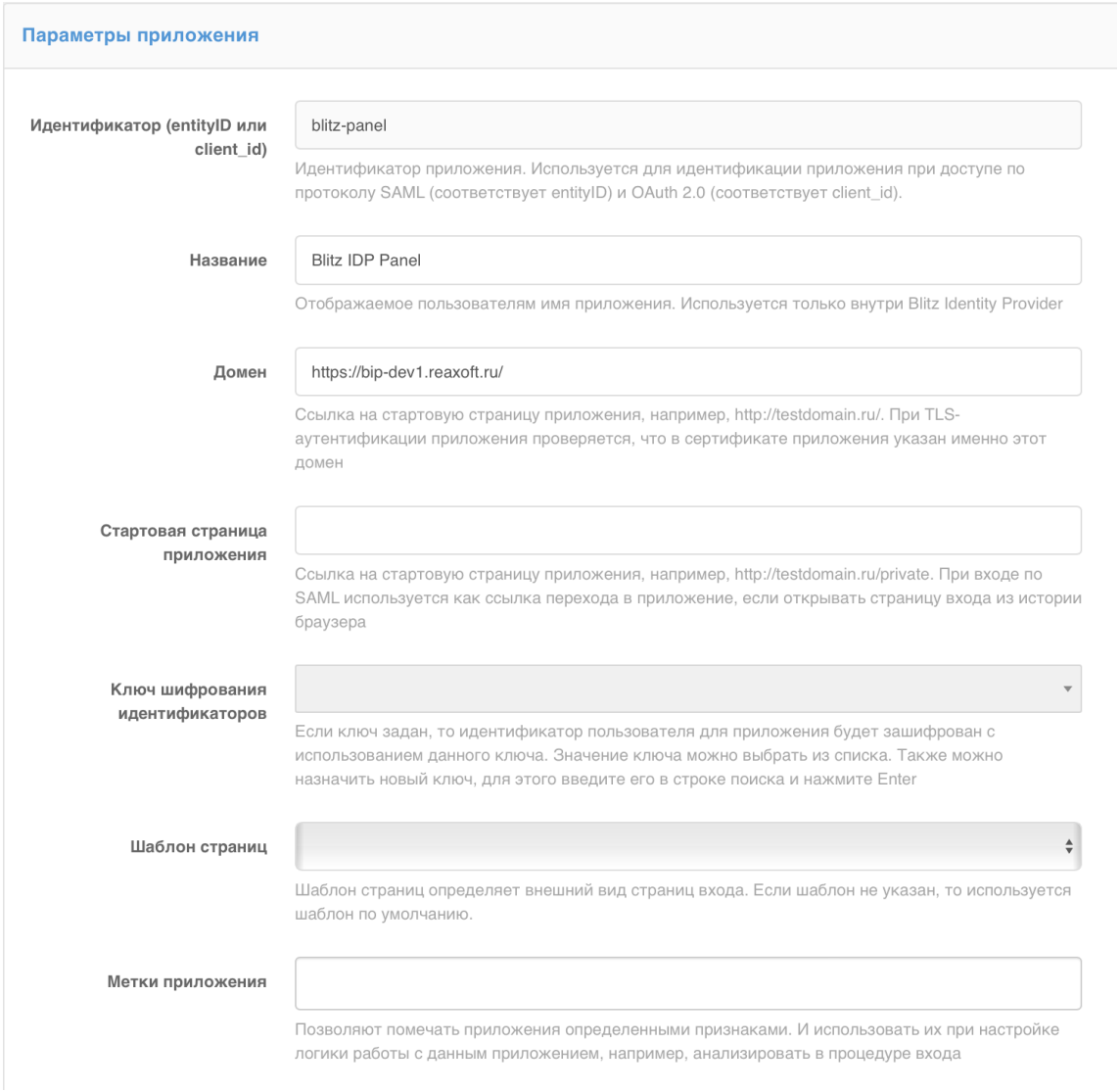
4.2.3 Настройка витрины

Для настройки витрины Blitz Panel выполните следующие действия:

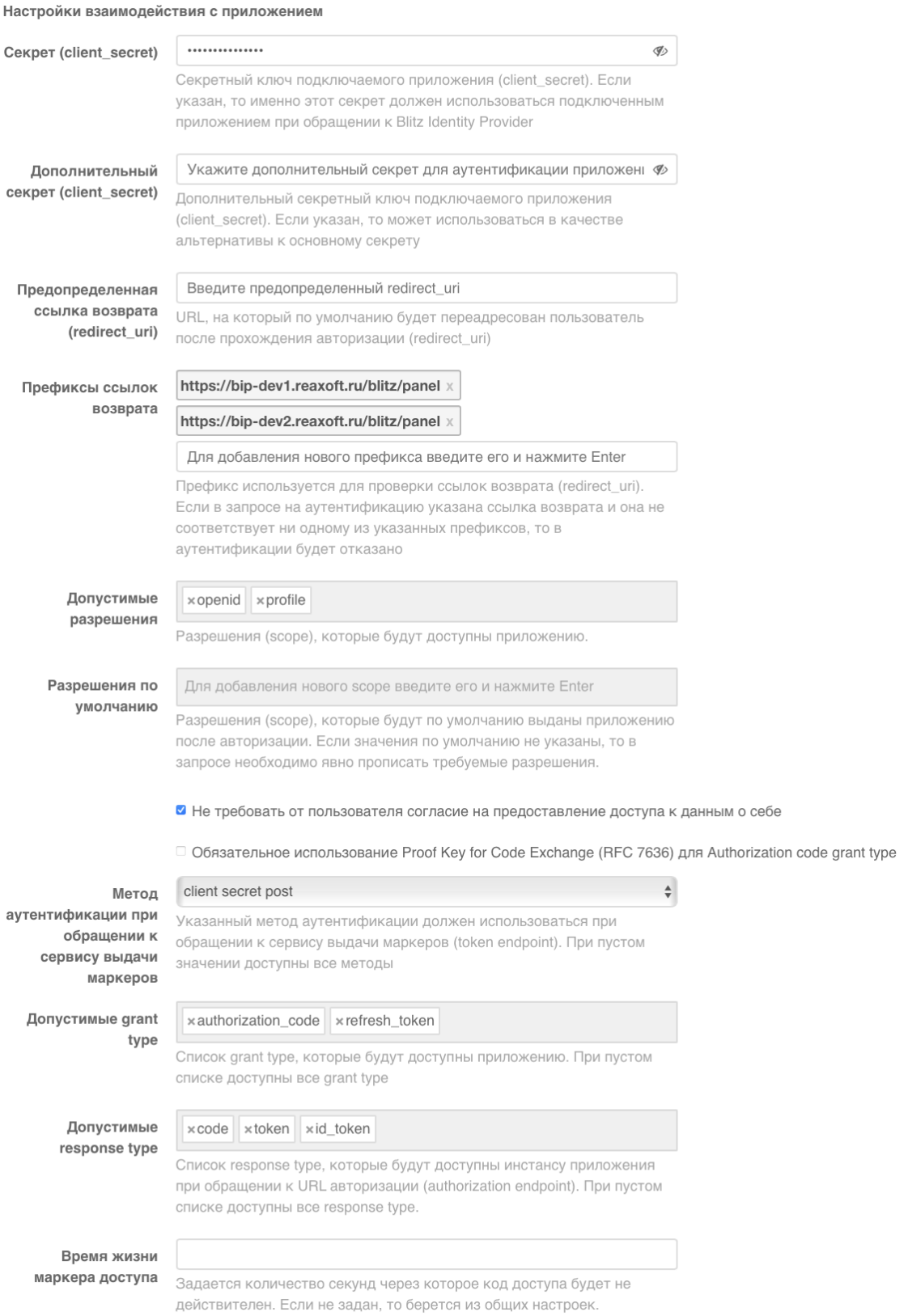
1. Положите в каталог `/usr/share/identityblitz/blitz-panel/static/resources/icons/` иконки приложений.

   Примечание: Поддерживаются следующие форматы:
   - SVG,
   - PNG, максимум 128px по минимальной стороне.

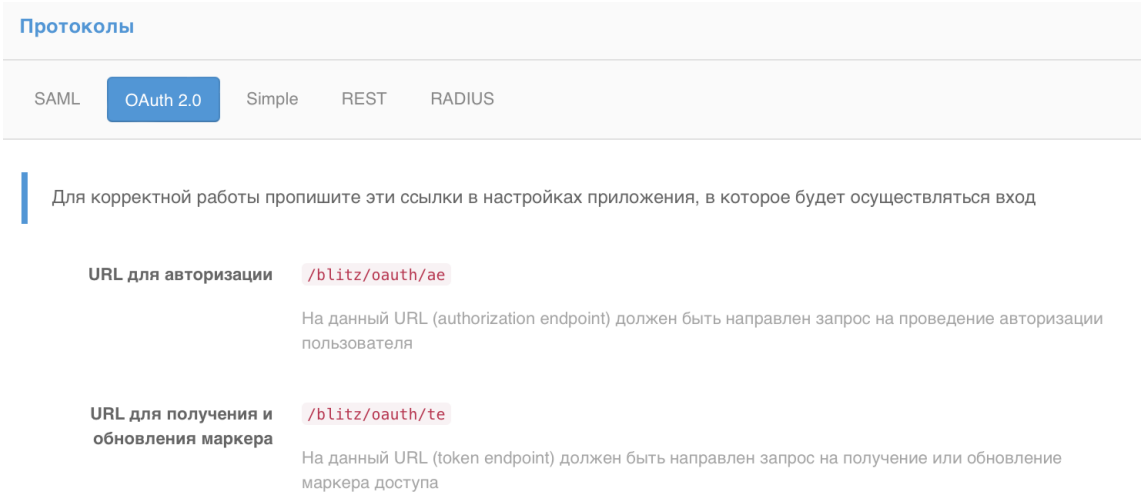
2. В Blitz Identity Provider *создайте* (страница 238) приложение для подключения Blitz Panel к Blitz Identity Provider по протоколу OAuth 2.0.



Укажите заданные при установке Blitz Panel `client_id` и `client_secret`.



Адреса запросов к Blitz Identity Provider для аутентификации пользователя, указанные в настройках протокола, далее необходимо указать в файле конфигурации `/etc/blitz-panel/app.conf`.



Совет: Значения `client_id` и `client_secret` при необходимости можно поменять в том же файле конфигурации.

3. Откройте файл конфигурации `/etc/blitz-panel/app.conf`. В секции `session` -> `oauth` задайте настройки подключения приложения Blitz Panel к Blitz Identity Provider по протоколу OAuth 2.0.

   - `name`: произвольное имя подключения;
   - `clientId`: проверьте, совпадает ли идентификатор приложения `client_id` с указанным в Blitz Identity Provider.
   - `clientSecret`: проверьте, совпадает ли секретный ключ приложения с указанным в Blitz Identity Provider.
   - `logoutUrl`: URL, на который Blitz Panel будет направлять запрос в Blitz Identity Provider на выход пользователя.
   - `authUrl`: URL, на который Blitz Panel будет направлять запрос в Blitz Identity Provider на проведение авторизации пользователя.
   - `tokenUrl`: URL, на который Blitz Panel будет направлять запрос в Blitz Identity Provider на получение или обновление маркера доступа.
   - `me`: URL, на который Blitz Panel будет направлять запрос в Blitz Identity Provider на получение данных о пользователе (`url`), и атрибут для поиска пользователя в хранилище (`subjectIdAttr`).
   - `scopes`: список разрешений, которые будут доступны Blitz Panel.

```
"session": {
    "oauth": {
        "name": "Blitz IdP",
        "clientId": "CHANGE_CLIENT_ID",
        "clientSecret": "CHANGE_CLIENT_SECRET",
        "logoutUrl": "https://CHANGE_IDP_DOMAIN/blitz/login/logout",
        "authUrl": "https://CHANGE_IDP_DOMAIN/blitz/oauth/ae",
        "tokenUrl": "https://CHANGE_IDP_DOMAIN/blitz/oauth/te",
        "me": {
            "url": "https://CHANGE_IDP_DOMAIN/blitz/oauth/me",
            "subjectIdAttr": "sub"
        },
        "scopes": [
            "openid",
            "profile"
        ]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    },
    ...
},
...
```

4. При необходимости задайте параметры сессии пользователя: URL, на который будет переадресован пользователь после выхода, значение TTL, максимальный период отсутствия активности в секундах, период проверки наличия сессии в миллисекундах, имя создаваемой `cookie` и др.

```
"session": {
    ...
    "postLogoutUrl": "/blitz/panel",
    "ttlInSec": 36000,
    "inactivityPeriodInSec": 3600,
    "checkSessionPeriodInMs": 1000,
    "cookie": {
        "name": "scs",
        "path": "/blitz/panel",
        "transient": true
    },
    "useCompression": false,
    "encodingKey": "CHANGE_SCS_ENC",
    "hmacKey": "CHANGE_SCS_HMAC"
},
...
```

5. Секция `apps` -> `sources` содержит группы приложений, которые можно сформировать по произвольным признакам (статические, динамические и пр.). Для каждой группы указывается имя, список приложений в составе группы и правила, определяющие, для каких пользователей отображаются приложения.

   В секции `apps` -> `sources`-> `rules` задайте правила, определяющие, для каких пользователей будут отображаться те или иные приложения.

   Каждое правило состоит из следующих частей:

   - `name`: имя правила.
   - `conditions`: условия выбора пользователей.

     Поддерживается два типа условий:

     – `"typ": "userGroup"` — *группа пользователей* (страница 213). Необходимо указать имя профиля группы и ее идентификатор.

     – `"typ": "userClaims"` — гибкий выбор пользователей на основании утверждений относительно их атрибутов. Условие этого типа может содержать утверждения по нескольким атрибутам. Для того чтобы пользователь был выбран согласно условию, он должен удовлетворять **всем** утверждениям в нем.

     Внимание: Правило может содержать несколько условий. Правило применяется к пользователю, если он удовлетворяет **хотя бы** одному из них.

   - `tags`: метки, связывающие условия выбора пользователей и приложения.

     Поддерживаются следующие типы меток:

     – произвольный параметр (например, `role`, `department` и пр.);

     – идентификатор приложения (задается в списке `appId`).

Внимание: Правило применяется к приложению, если в настройках приложения (см. следующий шаг) присутствует хотя бы одно из значений, заданных в данном разделе.

6. В секции `apps` -> `sources` -> `apps` задайте список подключенных к Blitz Identity Provider приложений, которые будут отображаться на витрине. Для каждого приложения укажите следующие параметры:

   - `id`: идентификатор приложения в Blitz Identity Provider.
   - `name`: имя приложения, которое будет отображаться на витрине, на необходимых языках.
   - `url`: URL стартовой страницы приложения.
   - `icon`: имя файла иконки в каталоге:

     ```
     /usr/share/identityblitz/blitz-panel/static/resources/icons/``
     ```

   - `tags`: метки, определяющие для каких пользователей на витрине будет отображаться приложение согласно заданным выше правилам (опционально).
   - `desc`: описание приложения на необходимых языках.

Список 2: Пример настройки правил и списка приложений

```
"apps": {
    "sources": [
        {
            "name": "Static Applications",
            "type": "static",
            "apps": [
                {
                    "id": "dev_portal",
                    "name": {
                      "ru": "Портал разработчика 24"
                    },
                    "url": "https://my.domain.com/dev/portal",
                    "icon": "confluence.svg",
                    "tags": {
                        "role": [
                            "admin",
                            "sys_admin"
                        ]
                    }
                },
                {
                    "id": "jira",
                    "url": "https://my.domain.com/dev/jira",
                    "name": {
                      "ru": "Jira"
                    },
                    "icon": "jira.svg",
                    "tags": {
                        "role": [
                            "admin",
                            "sys_admin"
                        ]
                    }
                },
                {
                    "id": "test-app",
                    "url": "https://my.domain.com/dev/test",
                    "name": {
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                "ru": "Тестовое приложение"
            }
        },
        {
            "id": "atom",
            "url": "https://my.domain.com/dev/atom",
            "name": {
              "ru": "Atom"
            },
            "desc": {
              "ru": "Редактор кода Atom",
              "en": "Atom is your essential companion"
            }
        },
        {
            "id": "call_center",
            "url": "https://my.domain.com/dev/call",
            "name": {
              "ru": "Центр обработки звонков"
            },
            "desc": {
              "ru": "Для управления Call-центром"
            },
            "tags": {
                "role": [
                    "admin",
                    "sys_admin"
                ]
            }
        },
        {
            "id": "web_mail",
            "name": {
              "ru": "Корпоративная почта"
            },
            "desc": {
              "ru": "Web-интерфейс корпоративной почты"
            },
            "icon": "gmail.svg",
            "url": "https://my.domain.com/dev/portal",
            "tags": {
                "role": [
                    "sys_admin"
                ]
            }
        },
        {
            "id": "yandex",
            "url": "https://my.domain.com/dev/yandex",
            "name": {
              "ru": "Поисковая система и почта"
            },
            "desc": {
              "ru": "Web-интерфейс поисковой системы Yandex"
            }
        }
    ],
    "rules": [
        {
            "name": "admin_role",
            "conditions": [
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        {
                            "typ": "userGroup",
                            "profile": "main_group_profile",
                            "id": "app_admin"
                        },
                        {
                            "typ": "userClaims",
                            "claims": {
                                "company_type": "IT",
                                "position": [
                                    "head",
                                    "master"
                                ]
                            }
                        },
                        {
                            "typ": "userClaims",
                            "claims": {
                                "company_name": "Моя компания"
                            }
                        }
                    ],
                    "tags": {
                        "appId": [
                            "dev_portal",
                            "yandex"
                        ],
                        "role": [
                            "admin",
                            "sys_admin"
                        ]
                    }
                },
                {
                    "name": "atom",
                    "conditions": [
                        {
                            "typ": "userClaims",
                            "claims": {
                                "tags": [
                                    "atom"
                                ]
                            }
                        }
                    ],
                    "tags": {
                        "appId": [
                            "atom"
                        ]
                    }
                }
            ]
        }
    ]
},
...
```

7. Добавьте сервис `blitz-panel` в автозапуск и запустите его:

```
systemctl enable blitz-panel
systemctl start blitz-panel
```

4.2.4 Дизайн и локализация витрины

Изменение внешнего вида

При необходимости вы можете изменить внешний вид витрины, внеся изменения в файлы каталога `/usr/share/identityblitz/blitz-panel/static`. Возможна кастомизация следующих элементов:

- значок вкладки (favicon);
- шаблон `index.html`;
- стили CSS (`.../resources/styles.css`).

Добавление языка

Для добавления языка положите файл с переводом `<двухбуквенный код языка>.json` (например, `ar.json` для арабского языка) в каталог `/usr/share/identityblitz/blitz-panel/static/resources/locales` и перезапустите сервис `blitz-panel`.

```
sudo systemctl restart blitz-panel
```

Новый язык появится в меню выбора языков витрины.

> **Примечание:** Локализация названия и описания вынесенных на витрину приложений *осуществляется* (страница 551) в файле `/etc/blitz-panel/app.conf`.

4.2.5 Просмотр логов

Работа сервиса `blitz-panel` записывается в отдельный лог. Для просмотра лога откройте файл `blitz-panel.log` в директории `/var/log/identityblitz/`.

```
sudo vim /var/log/identityblitz/blitz-panel.log
```