



# Blitz Identity Provider

*Версия 5.28*

**Руководство администратора**

19 марта 2025

# Оглавление

<b>1</b>	<b>Развертывание</b>	<b>1</b>
1.1	Архитектура развертывания	1
1.2	Системные требования	2
1.2.1	Операционные системы	2
1.2.2	Минимальные требования	2
1.2.3	Рекомендуемые требования для кластера	4
1.3	Общая инструкция по установке	7
1.3.1	Шаг 1. JDK	8
1.3.2	Шаг 2. Memcached	8
1.3.3	Шаг 3. СУБД	9
1.3.4	Шаг 4. RabbitMQ	11
1.3.5	Шаг 5. Blitz Identity Provider	12
1.3.6	Шаг 6. Синхронизация файлов конфигурации	15
1.3.7	Шаг 7. Веб-сервер	18
1.3.8	Шаг 8. LDAP-каталог	20
1.4	Экспресс-инструкции для разных ОС	21
1.4.1	Ограничения при использовании инструкций	22
1.4.2	Astra, Альт, Основа, РЕД ОС, РОСА	22
	Шаг 1. JDK	22
	Шаг 2. Memcached	23
	Шаг 3. PostgreSQL	24
	Шаг 4. RabbitMQ	31
	Шаг 5. 389 Directory Server	33
	Шаг 6. Nginx	35
	Шаг 7. Blitz Identity Provider	37
1.4.3	Rocky Linux, AlmaLinux, Oracle Linux и RHEL	43
	Шаг 1. JDK	44
	Шаг 2. Memcached	44
	Шаг 3. PostgreSQL	44
	Шаг 4. RabbitMQ	47
	Шаг 5. 389 Directory Server	49
	Шаг 6. Nginx	49
	Шаг 7. Blitz Identity Provider	50
1.5	Первые шаги после установки	52
1.5.1	Настройка опций запуска приложений Blitz Identity Provider	52
1.5.2	Вход в консоль управления	55
1.5.3	Установка лицензионного ключа	56
1.5.4	Управление учетными записями администраторов	56
1.5.5	Перезапуск сервисов Blitz Identity Provider	57
1.5.6	Удаление использованных при установке файлов	58
<b>2</b>	<b>Базовое конфигурирование</b>	<b>59</b>
2.1	Атрибуты учетных записей	59
2.1.1	Что представляет собой атрибут учетной записи	59
2.1.2	Конфигурирование доступных атрибутов	60
	Хранимые атрибуты	60

Вычисляемые атрибуты . . . . .	61
Правила преобразования входных значений . . . . .	62
Правила преобразования выходных значений . . . . .	63
Настройка назначения атрибутов . . . . .	63
2.1.3 Подключение хранилищ атрибутов . . . . .	64
Типы хранилищ . . . . .	64
Подключение хранилища по LDAP . . . . .	65
Подключение к хранилищу по REST . . . . .	69
Сервис поиска пользователей . . . . .	70
Сервис получения данных пользователя . . . . .	71
Сервис проверки логина и пароля . . . . .	72
Сервис смены пароля пользователем . . . . .	72
Сервис добавления нового пользователя . . . . .	73
Сервис изменения данных пользователя . . . . .	74
Сервис удаления пользователя . . . . .	75
Настройка внутреннего хранилища . . . . .	75
2.2 Аутентификация . . . . .	76
2.2.1 Как работать с настройками аутентификации . . . . .	76
2.2.2 Общие настройки . . . . .	78
2.2.3 Парольные политики . . . . .	80
2.2.4 Работа с ключами безопасности . . . . .	82
Настройка ключей безопасности . . . . .	82
Вход с помощью WebAuthn, Passkey, FIDO2 . . . . .	84
Подтверждение входа с помощью WebAuthn, Passkey, FIDO2, U2F . . . . .	85
2.2.5 TOTP-генераторы . . . . .	86
2.2.6 Вход по логину и паролю . . . . .	87
2.2.7 Вход с помощью средства электронной подписи . . . . .	90
Настройка метода аутентификации в консоли управления . . . . .	90
Использование и обновление плагина . . . . .	93
2.2.8 Вход через внешние сервисы идентификации . . . . .	93
2.2.9 Вход с помощью прокси-аутентификации . . . . .	94
2.2.10 Вход с помощью сеанса операционной системы . . . . .	95
Настройки контроллера домена (Kerberos-сервера) . . . . .	95
Настройки в консоли управления Blitz Identity Provider . . . . .	97
Настройки браузеров пользователей . . . . .	99
Настройки запуска приложений Blitz Identity Provider . . . . .	101
Настройки веб-сервера . . . . .	101
Отладка проблем с входом по сеансу операционной системы . . . . .	101
2.2.11 Вход с помощью электронной почты . . . . .	102
Шаг 1. Добавление метода в blitz.conf . . . . .	102
Шаг 2. Настройка метода в консоли . . . . .	102
2.2.12 Вход с помощью кодов подтверждения . . . . .	104
2.2.13 Вход с известного устройства . . . . .	106
2.2.14 Вход по разовой ссылке . . . . .	106
2.2.15 Вход по QR-коду . . . . .	107
2.2.16 Вход по входящему звонку . . . . .	107
Шаг 1. Добавление метода в blitz.conf . . . . .	108
Шаг 2. Настройка метода в консоли . . . . .	108
2.2.17 Вход разовым паролем на основе времени (TOTP) . . . . .	112
Шаг 1. Добавление метода в blitz.conf . . . . .	112
Шаг 2. Настройка метода в консоли . . . . .	113
2.2.18 Автоматическая идентификация пользователя по свойствам сессии . . . . .	113
Шаг 1. Добавление метода в blitz.conf . . . . .	114
Шаг 2. Настройка метода в консоли . . . . .	114
Шаг 3. Создание процедуры входа . . . . .	116
Шаг 4. Кастомизация текстов . . . . .	117
2.2.19 Подтверждение входа разовым паролем на основе состояния (HOTP) . . . . .	118
2.2.20 Подтверждение входа разовым паролем на основе времени (TOTP) . . . . .	119

2.2.21	Привязка устройств к учетным записям пользователей	120
	Привязка аппаратных брелоков	120
	Привязка мобильного приложения	121
2.2.22	Коды подтверждения, отправляемые в SMS и push-уведомлениях	122
2.2.23	Коды подтверждения, отправляемые по электронной почте	123
2.2.24	Подтверждение входа с помощью Duo Mobile	124
2.2.25	Повторное подтверждение при входе с известного устройства	127
2.2.26	Подтверждение ответом на контрольный вопрос	128
	Шаг 1. Добавление метода в blitz.conf	128
	Шаг 2. Создание справочника контрольных вопросов	128
	Шаг 3. Настройка метода в консоли	129
2.2.27	Подтверждение по входящему звонку	130
	Шаг 1. Добавление метода в blitz.conf	130
	Шаг 2. Настройка метода в консоли	131
2.2.28	Подтверждение входа по паролю	134
	Шаг 1. Добавление метода в blitz.conf	134
	Шаг 2. Настройка метода в консоли	134
2.2.29	Настройка внешнего метода аутентификации	135
2.2.30	Настройка процедуры имперсонификации	137
2.3	Внешние поставщики идентификации	137
2.3.1	Как настроить вход через внешние поставщики идентификации	137
2.3.2	Отечественные поставщики	138
	Яндекс	138
	ВКонтакте	140
	Одноклассники	142
	Mail ID	144
	VK ID	146
	Единая система идентификации и аутентификации (ЕСИА)	148
	Цифровой профиль ЕСИА	153
	Сбер ID	158
	T-ID	159
	ВТБ ID	161
	СберБизнес ID	163
	Альфа ID	164
	Mos ID (СУДИР)	166
2.3.3	Вход через другую установку Blitz Identity Provider	168
2.3.4	Настройки связывания учетных записей	170
	Базовая настройка	171
	Расширенная настройка	173
2.4	Пользовательские сервисы	175
2.4.1	Общие настройки	175
2.4.2	Регистрация пользователей	176
	Форма регистрации	176
	Настройки сервиса регистрации	179
	Процедура регистрации	179
	Изменение текста условий использования	180
2.4.3	Личный кабинет	180
	Отображение атрибутов пользователя	180
	Дополнительные параметры	182
2.4.4	Восстановление доступа	183
	Настройки в консоли	183
	Тексты формы	185
2.5	Администрирование пользователей	186
2.5.1	Управление учетными записями	186
2.5.2	Поиск пользователей	187
2.5.3	Добавление пользователя	187
2.5.4	Просмотр и изменение атрибутов пользователя	188
	Редактирование атрибутов	189

Смена пароля . . . . .	190
Просмотр и отвязка внешних поставщиков . . . . .	190
Привязка устройств для 2FA по разовому паролю . . . . .	190
Привязка Duo Mobile . . . . .	191
Управление членством в группах . . . . .	192
Просмотр, назначение и отзыв прав . . . . .	193
Запомненные устройства и браузеры . . . . .	195
Ключи безопасности . . . . .	195
Выданные приложениям разрешения . . . . .	196
2.5.5 Управление группами пользователей . . . . .	196
Включение отображения групп в blitz.conf . . . . .	196
Работа с группами . . . . .	198
2.5.6 Управление правами доступа . . . . .	199
2.5.7 Безопасность . . . . .	200
2.6 Уведомления и отправка сообщений . . . . .	200
2.6.1 Подключение к SMS-шлюзу . . . . .	203
2.6.2 Подключение к сервису отправки push-уведомлений . . . . .	204
2.6.3 Подключение к SMTP-шлюзу . . . . .	206
<b>3 Доступ в приложения и сетевые службы . . . . .</b>	<b>207</b>
3.1 Регистрация приложений в Blitz Identity Provider . . . . .	207
3.1.1 О приложениях . . . . .	207
3.1.2 Создание учетной записи нового приложения . . . . .	207
3.2 Схемы работы технологий SSO . . . . .	212
3.2.1 Подключение веб-приложения по OIDC . . . . .	212
3.2.2 Подключение мобильного приложения по OIDC . . . . .	214
3.2.3 Подключение приложения по SAML . . . . .	215
3.3 Настройка SAML и WS-Federation . . . . .	216
3.3.1 Подключение по SAML 1.0/1.1/2.0 . . . . .	216
3.3.2 Подключение по WS-Federation . . . . .	217
3.3.3 Загрузка SAML-метаданных . . . . .	218
3.3.4 Настройка SAML-атрибутов . . . . .	219
3.4 Настройка OAuth 2.0 и OpenID Connect 1.0 . . . . .	220
3.4.1 Настройка приложения . . . . .	220
3.4.2 Общие настройки OAuth 2.0 . . . . .	225
3.4.3 Добавление атрибутов в маркер идентификации . . . . .	227
3.4.4 Настройка динамической регистрации клиентов OAuth 2.0 . . . . .	230
3.5 Настройка Simple . . . . .	232
3.6 Взаимодействие по REST API . . . . .	236
3.7 Доступ к сетевым службам по RADIUS . . . . .	236
3.7.1 Шаг 1. Конфигурирование сервера RADIUS . . . . .	237
3.7.2 Шаг 2. Настройка приложения . . . . .	240
3.7.3 Шаг 3. Настройка на стороне сетевой службы . . . . .	241
<b>4 Кастомизация работы с помощью программирования на Java . . . . .</b>	<b>242</b>
4.1 Процедуры входа и их создание . . . . .	242
4.1.1 О процедурах входа . . . . .	242
4.1.2 Создание процедуры . . . . .	242
4.2 Готовые процедуры входа . . . . .	244
4.2.1 Принудительная двухфакторная аутентификация . . . . .	245
4.2.2 Ограничение перечня доступных методов первого фактора . . . . .	245
4.2.3 Вход только при определенном значении атрибута . . . . .	246
4.2.4 Запрет входа после истечения срока действия аккаунта . . . . .	248
4.2.5 Вход только из определенных сетей . . . . .	249
4.2.6 Запрет работы в нескольких одновременных сессиях . . . . .	250
4.2.7 Сохранение в утверждениях (claims) перечня групп пользователя . . . . .	250
4.2.8 Отображение пользователю объявления при входе . . . . .	251
Процедура . . . . .	251

	Добавление процедуры в blitz.conf . . . . .	253
4.2.9	Запрос ввода пользователем атрибута или актуализации телефона и email . . . . .	253
4.2.10	Запрос ввода пользователем контрольного вопроса . . . . .	256
4.2.11	Регистрация ключа безопасности (WebAuthn, Passkey, FIDO2) при входе . . . . .	257
4.2.12	Отображение пользователю списка выбора значений при входе . . . . .	259
	Процедура . . . . .	260
	Добавление процедуры в blitz.conf . . . . .	261
4.3	Функции и методы различного назначения в процедурах входа . . . . .	261
4.3.1	Получение геоданных пользователя . . . . .	261
4.3.2	Сброс сессии пользователя . . . . .	263
4.3.3	Кастомные ошибки и их вызов в скрипте . . . . .	265
4.3.4	Анализ меток приложений . . . . .	266
4.4	Кастомизация логики операций с хранилищами данных . . . . .	267
4.4.1	Принцип кастомизации . . . . .	267
4.4.2	Конфигурация . . . . .	267
4.4.3	Написание пользовательской процедуры . . . . .	268
4.5	Процедуры привязки аккаунтов внешних поставщиков . . . . .	268
4.5.1	Регистрация пользователя во внешнем поставщике . . . . .	270
4.5.2	Вычисление имени внешней учетной записи . . . . .	271
4.6	Процедуры обработки RADIUS-запросов . . . . .	272
4.6.1	О RADIUS-процедурах . . . . .	272
4.6.2	Интерфейс RadiusFlow . . . . .	272
4.6.3	Контекст входа RadiusContext . . . . .	273
4.6.4	Интерфейс RadiusResult . . . . .	274
4.6.5	Пример процедуры . . . . .	279
4.6.6	Тексты диалоговых сообщений . . . . .	281
<b>5</b>	<b>Дизайн и тексты интерфейса</b> . . . . .	<b>283</b>
5.1	Страница входа . . . . .	283
5.1.1	Редактирование шаблона по умолчанию . . . . .	283
5.1.2	Создание и изменение новых шаблонов с помощью конструктора . . . . .	286
5.1.3	Создание и изменение новых шаблонов в ручном режиме . . . . .	287
5.2	Личный кабинет . . . . .	290
5.2.1	Логотип в заголовке . . . . .	290
5.2.2	Логотип в футере . . . . .	291
5.2.3	Кастомизация цветовой схемы . . . . .	291
5.3	Мультиязычность . . . . .	292
5.4	Настройки текстов интерфейса . . . . .	294
5.4.1	Текстовые сообщения веб-интерфейса . . . . .	294
5.4.2	Шаблоны писем и SMS-сообщений . . . . .	294
5.4.3	Имена устройств и браузеров . . . . .	299
5.4.4	Сообщения для разных приложений . . . . .	300
5.4.5	Сообщения вспомогательных приложений (pipes) . . . . .	301
5.5	Логотипы кнопок входа через сервисы внешних поставщиков . . . . .	303
<b>6</b>	<b>Настройки конфигурационных файлов</b> . . . . .	<b>304</b>
6.1	Полный список файлов . . . . .	304
6.2	Настройки в файле blitz.conf . . . . .	305
6.2.1	Логины и пароли . . . . .	306
	Количество проверок пароля . . . . .	306
	Смена пароля при входе . . . . .	306
	Системные имена полей логина и пароля . . . . .	306
6.2.2	Атрибуты . . . . .	307
	Внешний валидатор атрибута . . . . .	307
	Транслятор атрибута . . . . .	308
6.2.3	Электронная подпись . . . . .	309
	Вызов внешнего сервиса проверки ЭП . . . . .	309
	Вызов плагина ЭП . . . . .	309

6.2.4	CAPTCHA	311
	reCAPTCHA	311
	Yandex SmartCaptcha	315
6.2.5	Сервер очередей	317
	Отправка событий в сервер очередей	317
	Сервер очередей как брокер сообщений	319
6.2.6	Хранилища и базы	320
	Хранение объектов в Couchbase Server	320
	Считывание конфигурации кластера Couchbase Server	321
	Время хранения объектов	321
	Расширенные настройки подключения к PostgreSQL	322
	Расширенные настройки подключения к LDAP	322
	База геоданных	324
	Использование нескольких СУБД	324
6.2.7	Домен Blitz Identity Provider	326
6.2.8	Пользователи	328
	Блокирование неактивного пользователя	328
	Запрет на использование ID удаленного пользователя	328
6.2.9	ЕСИА	329
	Контейнер ключей для ЕСИА и Цифрового профиля	329
	КриптоПро CSP на Windows	329
	КриптоПро CSP на Linux	335
	КриптоПро JCP	337
	Вход через ЕСИА в режиме выбора организации	337
	Добавление параметров в вызов ЕСИА	341
6.2.10	Сертификаты поставщиков WebAuthn, Passkey, FIDO2, U2F	341
6.2.11	OAuth/OIDC, SAML и внешние поставщики идентификации	342
	Сервис OIDC Discovery	342
	Адреса вызовов внешних поставщиков	342
	Внешний SAML-поставщик	343
	Внешний поставщик СУДИС	345
	Внешний экран согласия	347
6.2.12	Режим регистрации незавершенных попыток входа	348
6.2.13	Передача событий безопасности в файл или Kafka	349
6.2.14	Хранение настроек приложений в отдельных файлах	358
6.2.15	Продолжительность SSO-сессии	359
6.2.16	Разрешения (scopes)	359
6.2.17	REST API	361
6.3	Настройки консоли управления	365
6.3.1	Вход в консоль через SSO	365
6.3.2	Ограничение сессий	367
6.3.3	Роли и права доступа в консоль	368
6.3.4	Смена пароля администратора	368
6.3.5	Работа с тегами приложений	369
6.4	Настройка Token Exchange	369
6.4.1	Шаг 1. Создание правила доступа к сервисам	369
6.4.2	Шаг 2. Настройка обмена маркеров доступа	373
<b>7</b>	<b>Безопасность, обслуживание и устранение неисправностей</b>	<b>375</b>
7.1	Просмотр событий безопасности	375
7.2	Мониторинг функционирования приложений	376
	7.2.1 Стандартный сервис мониторинга	376
	7.2.2 Использование Grafana и Prometheus	378
7.3	Решение проблем	379
7.4	Шлюз безопасности	380
<b>8</b>	<b>Функциональная спецификация</b>	<b>381</b>
<b>9</b>	<b>Рекомендации ФСТЭК</b>	<b>390</b>

9.1	Идентификация и аутентификация субъектов и объектов доступа (ИАФ) . . . . .	390
9.2	Управление доступом субъектов к объектам доступа . . . . .	391
9.3	Регистрация событий безопасности (РСБ) . . . . .	393



# Глава 1

## Развертывание

### 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;
  - `blitz-panel` – витрина, 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 317) или в качестве *брокера сообщений* (страница 319).

Развертывание возможно в конфигурации с *минимальными ресурсами* (страница 2) либо в *кластерной конфигурации* (страница 4).

## 1.2 Системные требования

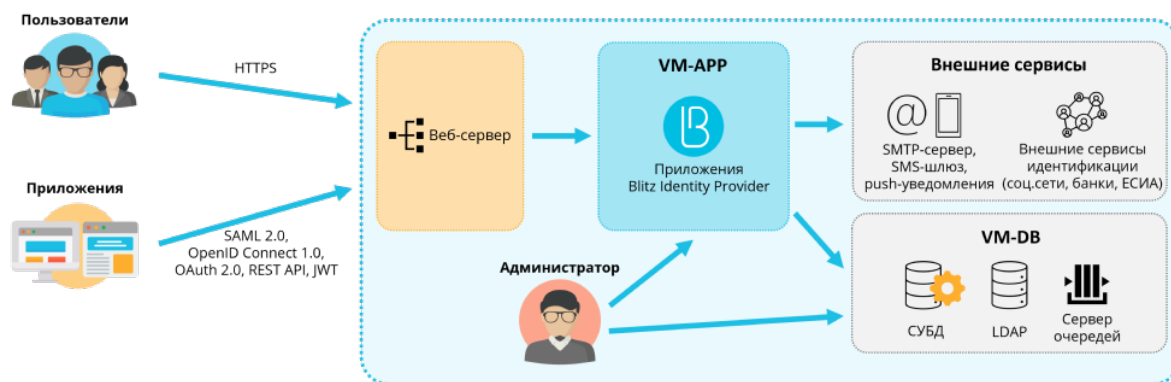
### 1.2.1 Операционные системы

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 7.3/8
- 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

### 1.2.2 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 одну виртуальную машину (далее – VM) со следующими характеристиками и ролями.

### Минимальные требования к серверам для развертывания

#### Развертывание на 1-й VM не в кластере

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

#### Развертывание на 2-х VM не в кластере

Описание	Технические характеристики	ПО
VM для приложений (VM APP)	4 ядра ЦПУ, 8 ГБ ОЗУ, 50 ГБ НЖМД (HDD)	Blitz Identity Provider: blitz-ldap, blitz-console, blitz-registration, blitz-recovery, blitz-keeper, blitz-panel; JDK, nginx или HAProxy, memcached
VM 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 VM в кластере

Описание	Кол-во	Технические характеристики	ПО
VM для приложений и СУБД	2	4 ядра ЦПУ, 8 ГБ ОЗУ, 50 ГБ НЖМД (HDD)	Blitz Identity Provider: blitz-ldap, 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	<a href="https://appleid.apple.com">https://appleid.apple.com</a>
	<a href="https://accounts.google.com">https://accounts.google.com</a>
	<a href="https://graph.facebook.com">https://graph.facebook.com</a> <sup>1</sup>
	<a href="https://oauth.yandex.ru">https://oauth.yandex.ru</a>
	<a href="https://oauth.vk.com">https://oauth.vk.com</a>
	<a href="https://account.mail.ru">https://account.mail.ru</a>
	<a href="https://api.ok.ru">https://api.ok.ru</a>
ЕСИА и цифровой профиль	<a href="https://esia-portal1.test.gosuslugi.ru">https://esia-portal1.test.gosuslugi.ru</a>
	<a href="https://esia.gosuslugi.ru">https://esia.gosuslugi.ru</a>
Банки	<a href="https://online.sberbank.ru">https://online.sberbank.ru</a>
	<a href="https://business.tinkoff.ru">https://business.tinkoff.ru</a>
	<a href="https://id.vtb.ru">https://id.vtb.ru</a>
	<a href="https://sbi.sberbank.ru:9443">https://sbi.sberbank.ru:9443</a>
	<a href="https://fintech.sberbank.ru:9443">https://fintech.sberbank.ru:9443</a>
	<a href="https://id-sandbox.alfabank.ru">https://id-sandbox.alfabank.ru</a>
СУДИР	<a href="https://login.mos.ru">https://login.mos.ru</a>
	<a href="https://login-tech.mos.ru">https://login-tech.mos.ru</a>
	<a href="https://sudir.mos.ru">https://sudir.mos.ru</a>
	<a href="https://sudir-test.mos.ru">https://sudir-test.mos.ru</a>

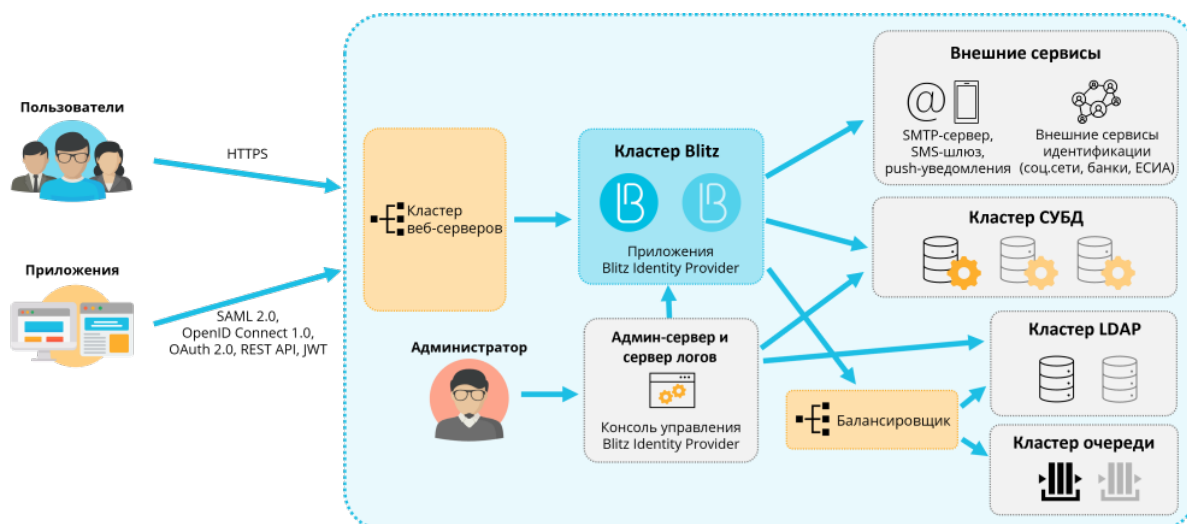
- к SMS-шлюзу (при его использовании);
- к SMTP (при его использовании);
- к сервису push-уведомлений (при его использовании);
- к сервису Kafka (при его использовании для приема событий безопасности).

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

### 1.2.3 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.

<sup>1</sup> 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 – VM) со характеристиками и ролями, указанными в таблице ниже.

### Рекомендуемые требования к серверам для развертывания в кластере

Описание	Кол-во	Технические характеристики	ПО
VM веб-серверов (VM-WEB)	1-2	4 ядра ЦПУ, 4 ГБ ОЗУ, 50 ГБ НЖМД (HDD)	nginx или HAProxy
VM 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
VM для консоли (VM-ADM)	1	2 ядра ЦПУ, 4 ГБ ОЗУ, 100 ГБ НЖМД (HDD)	memcached, JDK; Blitz Identity Provider: blitz-console
VM 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 <sup>3</sup> : 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)
VM для LDAP (VM-LDAP)	2	4 ядра ЦПУ, 8 ГБ ОЗУ, 100 ГБ НЖМД (HDD)	389 Directory Server
VM для сервера очередей (VM-MQ)	1-2	4 ядра ЦПУ, 8 ГБ ОЗУ, 50 ГБ НЖМД (HDD)	RabbitMQ версии 3.7.9
VM для балансировщика (VM-NLB)	1-2	2 ядра ЦПУ, 4 ГБ ОЗУ, 50 ГБ НЖМД (HDD)	HAProxy, keepalived

<sup>3</sup> <https://docs.couchbase.com/server/current/install/install-linux.html>

 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<sup>4</sup> 3 VM.

## • 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);

<sup>4</sup> <https://docs.couchbase.com/server/current/install/deployment-considerations-lt-3nodes.html>

## Ссылки на сервисы внешних поставщиков

Тип	Ссылка
Социальные сети	<a href="https://appleid.apple.com">https://appleid.apple.com</a>
	<a href="https://accounts.google.com">https://accounts.google.com</a>
	<a href="https://graph.facebook.com&lt;sup&gt;2&lt;/sup">https://graph.facebook.com<sup>2</sup></a>
	<a href="https://oauth.yandex.ru">https://oauth.yandex.ru</a>
	<a href="https://oauth.vk.com">https://oauth.vk.com</a>
	<a href="https://account.mail.ru">https://account.mail.ru</a>
ЕСИА и цифровой профиль	<a href="https://esia-portal1.test.gosuslugi.ru">https://esia-portal1.test.gosuslugi.ru</a>
	<a href="https://esia.gosuslugi.ru">https://esia.gosuslugi.ru</a>
Банки	<a href="https://online.sberbank.ru">https://online.sberbank.ru</a>
	<a href="https://business.tbank.ru">https://business.tbank.ru</a> <a href="https://id.tbank.ru">https://id.tbank.ru</a>
	<a href="https://id.vtb.ru">https://id.vtb.ru</a>
	<a href="https://sbi.sberbank.ru:9443">https://sbi.sberbank.ru:9443</a>
	<a href="https://fintech.sberbank.ru:9443">https://fintech.sberbank.ru:9443</a>
	<a href="https://id-sandbox.alfabank.ru">https://id-sandbox.alfabank.ru</a>
СУДИР	<a href="https://login.mos.ru">https://login.mos.ru</a>
	<a href="https://login-tech.mos.ru">https://login-tech.mos.ru</a>
	<a href="https://sudir.mos.ru">https://sudir.mos.ru</a>
	<a href="https://sudir-test.mos.ru">https://sudir-test.mos.ru</a>

- 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`.

### 1.3 Общая инструкция по установке

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

<sup>2</sup> Meta признана экстремистской организацией и запрещена в России, деятельность принадлежащих ей соц.сетей Facebook и Instagram также запрещена в РФ.

**💡 Совет**

В зависимости от используемой операционной системы есть своя специфика по установке необходимого окружения. Для удобства воспользуйтесь экспресс-инструкциями для *отечественных* (страница 22) и *зарубежных* (страница 43) ОС.

**📌 Важно**

Перед развертыванием ознакомьтесь с *архитектурой развертывания* (страница 1) Blitz Identity Provider.

**1.3.1 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<sup>5</sup>;
- Axiom JDK 11 Certified<sup>6</sup> Рекомендуется для сертифицированной версии;
- Oracle JDK 11<sup>7</sup>.

**1.3.2 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
```

<sup>5</sup> <https://docs.bell-sw.com/liberica-jdk/11.0.23b12/general/install-guide/>

<sup>6</sup> <https://axiomjdk.ru/pages/axiomjdk-install-guide-11.0.17/>

<sup>7</sup> <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.

## 1.3.3 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<sup>10</sup>. Дистрибутив Couchbase Server можно скачать здесь<sup>11</sup>.

### ⓘ 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<sup>12</sup> (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. Скрипты 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:

<sup>10</sup> <https://docs.couchbase.com/server/current/install/install-linux.html>

<sup>11</sup> <https://www.couchbase.com/downloads>

<sup>12</sup> <https://docs.couchbase.com/server/current/manage/manage-nodes/initialize-node.html>

```
./cb_init.sh
```

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<sup>13</sup>.

## Установка и настройка PostgreSQL

### Внимание

Версия PostgreSQL должна быть 9.6 или новее.

### CentOS и RHEL

Установить PostgreSQL согласно инструкции<sup>14</sup>.

### 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:

<sup>13</sup> <https://docs.couchbase.com/server/current/manage-backup-and-restore/manage-backup-and-restore.html>

<sup>14</sup> <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<sup>16</sup>.

### 1.3.4 Шаг 4. RabbitMQ

Опционально

Установка сервера очередей RabbitMQ опциональна и требуется, если сервер очередей будет использоваться для *передачи событий в смежные системы* (страница 317) или в качестве *брокера сообщений* (страница 319).

<sup>16</sup> <https://postgrespro.ru/docs/postgresql/9.6/backup-dump#backup-dump-all>

## CentOS и RHEL

Установить RabbitMQ согласно инструкции<sup>17</sup>.

### Astra Linux Special Edition 1.6:

1. Выполнить команду:

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

2. После завершения установки запустить сервис:

```
systemctl start rabbitmq-server
```

## 1.3.5 Шаг 5. Blitz Identity Provider

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

### 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 8) и *memcached* (страница 8).

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`.
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`.

Листинг 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"
```

Листинг 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:

<sup>17</sup> <https://www.rabbitmq.com/install-rpm.html>

### 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`).

Листинг 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`).

Листинг 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
You 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
```

### 1.3.6 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.
↳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
```

(продолжается на следующей странице)



(продолжение с предыдущей страницы)

```

→$# $%
/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 incron
systemctl start incron
```

## 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`.

### 1.3.7 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 завершения 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 перенаправления использования страниц ошибок из поставки `Blitz Identity Provider`.

### ⚠ Внимание

Статический контент, в отличие от `nginx`, `HAProxy` не поддерживается. 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`.

1. Блок `global` - определяет системные параметры (пользователь, от имени которого выполняется процесс, параметры логирования, разрешенные алгоритмы шифрования и т.д.).
2. Секция `defaults` - определяет общие параметры для следующих за ним секций.
3. Секция `frontend` - определяет правила обработки входящих запросов.
4. Секция `backend` - описывает серверы, 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. Открыть новое соединение:

```
http-check connect
```

4. Отправить `get`-запрос к `лк`:

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

5. Проверить установку сессионной `cookie` `blc`:

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

6. Проверить наличие редиректа:

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

7. 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.
8. Контролировать поток ошибок на уровне I7. Если возникнет более 10-ти ошибок, тогда пометить как недоступный. При успешном health-check будет отмечен, как доступный.
9. Постепенно ввести в балансировку 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
```

### 1.3.8 Шаг 8. LDAP-каталог

Опционально

↪ См. также

*Список поддерживаемых каталогов (страница 1).*

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
```

## 1.4 Экспресс-инструкции для разных ОС

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

### 1.4.1 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
```

### 1.4.2 Astra, Альт, ОСнова, РЕД ОС, РОСА

#### Важно

См. *ограничения* (страница 22) 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;
- РЕД ОС: РЕД ОС 7.3, РЕД ОС 8;
- РОСА: РОСА Хром 12.

#### Шаг 1. JDK

##### Astra 1.7

Установить дистрибутив Liberica JDK 11<sup>19</sup>. Также возможно использовать Axiom JDK 11 Certified<sup>20</sup>.

```
apt-get install openjdk-11-jdk
```

<sup>19</sup> <https://docs.bell-sw.com/liberica-jdk/11.0.23b12/general/install-guide/>

<sup>20</sup> <https://axiomjdk.ru/pages/axiomjdk-install-guide-11.0.5/>

**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<sup>21</sup>. Также возможно использовать Axiom JDK 11 Certified<sup>22</sup>.

```
apt-get install openjdk-11-jdk
```

**РЕД ОС**

Установить дистрибутив:

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

**РОСА**

Установить дистрибутив:

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

**Шаг 2. Memcached****Astra 1.7**

Установить дистрибутив:

```
apt install memcached
```

Запустить службу:

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

**Альт 8**

Установить дистрибутив:

```
apt-get install memcached
```

Запустить службу:

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

**Альт 10**

Установить дистрибутив:

<sup>21</sup> <https://docs.bell-sw.com/liberica-jdk/11.0.23b12/general/install-guide/>

<sup>22</sup> <https://axiomjdk.ru/pages/axiomjdk-install-guide-11.0.5/>

```
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
```

### РЕД ОС

Установить дистрибутив:

```
dnf install memcached
```

Запустить службу:

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

### РОСА

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
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/024-5.27.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/025-5.28.0.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
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/024-5.27.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/025-5.28.0.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
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/024-5.27.0.sql
psql -d blitzdb -U blitz -f ~/tmp/blitz/postgres/ddl/025-5.28.0.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
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/024-5.27.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/025-5.28.0.sql
```

**РЕД ОС**

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
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
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
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/024-5.27.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/025-5.28.0.sql
```

## ПОСА

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
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

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
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/024-5.27.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f /tmp/blitz/postgres/ddl/025-5.28.0.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 ".*" ".*" ".*"

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
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"
```



## РЕД ОС

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
```

### РЕД ОС

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
```

**РОСА**

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
```

## РЕД ОС

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_USERNAME=blitz
PG_USERPASSWORD=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
```

(продолжается на следующей странице)

(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`.

## 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_USERNAME=blitz
PG_USERPASSWORD=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

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
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, запустить nginx:

```
systemctl start nginx
```

Добавить сопоставление адреса loorback-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_USERNAME=blitz
PG_USERPASSWORD=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
You 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_USERNAME=blitz
PG_USERPASSWORD=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
You 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 loorback-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>.

## РЕД ОС

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
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
PG_HOSTNAME=127.0.0.1
PG_DB_NAME=blitzdb
PG_USERNAME=blitz
PG_USERPASSWORD=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`.

### 1.4.3 Rocky Linux, AlmaLinux, Oracle Linux и RHEL

#### ⓘ Важно

См. *ограничения* (страница 22) 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.
↪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
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/024-5.27.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/025-5.28.0.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
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/024-5.27.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/025-5.28.0.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

##
## 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
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
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

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
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
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`.

### 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`.

## 1.5 Первые шаги после установки

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

### 1.5.1 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 328);
- `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.clientAttachingTtlInSec` – 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.ldap.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 307).
- `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<sup>23</sup>:
  - `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"
```

<sup>23</sup> <https://docs.couchbase.com/server/current/learn/data/durability.html>

## 1.5.2 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:

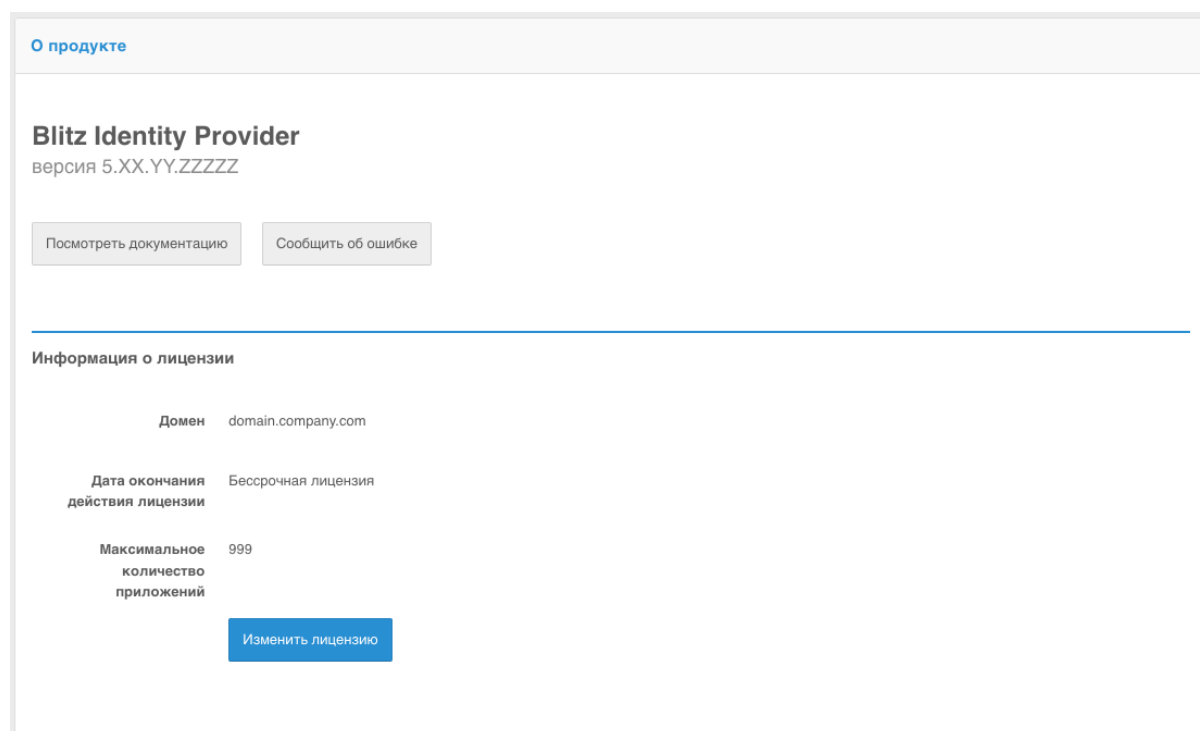
После успешного входа откроется главная страница консоли управления, вид которой приведен ниже. Навигация между различными настройками Blitz Identity Provider осуществляется с помощью меню, расположенного в левой части экрана.

### 1.5.3 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 необходим перезапуск приложений 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"  
}
```

### 1.5.4 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.



Администраторы

Логин	Роли	Пароль	
admin	* суперпользователь	Изменить пароль	✕
support	* администратор ТП	Изменить пароль	✕
security	* администратор ИБ	Изменить пароль	✕
sysadmin	* системный администратор	Изменить пароль	✕

+ Создать учетную запись администратора

Сохранить

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 ( <code>root</code> )	Доступно все
администратор ИБ ( <code>security</code> )	Администраторы, События
системный администратор ( <code>sysadmin</code> )	Источники данных, Аутентификация, Процедуры входа, Поставщики идентификации, SAML, OAuth 2.0, Устройства, Сообщения
администратор приложений ( <code>app_admin</code> )	Приложения
Администратор интерфейса ( <code>ui_admin</code> )	Сервисы самообслуживания, Внешний вид
администратор ТП ( <code>support</code> )	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`.

#### 1.5.5 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
```

### 1.5.6 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.*
```

## Глава 2

# Базовое конфигурирование

### 2.1 Атрибуты учетных записей

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

#### 2.1.1 Что представляет собой атрибут учетной записи

Учетная запись пользователя описывается набором атрибутов.

Значения атрибутов формируются следующими способами:

- считываются из *подключенных хранилищ атрибутов* (страница 65);
- 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.

## 2.1.2 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 65) 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 188) 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 294) 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 64)). 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`.

## Хранимые атрибуты

Определите атрибуты учетной записи пользователя. Для этого задайте *название* – уникальное имя атрибута в системе. Название атрибута может отличаться от его имени во внешнем хранилище, в таком случае укажите правило преобразования в настройках этого хранилища.

Также выберите *тип значения* – тип данных атрибута.

Укажите, какие атрибуты являются:

- *поисковыми (Поиск)* - эти атрибуты будут учтены при поиске учетной записи в разделе «Пользователи», при использовании внешнего хранилища по этим атрибутам следует предусмотреть индекс;
- *обязательными (Обяз.)* - эти атрибуты должны быть заданы при регистрации пользователя и не могут быть удалены в дальнейшем.
- *уникальными (Уник.)* - значения этих атрибутов должны быть уникальны в системе.

Наименование атрибута	Тип значения	Поиск	Обяз.	Уник.	
sub	String	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
family_name	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
given_name	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
middle_name	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
email	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
phone_number	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[+ Добавить атрибут](#)

Предусмотрена возможность назначить для атрибута LDAP-каталога транслятор, осуществляющий преобразование атрибута из хранимого в LDAP формате 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 308) 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<sup>24</sup>.

#### Вычисляемые атрибуты

При необходимости определите вычисляемые атрибуты – укажите их *наименование*, *тип значения*, а также настройте *правило вычисления* на основе хранимых атрибутов.

Вычисляемому атрибуту может быть присвоено константное значение.

[Примеры настройки](#)

Наименование атрибута	Тип значения	Правило вычисления	
preferred_username	String	<code>\${family_name} \${given_name}</code>	
adGroup	Array of strings	<code>\${memberOf}</code>	
onlyCN	Array of strings	<code>\${memberOf}</code>	
domain	String	<code>\${email##*@}</code>	

[+ Добавить атрибут](#)

### 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-}`.

<sup>24</sup> <http://tdp.org/LDP/abs/html/parameter-substitution.html>

## Правила преобразования входных значений

Эти правила позволяют проверять корректность формата ввода данных и обеспечивают сохранение данных в корректном формате. Правила задаются с помощью регулярных выражений.

[Примеры настройки](#)

Наименование атрибута	Декомпозиция	Компоновка	
email	<input type="text" value="^(.+)?@(.+).{1,6}\$"/>	<input type="text" value="\$0"/>	<input type="button" value="✘"/>
phone_number	<input type="text" value="^\+?[789]?[0-9]{3,12}\$"/>	<input type="text" value="+7(\$3)\$4-\$5-\$6"/>	<input type="button" value="✘"/>

[+ Добавить правило](#)

## 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 61) соответствующие вычисляемые атрибуты).

## Правила преобразования выходных значений

Эти правила позволяют совершить дополнительные преобразования с вычисляемыми атрибутами.

Наименование атрибута	Декомпозиция	Компоновка	
onlyCN	<input type="text" value="^cn=(.*)\$"/>	<input type="text" value="\$1"/>	<input type="button" value="✘"/>
adGroup	<input type="text" value="^cn=39SU-ABCD-(?!(TEST-IDEV-))(.*)\$"/>	<input type="text" value="\$0"/>	<input type="button" value="✘"/>

[+ Добавить правило](#)

## Настройка назначения атрибутов

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

**i** 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 65) 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).

#### Назначение атрибутов

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

Также можно указать, какие атрибуты используются:

- для определения заблокированных учетных записей. Этот атрибут должен быть булевым (Boolean);
- в качестве адреса электронной почты;
- в качестве номера мобильного телефона

Можно также указать правило, по которому будет формироваться имя пользователя для отображения в консоли

Идентификатор	<input type="text" value="sub"/>
Признак блокировки	<input type="text" value="locked"/>
Имя пользователя в консоли	<input type="text" value="\${family_name} \${given_name-} \${middle_name-}. \${email-}"/>
Электронная почта	<input type="text" value="x email"/>
Мобильный телефон	<input type="text" value="x phone_number"/>

Сохранить

### 2.1.3 Подключение хранилищ атрибутов

#### Типы хранилищ

В качестве хранилищ атрибутов пользователей Blitz Identity Provider позволяет использовать:

1. Внешнее (подключенное) хранилище. В качестве такого 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 69).
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.

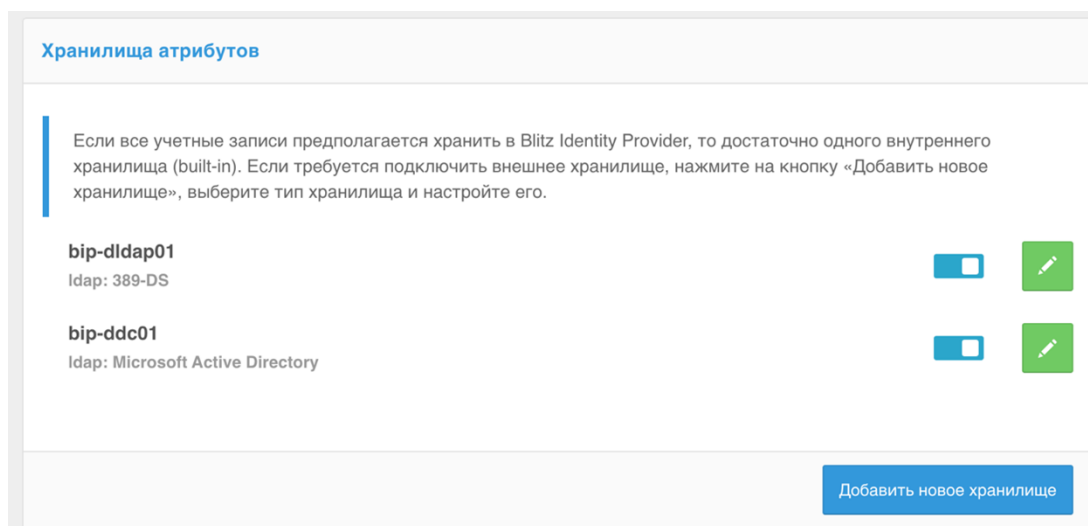
Для корректной работы Blitz Identity Provider требуется настройка хотя бы одного хранилища и конфигурирование *атрибутов* (страница 60). По умолчанию настроено внутреннее хранилище и добавлен ряд атрибутов.

#### 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.).

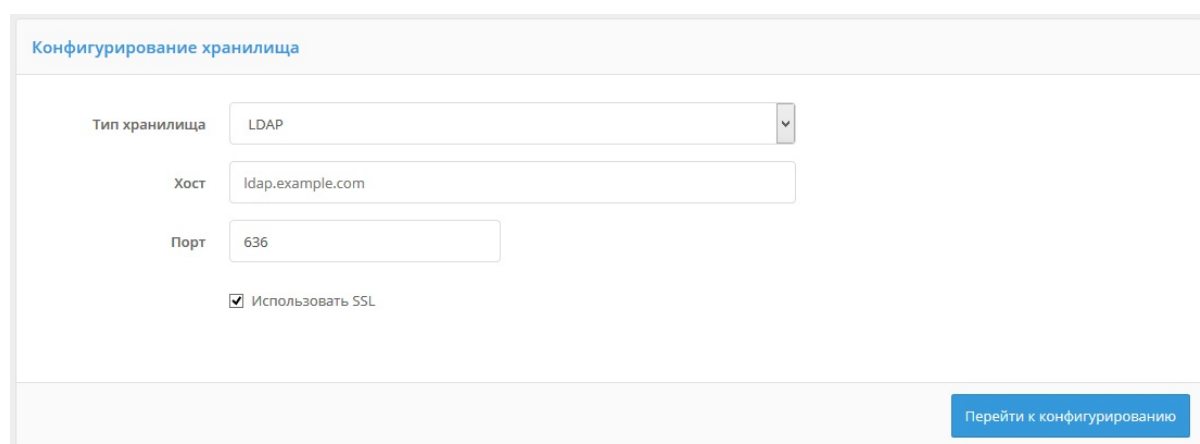
Выбор и настройка используемого хранилища осуществляется после настройки атрибутов в разделе Источники данных в разделе Хранилища атрибутов. По умолчанию настроено внутреннее хранилище. Для добавления внешнего хранилища следует нажать на кнопку Добавить новое хранилище, после чего указать тип внешнего 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);
  - использует 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;

**i** **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;

**i** **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).

### Параметры подключения к LDAP хранилищу

Идентификатор

Описание

Только для чтения  Нет v

---

#### Настройка соединения

Без балансировки
DNS-балансировка

Хост

Порт

Использовать SSL

---

#### Настройка пула соединений

Таймаут соединения, мс	<input type="text" value="3000"/>	Начальное количество соединений	<input type="text" value="10"/>
Таймаут ответа, мс	<input type="text" value="3000"/>	Максимальное количество соединений	<input type="text" value="10"/>

---

#### Учетная запись для работы с хранилищем

Для корректной работы должна быть указана учетная запись с правами на чтение данных из хранилища. Если планируется изменение/добавление данных средствами Blitz Identity Provider, то необходимы права на запись

Пользователь(DN)

Пароль [Изменить значение](#)

Базовый DN

---

#### Настройки поиска

Глубина поиска  v

Максимальное количество записей, возвращаемых при поиске

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-}`.

## Правила сопоставления атрибутов

Настройте правила сопоставления, если названия или форматы атрибутов в Blitz Identity Provider не совпадают с тем, как эти атрибуты определены в LDAP-каталоге. И для чтения, и для записи можно 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 `surname`, то выберите атрибут `surname` и укажите `sn` в качестве его названия в LDAP;
- 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-}`;

Атрибут	Название в LDAP	Запись		Чтение		
		Правило разбиения	Правило преобразования	Правило разбиения	Правило преобразования	
sub	uid					✖
family_name	sn					✖
given_name	givenName					✖
middle_name	middleName					✖
email	mail					✖
phone_number	mobile					✖

+ Добавить атрибут

Если хранение *созданного ранее* (страница 60) 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`.

### Параметры создания новых пользователей

Для корректной работы создания пользователя необходимо указать специфичные для LDAP хранилища параметры. При формировании значений параметров можно использовать строки подстановки из атрибутов пользователя. Списочное значение можно задать через запятую.

DN пользователей

Например, CN=\${mail},CN=users,DC=domain,DC=com

Первоначальные атрибуты Например: objectclass.	Название	Формат	Значение	
	objectClass	Array of strings	top,blitz-schema	✖

+ Добавить атрибут

## 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;
  - 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.

### Параметры REST-сервисов

Идентификатор	<input type="text" value="rest_test"/>
Описание	<input style="height: 20px;" type="text"/>
Только для чтения	<input type="button" value="Да"/>
Максимальное количество записей, возвращаемых при поиске	<input type="text" value="100"/>
Перечень доступных атрибутов	<input type="checkbox"/> sub <input type="checkbox"/> family_name <input type="checkbox"/> given_name <input type="checkbox"/> middle_name <input type="checkbox"/> email <input type="checkbox"/> phone_number

Атрибуты пользователя, которые доступны в запросах к REST-сервисам

---

### Адреса REST-сервисов

URL сервиса поиска пользователей	<input style="width: 90%;" type="text"/> <small>HTTP метод запроса: GET. Параметр запроса:</small> <ul style="list-style-type: none"> <li>• <code>rq1</code> — запрос в формате <b>Resource Query Language (RQL)</b>.</li> </ul> <small>Формат ответа: 200 OK, список пользователей в формате <b>JSON Array</b> в кодировке <b>UTF-8</b>.</small> <small><a href="#">Пример листинга</a></small>
URL сервиса получения данных пользователя	<input style="width: 90%;" type="text"/> <small>При указании URL необходимо использовать строку подстановки для идентификатора пользователя - <code>{id}</code>.</small> <small>HTTP метод запроса: GET.</small> <small>Формат ответа: 200 OK, данные пользователя в формате <b>JSON</b> в кодировке <b>UTF-8</b>.</small> <small>Если пользователь не найден: 400 Bad Request, код ошибки <b>USER_NOT_FOUND</b> в формате <code>text/plain; charset=utf-8</code>.</small> <small><a href="#">Пример листинга</a></small>

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)<sup>25</sup> 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:

```
[
  {
    "id": "ID123",
```

(продолжается на следующей странице)

<sup>25</sup> <https://github.com/kriszyp/rql>

(продолжение с предыдущей страницы)

```

"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"
  }
}

```

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

{
  "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.

Параметры встроенного хранилища

Идентификатор	<input type="text" value="built-in"/>
Описание	<input type="text" value="Built-in store"/>
Только для чтения	<input type="checkbox"/> Нет
Максимальное количество записей, возвращаемых при поиске	<input type="text" value="100"/>

[Удалить](#) [Сохранить](#)

### **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 Аутентификация**

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

### **2.2.1 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

TOTP-генераторы

Настройки TOTP-генераторов

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

Имперсонафикация

Настройки выбора вспомогательных учетных записей

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).

#### Настройки аутентификации

Общие настройки   Парольные политики   Ключи безопасности   TOTP-генераторы   **Первый фактор**   Второй фактор   Имперсонификация

[Добавить внешний метод аутентификации](#)

<p>Автоматическая идентификация <span style="float: right;"><input type="checkbox"/></span></p> <p>Для идентификации пользователя используются свойства сессии, которые предварительно вычисляются и сохраняются при старте процедуры входа.</p> <p><a href="#">Перейти к настройкам</a></p>	<p>Вход по сеансу операционной системы <span style="float: right;"><input checked="" type="checkbox"/></span></p> <p>При входе будет использоваться текущий сеанс операционной системы</p> <p><a href="#">Перейти к настройкам</a></p>
<p>Логин и пароль <span style="float: right;"><input checked="" type="checkbox"/></span></p> <p>При входе в систему пользователю необходимо ввести логин и пароль</p> <p><a href="#">Перейти к настройкам</a></p>	<p>Разовый пароль на основе времени (TOTP) <span style="float: right;"><input type="checkbox"/></span></p> <p>Задайте настройки привязки новых TOTP-генераторов к учетной записи. Параметры использования TOTP-генераторов при входе задаются в настройках соответствующих методов аутентификации</p> <p><a href="#">Перейти к настройкам</a></p>
<p>Средство электронной подписи <span style="float: right;"><input checked="" type="checkbox"/></span></p> <p>При входе в систему пользователю необходимо использовать средство электронной подписи или смарт-карту</p> <p><a href="#">Перейти к настройкам</a></p>	<p>Прокси-аутентификация <span style="float: right;"><input type="checkbox"/></span></p> <p>Для аутентификации используются HTTP-заголовки прокси-сервера. В частности, в таком заголовке может быть указан сертификат, полученный в результате установленного двустороннего SSL/TLS соединения</p> <p style="text-align: center; background-color: #fff9c4; padding: 5px;"><a href="#">Сконфигурируйте метод для использования</a></p>
<p>Вход через внешние сервисы идентификации <span style="float: right;"><input checked="" type="checkbox"/></span></p> <p>Для входа пользователь будет перенаправлен на внешний сервис идентификации. Пользователю потребуются дать согласие на передачу данных своей учетной записи в Blitz Identity Provider.</p> <p><a href="#">Перейти к настройкам</a></p>	<p>Вход по временной ссылке <span style="float: right;"><input checked="" type="checkbox"/></span></p> <p>Вход в систему осуществляется по ссылке. Ссылка действует в течение ограниченного времени.</p> <p><a href="#">Перейти к настройкам</a></p>
<p>Вход с известного устройства <span style="float: right;"><input checked="" type="checkbox"/></span></p> <p>После успешного входа устройство запоминается. В течение определенного периода вход в систему осуществляется автоматически</p> <p><a href="#">Перейти к настройкам</a></p>	<p>Вход по коду из SMS/push <span style="float: right;"><input checked="" type="checkbox"/></span></p> <p>Для первичного входа пользователю нужно ввести код из push-уведомления или SMS, отправленного на номер мобильного телефона</p> <p><a href="#">Перейти к настройкам</a></p>
<p>Вход по QR-коду <span style="float: right;"><input checked="" type="checkbox"/></span></p> <p>Для входа в систему пользователь должен считать QR-код в мобильном приложении</p> <p><a href="#">Перейти к настройкам</a></p>	<p>Вход с помощью ключа безопасности <span style="float: right;"><input checked="" type="checkbox"/></span></p> <p>Аутентификация осуществляется с помощью ключей безопасности WebAuthn или U2F</p> <p><a href="#">Перейти к настройкам</a></p>

**Настройки аутентификации**

Общие настройки Парольные политики Ключи безопасности TOTP-генераторы Первый фактор **Второй фактор** Третий фактор Имперсонификация

[Добавить внешний метод аутентификации](#)

<p>Разовый пароль на основе секрета (НОТР) <input checked="" type="checkbox"/></p> <p>После успешного первичного входа пользователю нужно ввести код, сгенерированный специальным устройством - генератором одноразовых паролей</p> <p><a href="#">Перейти к настройкам</a></p>	<p>Разовый пароль на основе времени (TOTP) <input checked="" type="checkbox"/></p> <p>Задайте настройки привязки новых TOTP-генераторов к учетной записи. Параметры использования TOTP-генераторов при входе задаются в настройках соответствующих методов аутентификации</p> <p><a href="#">Перейти к настройкам</a></p>
<p>Вход с известного устройства <input checked="" type="checkbox"/></p> <p>Позволяет не требовать усиленную аутентификацию (второй фактор) при входе с известного устройства.</p>	<p>Подтверждение с помощью электронной почты <input checked="" type="checkbox"/></p> <p>После успешного первичного входа пользователю нужно ввести код из сообщения, переданного на адрес электронной почты</p> <p><a href="#">Перейти к настройкам</a></p>
<p>Внешний метод "mbs" <input type="checkbox"/></p> <p>После успешного первичного входа будет произведена дополнительная проверка с помощью внешнего сервиса аутентификации</p> <p><a href="#">Перейти к настройкам</a></p>	<p>Подтверждение по коду из SMS/push <input checked="" type="checkbox"/></p> <p>После успешного первичного входа пользователю нужно ввести код из сообщения, переданного на номер мобильного телефона</p> <p><a href="#">Перейти к настройкам</a></p>
<p>Подтверждение с помощью ключа безопасности <input checked="" type="checkbox"/></p> <p>Аутентификация осуществляется с помощью ключей безопасности WebAuthn или U2F</p> <p><a href="#">Перейти к настройкам</a></p>	<p>Подтверждение ответом на контрольный вопрос <input checked="" type="checkbox"/></p> <p>После успешного первичного входа пользователю нужно ввести ответ на контрольный вопрос</p> <p><a href="#">Перейти к настройкам</a></p>
<p>Подтверждение по входящему звонку <input checked="" type="checkbox"/></p> <p>После успешного первичного входа пользователю нужно ввести код, образованный последними цифрами номера входящего звонка</p> <p><a href="#">Перейти к настройкам</a></p>	<p>Подтверждение с помощью пароля <input checked="" type="checkbox"/></p> <p>При входе в систему пользователю необходимо ввести пароль</p> <p><a href="#">Перейти к настройкам</a></p>

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

## 2.2.2 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 ис-

течет, даже при активной SSO-сессии. В этом случае необходимо *внести изменения* (страница 359) в файл конфигурации.

- Время отображения экрана логута, сек.: время в секундах, в течение которого пользователю будет показываться экран выхода до автоматического перенаправления пользователя на страницу перехода в приложение после логута.
- 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.

### Настройки аутентификации

- Общие настройки
- Парольные политики
- Ключи безопасности
- TOTP-генераторы
- Первый фактор
- Второй фактор
- Имперсонификация

**Уровень аутентификации по умолчанию**

Укажите требование к аутентификации пользователей по умолчанию. Если указан вариант "первый и второй фактор", то по умолчанию все пользователи должны пройти двухфакторную аутентификацию.

**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

**Время отображения экрана логгута, сек.**

---

**Запоминание учетных записей**

Включено

**Запоминание учетных записей**

**Отображаемое имя пользователя**

**Отображаемый идентификатор пользователя**

Отображать аватар

Сохранить

### 2.2.3 Парольные политики

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



**Настройки аутентификации**

Общие настройки **Парольные политики** Ключи безопасности TOTP-генераторы Первый фактор Второй фактор Имперсонификация

---

**Сложность пароля**

Минимальная длина пароля   
Укажите минимальное количество символов в пароле

Словарь паролей  Обзор...  
Выберите файл со словарем паролей, где каждый пароль размещен на новой строке. Формат файла должен быть txt.

Группы символов   
Задайте минимальное количество групп символов, необходимых в пароле

Название группы	Допустимые символы	Минимум символов
Цифры	<input type="text" value="[0-9]"/>	<input type="text" value="1"/>
Нижний регистр	<input '"="" \]&gt;&lt;="~/\_" ]"="" type="text" value="[a-z!@#%&amp;'()*+,-?,:;"/>	<input type="text" value="1"/>
Верхний регистр	<input type="text" value="[A-Z]"/>	<input type="text" value="1"/>
Специальные символы	<input type="text"/>	<input type="text"/>

---

**Политика повторного использования**

Запрет использования старых паролей, шт.

Минимальное время жизни пароля, сек.

Максимальное время жизни пароля, сек.

Минимальное число отличающихся символов, шт.

[Сохранить](#)

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 – `[0-9+-]`, можно расширить допустимые наборы символов букв – `[a-zA-я]` и `[A-ZА-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 ис-

пользованных паролей.

- Минимальное время жизни пароля – минимальное время жизни пароля, в секундах; пока это время 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.

## 2.2.4 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<sup>26</sup>.

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.

---

<sup>26</sup> <https://fidoalliance.org/fido2/>

**Настройки аутентификации**

Общие настройки Парольные политики **Ключи безопасности** TOTP-генераторы Первый фактор Второй фактор Имперсонификация

**Имя системы аутентификации**  Отображаемое пользователю имя системы аутентификации.

**Домен системы аутентификации**  Идентификатор системы аутентификации. Должен совпадать с доменом системы аутентификации или вышестоящим доменом.

**Алгоритмы подписи**   Используемые при аутентификации алгоритмы подписи

**Ограничение разрешенных средств аутентификации**  Если настройка задана, то используются только средства аутентификации указанного в настройке типа.

**Режим проверки наличия ключей**  Список доступных ключей безопасности определяется сервером на основе введенного пользователем логина или браузер самостоятельно запрашивает у пользователя выбор ключа из всех доступных.

**Время ожидания, мс**  Указывается время в мс, в течение которого сервер аутентификации будет ждать обработки запроса браузером.

**Отображаемое имя пользователя**  Отображается пользователю на странице входа при входе с помощью ключа безопасности. Используйте строки подстановки для формирования имени. Например, "{family\_name-} {given\_name-}"

**Отображаемый идентификатор учетной записи**  Отображается пользователю на устройстве, используемом для входа с помощью ключа безопасности. Используйте строки подстановки для формирования идентификатора. Например, "{email-}"

**Нормальный сдвиг счетчика аутентификаций**  При аутентификации сервер проверяет, что переданный счетчик подписей соответствует текущему счетчику на сервере с допустимым расхождением. Рекомендуется задавать значение 1.

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 341) 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<sup>27</sup>) 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 87), при выполнении которых поиск пользователя будет осуществляться в определенном хранилище.

<sup>27</sup> <https://fidoalliance.org/fido2/>

### Аутентификация по ключам безопасности

**Разрешенные режимы аттестации**

× FULL
× FULL\_NO\_ROOT
× SELF

Режим аттестации проверяется при регистрации ключа безопасности пользователем. FULL - при проверке проверяется наличие root сертификата в доверенном хранилище. FULL\_NO\_ROOT - наличие root сертификата не обязательно. SELF - позволяет принимать самоподписанное аттестационное утверждение. По умолчанию разрешен только режим FULL.

Показывать метод только пользователям, которые привязали к учетной записи ключ безопасности. По умолчанию метод показывается всем пользователям.

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

Для корректной работы входа по ключам безопасности укажите, каким образом должен формироваться логин и какому атрибуту в источнике данных он соответствует. Вы можете создать несколько альтернативных правил определения логина.

Для создания правила используйте **строки подстановки**. Например, правило `CN-$(login)` означает, что строка, введенная пользователем, будет сравниваться с атрибутом `CN` в хранилище данных.

[Посмотреть строки подстановки](#)

sub

=

\$(login)

×

+ добавить условие

OR

email

=

\$(login)

×

+ добавить условие

+ добавить альтернативное правило

Отменить
Сохранить

---

#### Правила выбора хранилища атрибутов

По умолчанию поиск пользователей для аутентификации происходит во всех активных хранилищах. В данном блоке 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 **строки подстановки**.

[Посмотреть строки подстановки](#)

[Создать правило](#)

## Подтверждение входа с помощью WebAuthn, Passkey, FIDO2, U2F

Существует возможность использовать ключи безопасности (WebAuthn, Passkey, FIDO2<sup>28</sup>, 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.
- Показывать метод только пользователям, которые привязали к учетной записи ключ безопасности –

<sup>28</sup> <https://fidoalliance.org/fido2/>

если ключи безопасности не настроены, то можно настроить, чтобы пользователю метод подтверждения входа с помощью ключа безопасности не показывался.

**Аутентификация по ключам безопасности**

Разрешенные режимы аттестации:  FULL  FULL\_NO\_ROOT  SELF

Режим аттестации проверяется при регистрации ключа безопасности пользователем. FULL - при проверке проверяется наличие root сертификата в доверенном хранилище. FULL\_NO\_ROOT - наличие root сертификата не обязательно. SELF - позволяет принимать самоподписанное аттестационное утверждение. По умолчанию разрешен только режим FULL.

Показывать метод только пользователям, которые привязали к учетной записи ключ безопасности. По умолчанию метод показывается всем пользователям.

## 2.2.5 TOTP-генераторы

TOTP-генераторы настраиваются на вкладке TOTP-генераторы раздела Аутентификация консоли управления. Также TOTP-генераторы настраиваются на первом и втором факторах на соответствующих вкладках. Для получения дополнительной информации см. *Вход разовым паролем на основе времени (TOTP)* (страница 112) и *Подтверждение входа разовым паролем на основе времени (TOTP)* (страница 119).

**Настройки аутентификации**

Общие настройки | Парольные политики | Ключи безопасности | **TOTP-генераторы** | Первый фактор | Второй фактор | Имперсонификация

---

**Параметры проверки разовых кодов**

Допустимое отклонение (вперед)   
Количество последующих по времени кодов, которые могут быть введены

Допустимое отклонение (назад)   
Количество предыдущих по времени кодов, которые могут быть введены

---

**Настройка отображения генераторов разовых паролей**

Атрибут с именем пользователя   
Имя пользователя будет отображаться в генераторе разовых паролей после привязки

Название единой системы входа   
Название системы будет отображаться в генераторе разовых паролей после привязки

---

**Ссылки на приложения - генераторы разовых паролей**

Укажите для каждой ОС, какие мобильные приложения рекомендуется использовать для генерации разовых паролей. Если ссылка не указана, то пользователям не будет предложено загрузить приложение для данной ОС.

IOS

Android

Предусмотрены следующие настройки:

### 1. Параметры проверки разовых кодов

- Допустимое отклонение (вперед) / Допустимое отклонение (назад) - 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. Настройка отображения генераторов разовых паролей

- Атрибут с именем пользователя - имя пользователя будет отображаться в генераторе разовых паролей после привязки учетной записи пользователя;
- Название единой системы входа - название системы будет отображаться в генераторе разовых паролей после привязки учетной записи пользователя.

## 3. Ссылки на приложения - генераторы разовых паролей

- Следует указать для каждой ОС, какие мобильные приложения рекомендуется использовать для генерации разовых паролей. Эти ссылки будут предложены пользователю в веб-приложении «Личный кабинет». Если ссылка не указана, то пользователям не будет предложено загрузить приложение для данной ОС.

## 2.2.6 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;

**Вход по логину и паролю**

Для корректной работы входа по паролю укажите, каким образом должен формироваться логин и какому атрибуту в источнике данных он соответствует. Вы можете создать несколько альтернативных правил определения логина. Ввод логина не чувствителен к регистру.

Для создания правила используйте [строки подстановки](#). Например, правило `cn=${login}` означает, что строка, введенная пользователем, будет сравниваться с атрибутом `cn` в хранилище данных.

[Посмотреть строки подстановки](#)

<input type="text" value="sub"/>	=	<input type="text" value="\${login}"/>	<input type="checkbox"/>	
				<a href="#">+ добавить условие</a>
OR				
<input type="text" value="email"/>	=	<input type="text" value="\${login}"/>	<input type="checkbox"/>	
				<a href="#">+ добавить условие</a>
OR				
<input type="text" value="phone_number"/>	=	<input type="text" value="\${login}"/>	<input type="checkbox"/>	
				<a href="#">+ добавить условие</a>
				<a href="#">+ добавить альтернативное правило</a>

В настройках входа по логину и пароля можно включить проверку на соответствие пароля *парольной политике* (страница 305). 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).

#### Соответствие пароля парольной политике

	<input type="checkbox"/>	Разрешить пользователю пропустить смену пароля, не соответствующего парольным политикам
	<input checked="" type="checkbox"/>	Всегда проверять текущий пароль пользователя на соответствие парольной политике
<b>Проверять при наличии HTTP заголовка</b>	<input type="text"/>	Если HTTP-запрос будет содержать указанный заголовок со значением true, то текущий пароль пользователя будет проверен на соответствие парольной политике
<b>Кол-во неудачных попыток для временной блокировки</b>	<input type="text" value="5"/>	После указанного кол-ва неудачных попыток будет установлена временная блокировка пользователю на использование данного метода аутентификации
<b>Длительность временной блокировки</b>	<input type="text" value="1"/>	Определяет длительность временной блокировки в минутах по истечению которых пользователь снова сможет использовать данный метод аутентификации

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 неудачных аутентификаций превысит «Порог включения системной защиты», и выключится, если доля неудачных аутентификаций станет ниже «Порог выключения системной защиты»;
- Автоматический режим на уровне пользователей – защита сработает в отношении пользователей, по которым будет превышено число неудачных 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 неудачных аутентификаций в общем числе попыток аутентификации.



Пример настройки защиты от подбора пароля представлен ниже.

**Защита от подбора пароля**

При включенной защите происходит замедление процесса аутентификации. В этом случае после ввода пароля пользователь будет ожидать результата в течение периода, определенного настройкой «Время задержки». Предусмотрены следующие автоматические режимы защиты:

- на уровне системы в целом. Включается, если процент неуспешных аутентификаций достигнет определенного порога (настройка «Порог включения системной защиты»);
- на уровне пользователя. Включается, если пользователь вводит подряд определенное количество неверных паролей (настройка «Порог включения пользовательской защиты»).

Защита	<input type="text" value="Отключена"/>
Время задержки, в секундах	<input type="text" value="10"/>
Порог включения пользовательской защиты	<input type="text" value="5"/>
Порог включения системной защиты	<input type="text" value="40"/>
Порог выключения системной защиты	<input type="text" value="30"/>

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.

**Доказательство выполнения работы**

При каждой проверке пароля на стороне браузера будет выполняться достаточно длительная работа. Результат выполнения работы будет проверяться сервером одновременно с проверкой пароля.

Запрашивать доказательство выполнения работы

Запрашивать только при наличии HTTP заголовка

Доказательство выполнения работ будет запрашиваться только при наличии в запросе HTTP заголовка с значением 1

Показатель сложности работы

Коэффициент от 1 до 160 бит. Каждый бит увеличивает сложность в 2 раза.

Максимальный срок решения

Максимальное время в секундах, за которое браузер должен прислать результат работы. Если значение не задано, то решение задачи ожидается за 1800 секунд.

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, заголовков 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`.

Правила выбора хранилища атрибутов

По умолчанию поиск пользователей для аутентификации происходит во всех активных хранилищах. В данном блоке 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 строки подстановки.

[Посмотреть строки подстановки](#)

Хранилище атрибутов	Правило соответствия
main	<input type="checkbox"/> not <input type="text" value="\$_rpId_"/> <input type="text" value="~*"/> <input type="button" value="✕"/>
test_db	<input type="checkbox"/> not <input type="text" value="\$_rpId_"/> <input type="text" value="test_app"/> <input type="button" value="✕"/>

[+ Добавить альтернативное условие](#)

[+ Добавить альтернативное условие](#)

[+ Добавить правило](#)

## 2.2.7 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 309) 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**

Привязать 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

---

**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](#)

---

**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}	<input type="checkbox"/>	✖
sub	= \${SUBJECT.E}	<input type="checkbox"/>	✖

[+ Добавить атрибут](#)

---

**Сертификаты**

Загрузите сертификаты удостоверяющих центров (CA), подтверждающих подлинность ключей электронной подписи пользователей.

Укажите путь к сертификату для загрузки

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 309) настройки использования плагина электронной подписи.

## 2.2.8 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<sup>1</sup>;
- 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 137) 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.

<sup>1</sup> Meta признана экстремистской организацией и запрещена в России, деятельность принадлежащих ей соц.сетей Facebook и Instagram также запрещена в РФ.

Вход через внешние сервисы идентификации

Для добавления и настройки поставщиков идентификации используйте раздел консоли «Поставщики идентификации».

Название поставщика	Уникальное название	Тип поставщика	
Google	google_1	google	<input checked="" type="checkbox"/>
Apple ID	apple_1	apple	<input checked="" type="checkbox"/>
Яндекс	yandex_1	yandex	<input checked="" type="checkbox"/>
Mail ID	mail_1	mail	<input checked="" type="checkbox"/>
Facebook*	facebook_1	facebook	<input checked="" type="checkbox"/>
VK	vk_1	vk	<input checked="" type="checkbox"/>
Одноклассники	ok_1	ok	<input checked="" type="checkbox"/>
ESIA	esia_1	esia	<input checked="" type="checkbox"/>
Сбер ID	sbrf_1	sbrf	<input checked="" type="checkbox"/>
Цифровой профиль ЕСИА	esiadp_1	esiadp	<input checked="" type="checkbox"/>

## 2.2.9 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:

**Прокси-аутентификация**

Чтобы использовать данный метод аутентификации, обязательно должен быть настроен прокси-сервер, передающий в HTTP-заголовках идентификационную информацию пользователя. Метод применяется автоматически, если в HTTP-заголовках получены необходимые для идентификации пользователя данные. Если заголовки не обнаружены, то будут использованы другие методы аутентификации

---

**HTTP-заголовки**

Требуемые HTTP-заголовки

X-SSL-Client-CERT x X-SSL-Client-Serial x X-SSL-Client-S-DN x X-SSL-Client-Email x

Для добавления HTTP-заголовка введите его и нажмите Enter

Укажите названия HTTP-заголовков, которые должны присутствовать для проведения аутентификации пользователя. Если заголовки не указаны, то аутентификация будет возможна при любом наборе заголовков

HTTP-заголовок с сертификатом пользователя

X-SSL-Client-CERT

Заголовок, в котором передается сертификат пользователя. Если указан, то возможна идентификация пользователя по атрибутам сертификата

---

**Правила соответствия**

Для корректной работы прокси-аутентификации укажите, какие HTTP-заголовки соответствуют каким атрибутам в источнике данных. Вы можете создать несколько альтернативных правил.

Для обозначения заголовков используйте строки подстановки. Например, правило `CN=${HTTP_X_SSL_CLIENT_CN}` означает, что заголовок `HTTP_X_SSL_CLIENT_CN` будет сравниваться с атрибутом CN в хранилище данных.

Если настроено считывание сертификата из определенного заголовка, то можно настроить правила соответствия полей сертификата и атрибутов в хранилище данных, используя строки подстановки.

[Посмотреть строки подстановки для X509 сертификата.](#)

email

=

X-SSL-Client-Email

x

[+ добавить условие](#)  
[+ добавить альтернативное правило](#)

Отмена Сохранить

## 2.2.10 Вход с помощью сеанса операционной системы

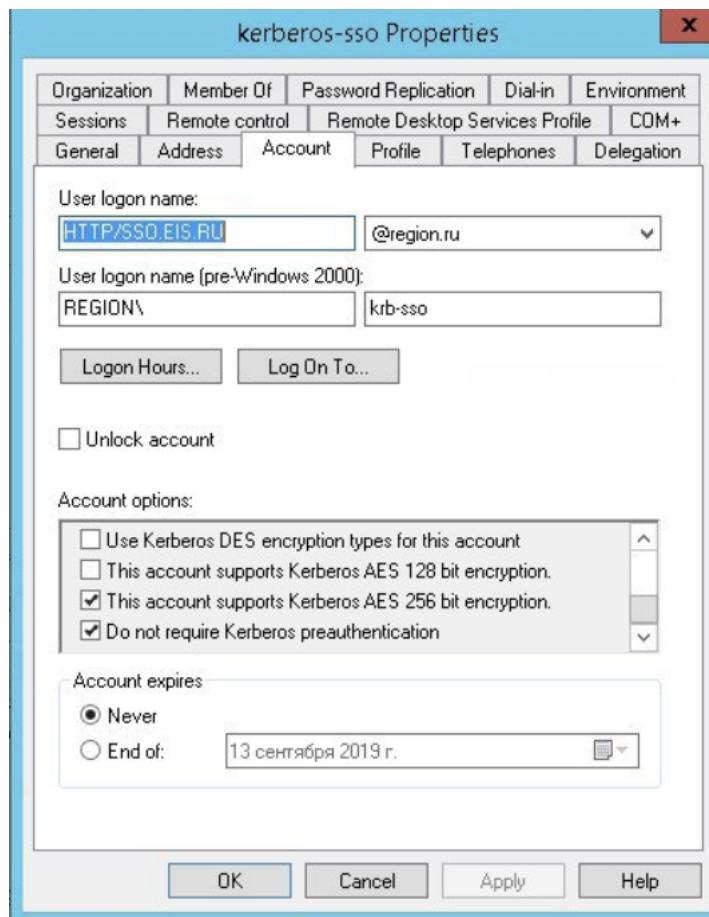
Способ входа с использованием сеанса операционной системы позволяет пользователям не проходить дополнительно идентификацию и аутентификацию в Blitz Identity Provider, если они ранее вошли со своего ПК в сеть организации и прошли идентификацию и аутентификацию в операционной системе (вошли в сетевой домен). Такие пользователи получают 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)

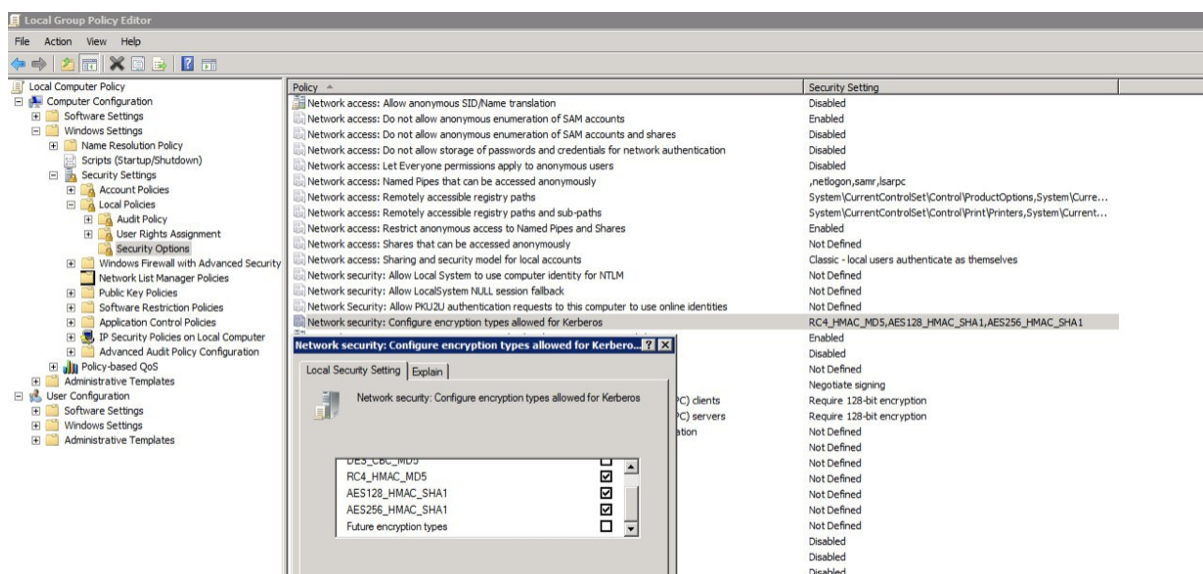
В контролере домена необходимо зарегистрировать учетную запись для сервера Blitz Identity Provider. Для созданной учетной записи нужно на странице Account в блоке Account options оснастки контроллера домена включить настройки User cannot change password и Password never expires.

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.

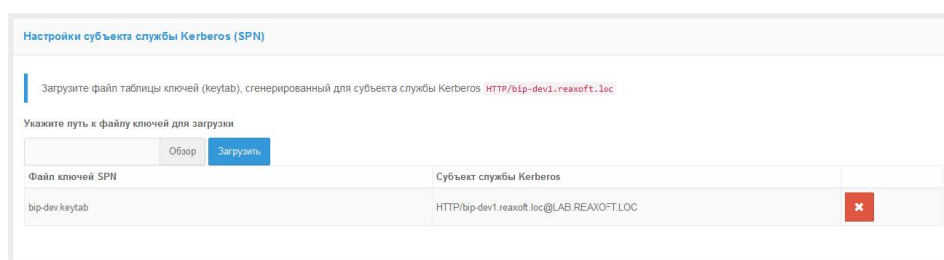
### Настройки в консоли управления Blitz Identity Provider

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.

**Правила соответствия**

Для корректной работы входа по сеансу операционной системы укажите, каким атрибутам в источнике данных соответствуют имя и домен пользователя из текущего сеанса операционной системы. Вы можете создать несколько альтернативных правил.

Для создания правила используйте **строки подстановки**. Например, правило `userPrincipalName=${username}@${domain}` означает, что имя пользователя с доменом из сеанса операционной системы, будет сравниваться с атрибутом `userPrincipalName` в хранилище данных.

[Посмотреть строки подстановки](#)

sAMAccountName = \${username} ✖

[+ добавить условие](#)

[+ добавить альтернативное правило](#)

Отмена Сохранить

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.

**Дополнительные настройки**

Время задержки перед запуском метода   
Количество секунд, в течение которых пользователь может переключиться на другой метод аутентификации

Время ожидания получения токена   
Количество секунд ожидания получения токена. По окончании периода возвращается сообщение об ошибке

Отмена Сохранить

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

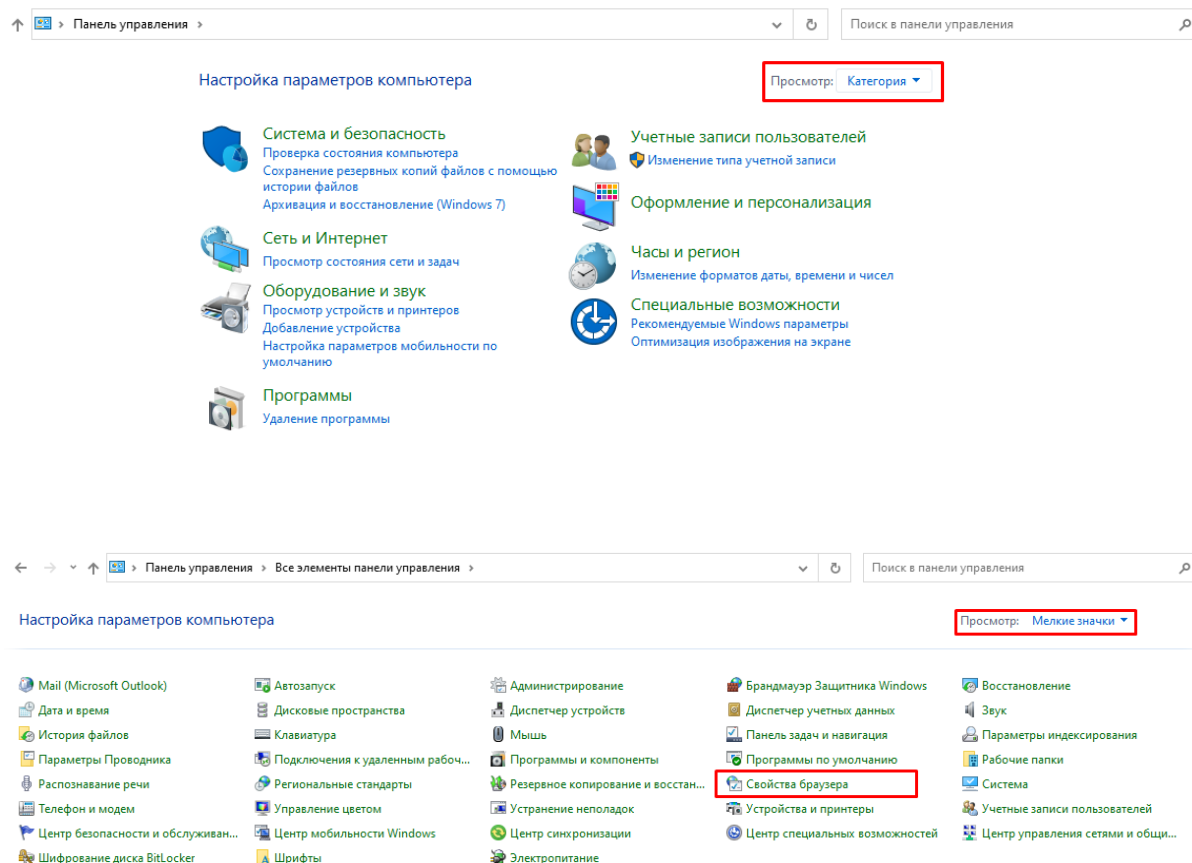
правила, при выполнении которых поиск пользователя будет осуществляться в *определенном хранилище* (страница 87).

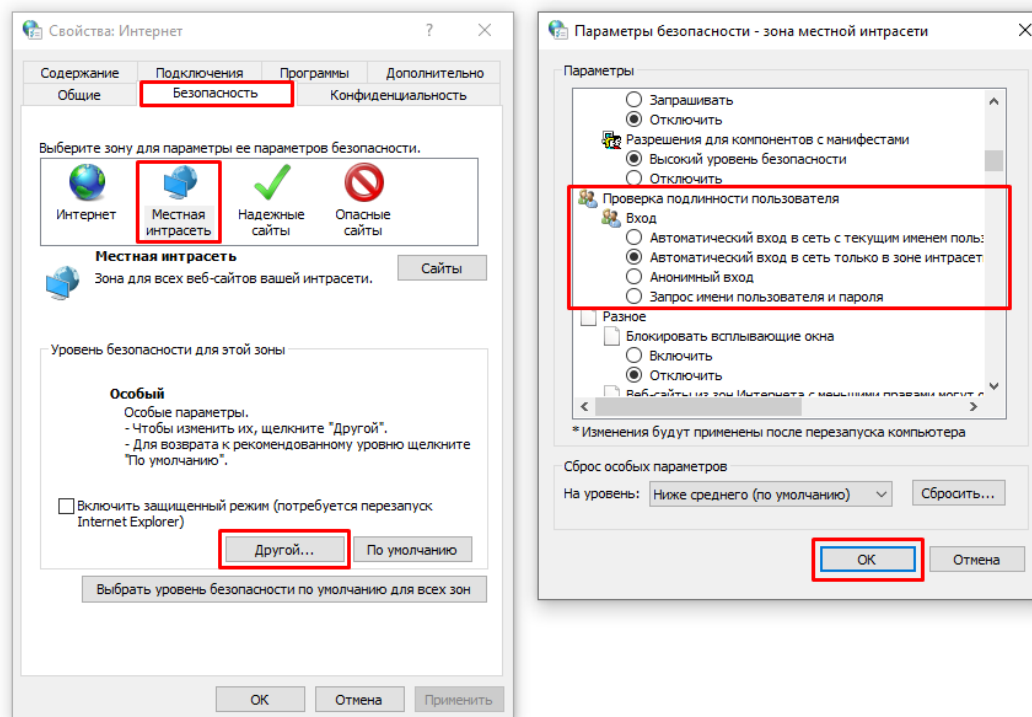
## Настройки браузеров пользователей

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 `OK`.

- 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.

## 2.2.11 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}`.

Для корректной идентификации пользователя укажите, каким образом должен формироваться логин и какому атрибуту в источнике данных он соответствует. Вы можете создать несколько альтернативных правил определения логина. Ввод логина не чувствителен к регистру.

Для создания правила используйте [строки подстановки](#). Например, правило `CN=${login}` означает, что строка, введенная пользователем, будет сравниваться с атрибутом `CN` в хранилище данных.

[Посмотреть строки подстановки](#)

email	=	\${login}	✕	<a href="#">+ добавить условие</a>
OR				
sub	=	\${login}	✕	<a href="#">+ добавить условие</a>
<a href="#">+ добавить альтернативное правило</a>				

#### Параметры кодов подтверждения

<b>Длина</b>	6	Количество символов в коде подтверждения
<b>Время действия</b>	120	Количество секунд, после которого код подтверждения перестает действовать. Необходима отправка нового кода
<b>Количество попыток за 1 вход</b>	3	Количество неудачных попыток ввода кода подтверждения при одной попытке входа. Если количество попыток превышено, требуется отправка нового кода
<b>Общее количество попыток</b>	5	Общее число отправок кодов подтверждения и попыток ввода кода подтверждения, после которого способ аутентификации будет временно заблокирован
<b>Время блокировки при превышении попыток, в мин.</b>	5	В течение указанного времени способ аутентификации будет недоступен пользователю

#### Параметры отправки

<b>Атрибут с контактом</b>	\${email-}	Выражение, по которому будет формироваться адрес электронной почты для отправки кода подтверждения
----------------------------	------------	--

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

#### Правила выбора хранилища атрибутов

По умолчанию поиск пользователей для аутентификации происходит во всех активных хранилищах. В данном блоке 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 [строки подстановки](#).

[Посмотреть строки подстановки](#)

[Создать правило](#)

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 200).

### 2.2.12 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 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}`;



## Вход по коду из SMS/push

Для корректной идентификации пользователя укажите, каким образом должен формироваться логин и какому атрибуту в источнике данных он соответствует. Вы можете создать несколько альтернативных правил определения логина. Ввод логина не чувствителен к регистру.

Для создания правила используйте [строки подстановки](#). Например, правило `CN=${login}` означает, что строка, введенная пользователем, будет сравниваться с атрибутом `CN` в хранилище данных.

[Посмотреть строки подстановки](#)

phone_number	=	\$(login)	
--------------	---	-----------	--

[+ добавить условие](#)

[+ добавить альтернативное правило](#)

## Параметры кодов подтверждения

Длина	<input type="text" value="6"/>	Количество символов в коде подтверждения
Время действия	<input type="text" value="300"/>	Количество секунд, после которого код подтверждения перестает действовать. Необходима отправка нового кода
Количество попыток за 1 вход	<input type="text" value="3"/>	Количество неудачных попыток ввода кода подтверждения при одной попытке входа. Если количество попыток превышено, требуется отправка нового кода
Общее количество попыток	<input type="text" value="5"/>	Общее число отправок кодов подтверждения и попыток ввода кода подтверждения, после которого способ аутентификации будет временно заблокирован
Время блокировки при превышении попыток, в мин.	<input type="text" value="3"/>	В течение указанного времени способ аутентификации будет недоступен пользователю

## Способы отправки кода

Настройте способы отправки кодов подтверждения. Если будет выбрано более одного способа, то первый будет рассматриваться как основной, а остальные как резервные.

Отправлять	Атрибут с контактом	
<input type="text" value="SMS"/>	<input type="text" value="\$(phone_number-)"/>	

[+ Добавить способ отправки](#)

– правило выбора хранилища атрибутов для поиска введенного пользователем телефона.

## Правила выбора хранилища атрибутов

По умолчанию поиск пользователей для аутентификации происходит во всех активных хранилищах. В данном блоке 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 [строки подстановки](#).

[Посмотреть строки подстановки](#)

[Создать правило](#)

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

(страница 200).

### 2.2.13 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 входить 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 с известного устройства

Длительность запоминания устройства

Количество дней, в течение которого пользователю не потребуется повторный вход с известного устройства. Изменение будет доступно после перезапуска приложения.

### 2.2.14 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](#).

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 разовой ссылке

Время действия ссылки

Количество секунд, в течение которых действует автоматический вход по ссылке после успешной регистрации пользователя или восстановления забытого пароля. Изменение будет доступно после перезапуска приложения.

### 2.2.15 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.

**Вход по QR-коду**

Время действия QR-кода	<input type="text" value="120"/>
	Время в секундах, в течение которого действителен QR-код
Ссылка	<input type="text" value="app.link://universal.link"/>
	Ссылка в формате URI, которая будет закодирована в QR-коде
Ссылка на логотип	<input type="text" value="https://login.company.com/qrcode_logo.png"/>
	Ссылка на логотип в формате png. Если логотип указан, то он размещается в центре QR-кода.

Отмена
Обновить

### 2.2.16 Вход по входящему звонку

Blitz Identity Provider позволяет передавать одноразовые коды для реализации первого фактора аутентификации в номере входящего звонка (метод Flash Call). В этом случае на номер пользователя будет выполнен звонок с заранее неизвестного телефонного номера, последние цифры которого потребуются ввести для входа. Звонок выполняется с разрешения пользователя.

Для настройки метода входа по Flash Call 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 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` в первом списке добавьте блок настроек с методом `flashCall`:

```
"login" : {
  "factors" : [
    [
      {
        "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. Перейдите в раздел Аутентификация на вкладку Первый фактор. В разделе Вход по входящему звонку задайте следующие настройки:

**Вход по входящему звонку**

Подтверждения по входящему звонку Драйвер провайдера звонков

Для корректной идентификации пользователя укажите, каким образом должен формироваться логин и какому атрибуту в источнике данных он соответствует. Вы можете создать несколько альтернативных правил определения логина. Ввод логина не чувствителен к регистру.

Для создания правила используйте **строки подстановки**. Например, правило `CN=${login}` означает, что строка, введенная пользователем, будет сравниваться с атрибутом `CN` в хранилище данных.

[Посмотреть строки подстановки](#)

mobile  =

[+ добавить условие](#)

[+ добавить альтернативное правило](#)

Для подтверждения входа будет произведен звонок с заранее неизвестного телефонного номера. Пользователю необходимо ввести последние цифры входящего номера

**Длина кода**   
Количество последних цифр номера телефона входящего звонка

**Время действия**   
Количество секунд, после которого код подтверждения перестает действовать, и необходим повторный звонок

**Количество попыток за 1 вход**   
Количество неудачных попыток ввода кода подтверждения при одной попытке входа. Если количество попыток превышено, требуется повторный звонок

**Общее количество попыток ввода кода**   
Общее число попыток ввода кода подтверждения, после которого способ аутентификации будет временно заблокирован

**Время блокировки при превышении общего количества попыток ввода кода, в мин.**   
В течение указанного времени способ аутентификации будет недоступен пользователю

**Общее количество попыток звонка**   
Общее число звонков, после которого способ аутентификации будет временно заблокирован

**Время блокировки при превышении общего количества звонков, в мин.**   
В течение указанного времени способ аутентификации будет недоступен пользователю

**Название атрибута с мобильным номером пользователя**   
На телефон из данного атрибута пользователя будет производиться звонок

- **Длина кода:** количество последних цифр номера входящего звонка, которые будут использоваться в качестве кода на первом 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.
- **Общее количество попыток звонка:** общее число звонков, после которого способ аутентификации будет временно заблокирован.
- **Время блокировки при превышении общего количества звонков, в минутах:** в течение

ние указанного времени способ аутентификации будет недоступен пользователю.

- Название атрибута с мобильным номером пользователя: выберите из списка атрибут, в котором хранится номер телефона пользователя. На телефон из данного атрибута пользователя будет производиться звонок.

**Важно**

В разделе Источники данных убедитесь, что данный атрибут может быть использован для операций поиска.

Наименование атрибута	Тип значения	Поиск	Обяз.	Уник.	
uid	String	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
email	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
surname	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
mobile	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
name	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Нажмите Сохранить. В результате конфигурация метода будет обновлена и отобразится вкладка Драйвер провайдера звонков.

Вход по входящему звонку

Подтверждение по входящему звонку [Драйвер провайдера звонков](#)

```

1 package flashcall;
2
3 import org.slf4j.LoggerFactory;
4 import org.slf4j.Logger;
5 import com.identityblitz.core.loop.http.HttpLoop;
6 import com.identityblitz.core.loop.http.HttpLoopRequest;
7 import com.identityblitz.core.loop.http.HttpLoopResult;
8 import com.identityblitz.core.loop.*;
9 import com.identityblitz.core.loop.http.*;
10 import com.identityblitz.json.Json;
11 import java.util.Collections;
12
13 public class FlashCallFactorFlow implements HttpLoop {
14     private final org.slf4j.Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.flow.dynamic");
15
16     @Override
17     public HttpLoopRequest run(final JsonObject obj, final HttpLoopResult result) {
18         if (result == null) {
19             final String number = obj.asString("mobile");
20             return HttpLoop.callBuilder("POST", "https://03b608980231a301322720ef274b8cfe.m.pipedream.net")
21                 .withHeader("X-Token", "17acf043a7767f9840ab7de86b79838c")
22                 .withBody(Json.empty().addString("id", "test_project").addString("dst_number",
23                     number.substring(number.length() - 10)))
24                 .withTimeout(2000)
25                 .build(Json.empty());
26         } else if (result.status() == 200) {
27             final JsonObject body = result.body();
28             String callerInfo = body.asString("CallerID").substring(0, body.asString("CallerID").length() - 4) + "****";
29             if (callerInfo.startsWith("8"))
30                 callerInfo = "7" + callerInfo.substring(1);
31             return HttpLoop.ok(Json.empty().addString("code", body.asString("CallerID")).addString("caller_info", callerInfo));
32             //return HttpLoop.ok(Json.empty().addString("code", "8499511111"));
33         } else if (result.status() == 502) {
34             return HttpLoop.error("bad_gateway",
35                 Collections.<String, String>singletonMap("status", "" + result.status()));
36         } else {
37             return HttpLoop.error("wrong_http_status",
38                 Collections.<String, String>singletonMap("status", "" + result.status()));
39         }
40     }
41 }
42 }

```

[Отменить](#) [Сохранить](#)

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.

Листинг 1: Пример процедуры для интеграции с провайдером Flash Call

```

package flashcall;

import org.slf4j.LoggerFactory;
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 FlashCall1FactorFlow 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("mobile");
            return HttpLoop.callBuilder("POST", "https://
↳03b60898.m.pipedream.net")
                .withHeader("X-Token", "17acf043")
                .withBody(JsObj.empty.addString("id",
↳"test_project").addString("dst_number",
                    number.
↳substring(number.length() - 10)))
                .withTimeout(2000)
                .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) + "****";
            if (callerInfo.startsWith("8"))
                callerInfo = "7" + callerInfo.substring(1);
            return HttpLoop.Ok(JsObj.empty.addString("code",
↳body.asString("CallerID")).addString("caller_info", callerInfo));
            //return HttpLoop.Ok(JsObj.empty.addString("code
↳", "84995111111"));
        } 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 265) об имплементации кастомных ошибок.

3. Включите метод Вход по входящему звонку в списке методов на вкладке Аутентификация -> Первый фактор.

### 2.2.17 Вход разовым паролем на основе времени (TOTP)

Для проверки первого 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»<sup>29</sup>. 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.

Для настройки метода входа по Разовому паролю на основе времени (TOTP) 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 *Разовый пароль на основе времени (TOTP)*, выполните следующие действия:

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

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

2. В разделе `blitz.prod.local.idp.login.factors` в первый фактор добавьте:

```
"login" : {
  "factors" : [
    [
      {
        "enabled" : false,
        "method" : "totp"
      },
      ...
    ],
    [
      ...
    ]
  ],
  ...
}
```

3. Перезапустите сервисы.

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

<sup>29</sup> <https://tools.ietf.org/html/rfc6238>



## Шаг 2. Настройка метода в консоли

Общие настройки для данного метода задаются в разделе *TOTP-генераторы* (страница 86), а на вкладке Первый фактор в разделе Разовый пароль на основе времени (TOTP) только специфичные.

В консоли управления настройте и активируйте метод входа по TOTP. Для этого перейдите в раздел Аутентификация на вкладку Первый фактор. В разделе Разовый пароль на основе времени (TOTP) задайте следующие настройки:

- Для корректной идентификации пользователя укажите, каким образом должен формироваться логин и какому атрибуту в источнике данных он соответствует.
- **Общее количество попыток:** общее число попыток ввода кода подтверждения, после которого способ аутентификации будет временно заблокирован.
- **Время блокировки при превышении попыток, в минутах:** в течение указанного времени способ аутентификации будет недоступен пользователю.

**Разовый пароль на основе времени (TOTP)**

Для корректной идентификации пользователя укажите, каким образом должен формироваться логин и какому атрибуту в источнике данных он соответствует. Вы можете создать несколько альтернативных правил определения логина. Ввод логина не чувствителен к регистру.

Для создания правила используйте **строки подстановки**. Например, правило `CN=${login}` означает, что строка, введенная пользователем, будет сравниваться с атрибутом `CN` в хранилище данных.

[Посмотреть строки подстановки](#)

mail = \$(login) ✖

[+ добавить условие](#)

[+ добавить альтернативное правило](#)

---

**Базовые настройки метода**

Для корректной работы входа с помощью разового пароля, сгенерированного методом TOTP, необходимо указать базовые настройки метода. Некоторые настройки метода указываются при привязке устройства к учетной записи пользователя (см. раздел "Пользователи").

Общее количество попыток:   
Общее число попыток ввода кода подтверждения, после которого способ аутентификации будет временно заблокирован

Время блокировки при превышении попыток, в мин.:   
В течение указанного времени способ аутентификации будет недоступен пользователю

[Отмена](#) [Сохранить](#)

В блоке Правила выбора хранилища атрибутов можно настроить правила, при выполнении которых поиск пользователя будет осуществляться только в указанном хранилище. Правила выбора хранилища атрибутов описаны в разделе *вход по логину и паролю* (страница 90).

### 2.2.18 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. 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`.

**i Примечание**

Можно добавить несколько методов.

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

3. Перезагрузите сервисы.

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

**Шаг 2. 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 выполняться одновременно, чтобы пользователь был идентифицирован, а также ввести альтернативное правило.

**Автоматическая идентификация**

Идентификатор

Для корректной работы метода укажите, какие свойства текущей сессии соответствуют каким атрибутам в источнике данных. Вы можете создать несколько альтернативных правил.  
 Например, правило `phone_number=${p_msisdn}` означает, что свойство сессии `p_msisdn`, будет сравниваться с атрибутом `phone_number` в хранилище данных.

=

[+ добавить условие](#)

[+ добавить альтернативное правило](#)

Не показывать пользователю экран с подтверждением входа

Идентификатор пользователя

Выражение формируется из текущих атрибутов пользователя. Вычисленный результат отображается пользователю на странице подтверждения входа.

- 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.

- Нажмите Сохранить.
- 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.

**Правила выбора хранилища атрибутов**

По умолчанию поиск пользователей для аутентификации происходит во всех активных хранилищах. В данном блоке 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 [строки подстановки](#).

[Посмотреть строки подстановки](#)

**`\${\_rpId}`** идентификатор приложения (client\_id), в которое входит пользователь

Хранилище атрибутов	Правило соответствия	
<div style="border: 1px solid #ccc; padding: 2px;">bip-dldap01</div>	<input type="checkbox"/> not	<div style="border: 1px solid #ccc; padding: 2px;"> <input type="text" value="\${phone_number}"/> <input style="width: 150px;" type="text" value="+7(980).*"/> <span style="float: right; color: red; font-weight: bold;">✖</span> </div> <p style="text-align: right; font-size: small; color: blue;">+ Добавить альтернативное условие</p>
<p style="color: blue; font-weight: bold;">+ Добавить правило</p>		

Отмена
Сохранить

6. Нажмите Сохранить.

### Шаг 3. Создание процедуры входа

Для использования автоматической идентификации необходимо *создать процедуру входа* (страница 242), 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.

Пример кода, запускающего вход методом автоматической идентификации (`sprop_msisdn`), если свойство сессии `p_msisdn` имеет значение `+79161234567`:

```
//Свойство сессии и его значение, при которых будет выполнен вход
String propName = "p_msisdn";
String propValue = "+79161234567";
String propMethod = "sprop_msisdn";
// Добавление свойства сессии в контекст
Props sProps = ctx.sessionProps().addStringProp(propName, propValue);
return StrategyState
    .MORE_BUILDER()
    //Запуск входа по свойству сессии (метод с идентификатором sprop_msisdn),
    ↳альтернативные методы - вход по паролю и SMS.
    .addMethods(new String[] {propMethod, "password", "sms"})
    //Установка свойств сессии из контекста
    .setSessionProps(sProps)
    //Запуск метода sprop_msisdn (при условии, что он включен и сконфигурирован),
    ↳поиск пользователя с номером +79161234567, аутентификация, если пользователь
    ↳найден.
    .build();
```

#### Шаг 4. Кастомизация текстов

Если вы используете несколько методов *автоматической идентификации* (страница 113), следует провести кастомизацию текстов интерфейса для каждого из них, руководствуясь *алгоритмом* (страница 294).

В идентификатор текстовой строки понадобится включить имя метода или идентификатор метода. Имя метода *определено* (страница 114) 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
```

### 2.2.19 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»<sup>30</sup>.

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».

**Разовый пароль на основе секрета (HOTP)**

Для корректной работы входа с помощью разового пароля, сгенерированного методом HOTP, необходимо указать базовые настройки метода. Специфические настройки метода указываются при привязке устройства к учетной записи пользователя (см. раздел "Пользователи").

Допустимое отклонение	<input type="text" value="1"/>	Количество последующих кодов, которые могут быть введены для успешного входа
Отклонение для синхронизации	<input type="text" value="1000"/>	Величина диапазона, в пределах которого будет произведен поиск кодов при выполнении синхронизации
Общее количество попыток	<input type="text" value="5"/>	Общее число попыток ввода кода подтверждения, после которого способ аутентификации будет временно заблокирован
Время блокировки при превышении попыток, в мин.	<input type="text" value="15"/>	В течение указанного времени способ аутентификации будет недоступен пользователю

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

<sup>30</sup> <https://tools.ietf.org/html/rfc4226>

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;

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

## 2.2.20 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»<sup>31</sup>. В качестве таких 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.

Общие настройки для данного метода задаются в разделе *TOTP-генераторы* (страница 86), а на вкладке Второй фактор в разделе Разовый пароль на основе времени (TOTP) только специфичные.

Укажите специфичные настройки в данном разделе:

- Общее количество попыток - укажите общее число попыток ввода кода подтверждения, после которого способ аутентификации будет временно заблокирован;
- Время блокировки при превышении попыток, в минутах - укажите в течение какого времени способ аутентификации будет недоступен пользователю.

**Разовый пароль на основе времени (TOTP)**

---

**Базовые настройки метода**

Для корректной работы входа с помощью разового пароля, сгенерированного методом TOTP, необходимо указать базовые настройки метода. Некоторые настройки метода указываются при привязке устройства к учетной записи пользователя (см. раздел "Пользователи").

Общее количество попыток

Общее число попыток ввода кода подтверждения, после которого способ аутентификации будет временно заблокирован

Время блокировки при превышении попыток, в мин.

В течение указанного времени способ аутентификации будет недоступен пользователю

<sup>31</sup> <https://tools.ietf.org/html/rfc6238>

## 2.2.21 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 загружаемых генераторов (это может быть, например, имя устройства), формат данных, а также путь к файлу с описаниями устройств. По нажатию 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 присутствует 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 190));
- 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.

Генератор паролей на основе времени (TOTP)

Серийный номер	<input style="width: 90%;" type="text"/>
	Серийный номер устройства генерации разовых паролей
Значение	<input style="width: 90%;" type="text"/>

### 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 190));
- 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».

Генератор паролей на основе времени (TOTP)


Название генератора

Алгоритм шифрования

Длина пароля   
Число символов, из которых будет состоять разовый пароль

Время обновления пароля   
Время (в секундах), в течение которого будет обновляться разовый пароль

Секрет   
Секрет закодирован в Base32 кодировке



### 2.2.22 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 200)).

### Внимание

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


#### Подтверждение по коду из SMS/push

##### Параметры кодов подтверждения

Длина	<input type="text" value="6"/>	Количество символов в коде подтверждения
Время действия	<input type="text" value="60"/>	Количество секунд, после которого код подтверждения перестает действовать. Необходима отправка нового кода
Количество попыток за 1 вход	<input type="text" value="3"/>	Количество неудачных попыток ввода кода подтверждения при одной попытке входа. Если количество попыток превышено, требуется отправка нового кода
Общее количество попыток	<input type="text" value="5"/>	Общее число отправок кодов подтверждения и попыток ввода кода подтверждения, после которого способ аутентификации будет временно заблокирован
Время блокировки при превышении попыток, в мин.	<input type="text" value="3"/>	В течение указанного времени способ аутентификации будет недоступен пользователю

##### Способы отправки кода

Настройте способы отправки кодов подтверждения. Если будет выбрано более одного способа, то первый будет рассматриваться как основной, а остальные как резервные.

Отправлять	Атрибут с контактом	
<input type="text" value="SMS"/>	<input type="text" value="{phone_number-}"/>	

[+ Добавить способ отправки](#)

### 2.2.23 Коды подтверждения, отправляемые по электронной почте

Для подтверждения входа можно использовать отправляемые по электронной почте коды подтверждения.

**Подтверждение с помощью электронной почты**

---

**Параметры кодов подтверждения**

Длина   
Количество символов в коде подтверждения

Время действия   
Количество секунд, после которого код подтверждения перестает действовать. Необходима отправка нового кода

Количество попыток за 1 вход   
Количество неудачных попыток ввода кода подтверждения при одной попытке входа. Если количество попыток превышено, требуется отправка нового кода

Общее количество попыток   
Общее число отправок кодов подтверждения и попыток ввода кода подтверждения, после которого способ аутентификации будет временно заблокирован

Время блокировки при превышении попыток, в мин.   
В течение указанного времени способ аутентификации будет недоступен пользователю

---

**Параметры отправки**

Атрибут с контактом   
Выражение, по которому будет формироваться адрес электронной почты для отправки кода 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 200) подключение Blitz Identity Provider к SMTP-сервису.

### 2.2.24 Подтверждение входа с помощью Duo Mobile

Можно использовать мобильное приложение Duo Mobile<sup>32</sup> (компания Cisco) для подтверждения входа (второго фактора аутентификации).

Для этого необходимо выполнить настройки на стороне сервиса Duo Security:

<sup>32</sup> <https://duo.com/product/multi-factor-authentication-mfa/duo-mobile-app>

- зарегистрировать учетную запись на сайте Duo<sup>33</sup>;
- войти в панель администратора<sup>34</sup> 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.

<sup>33</sup> <https://signup.duo.com/>

<sup>34</sup> <https://admin.duosecurity.com/>

### Настройки Duo push-аутентификации

Для использования push-аутентификации от Duo Security необходимо:

- зарегистрировать учетную запись на [сайте Duo](#);
- войти в [панель администратора](#) 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.

---

**Учетная запись**

Имя хоста (API hostname)

Интеграционный ключ (integration key) [Изменить значение](#)

Секретный ключ (secret key) [Изменить значение](#)

---

**Параметры взаимодействия**

Шаблон имени пользователя   
Строка подстановки, определяющая имя пользователя в запросе на вход. Например, "{mail}"


Время действия кода активации   
Время (в секундах), в течение которого действителен код привязки (штрихкод)

---

**Данные для отображения в приложении**

При аутентификации может быть передана информация, которая будет отображена в мобильном приложении в виде "ключ: значение". Задайте необходимые ключи и их значения, используя строки подстановки. Например, `Имя = ${name} ${surname}` позволит передать ключ `Имя` со значением из атрибутов `name` и `surname`.

[Посмотреть строки подстановки](#)

Имя пользователя  =   [+ Добавить](#)

---

**Ссылки на приложения - Duo Mobile**

Укажите для каждой ОС, какие мобильные приложения рекомендуется использовать для push-аутентификации. Если ссылка не указана, то пользователям не будет предложено загрузить приложение для данной ОС.

iOS

Android

Windows Mobile

[Отмена](#) [Сохранить](#)

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:

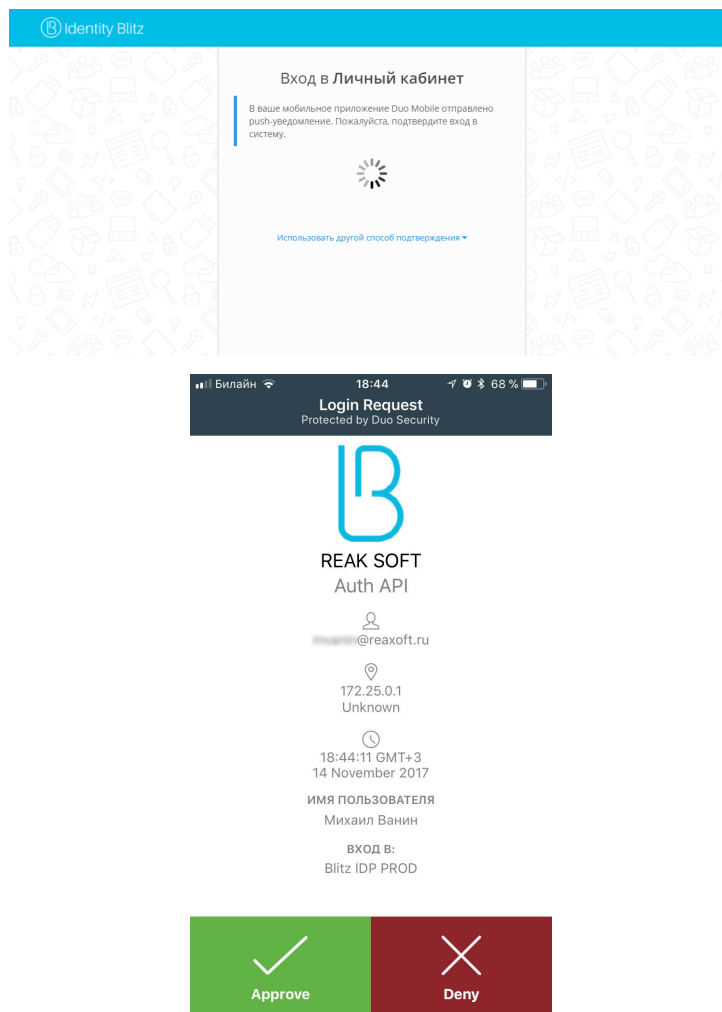
1. Выбрать способ подтверждения входа – Подтверждение с помощью мобильного приложения Duo Mobile.

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.

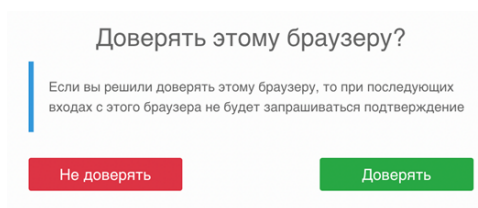
На рисунках приведен пример внешнего вида страницы входа при подтверждении входа с помощью push-уведомления в приложении Duo Mobile.



### 2.2.25 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.

## 2.2.26 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 128) контрольных вопросов.

**Подтверждение ответом на контрольный вопрос**

Для корректной работы входа с помощью контрольного вопроса укажите базовые настройки метода. Список контрольных вопросов доступен только для чтения. Для редактирования данного списка необходимо внести исправления в файл со строками пользовательского интерфейса.

Общее количество попыток   
 Общее число попыток ввода ответа, после которого способ аутентификации будет временно заблокирован

Время блокировки при превышении попыток, в мин.   
 В течение указанного времени способ аутентификации будет недоступен пользователю

---

**Список доступных контрольных вопросов**

0. Какая девичья фамилия у вашей матери

1. Какая девичья фамилия у вашей бабушки

2. Какой фильм вы впервые посмотрели в кинотеатре

3. Какое ваше любимое литературное произведение

4. Как звали вашего учителя в третьем классе

5. Первое блюдо, которое вы научились готовить

6. Как звали вашего первого питомца

7. Кем вы хотели стать в детстве

8. Как называлась первая школа, в которую вы ходили

9. Как называлась первая улица, где вы жили в детстве

## 2.2.27 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" : [
    [
      ...
    ],
    [
      {
        "enabled" : false,
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
        "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 мобильным номером пользователя:** выберите из списка атрибут, в котором хранится номер телефона пользователя для совершения звонка.

Подтверждение по входящему звонку

Подтверждение по входящему звонку

Для подтверждения входа будет произведен звонок с заранее неизвестного телефонного номера, а пользователю необходимо ввести последние цифры входящего номера

Длина кода   
Количество последних цифр номера телефона входящего звонка

Время действия   
Количество секунд, после которого код подтверждения перестает действовать и необходим повторный звонок

Количество попыток за 1 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 способ аутентификации будет временно заблокирован

Время блокировки при превышении общего количества попыток, в мин.   
В течение указанного времени способ аутентификации будет недоступен пользователю

Название атрибута с мобильным номером пользователя   
На телефон из данного атрибута пользователя будет производиться звонок

Нажмите Сохранить. В результате конфигурация метода будет обновлена и отобразится вкладка Драйвер провайдера звонков.

Конфигурация метода успешно обновлена

Подтверждение по входящему звонку

Подтверждение по входящему звонку

```

1
2 package flashcall;
3
4 import com.identityblitz.core.loop.JsonObj;
5 import com.identityblitz.core.loop.http.HttpLoop;
6 import com.identityblitz.core.loop.http.HttpLoopRequest;
7 import com.identityblitz.core.loop.http.HttpLoopResult;
8
9 public class FlashCallFlow implements HttpLoop {
10
11     @Override
12     public HttpLoopRequest run(JsonObj obj, HttpLoopResult result) {
13         throw new UnsupportedOperationException("Unimplemented method 'run'");
14     }
15
16 }
17

```

- 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.

Листинг 2: Пример процедуры для интеграции с провайдером Flash Call

```

package flashcall;

import org.slf4j.LoggerFactory;

```

(продолжается на следующей странице)

(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 265) об имплементации кастомных ошибок.

3. Включите метод Подтверждение по входящему звонку в списке методов на вкладке Аутентификация -> Второй фактор.

## 2.2.28 Подтверждение входа по паролю

Для настройки метода подтверждения входа по паролю выполните описанную ниже последовательность действий.

### Шаг 1. Добавление метода в blitz.conf

Если в методах аутентификации на вкладке Второй фактор отсутствует метод аутентификации Подтверждение с помощью пароля, то выполните следующие действия:

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

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

2. Добавьте метод входа по паролю в раздел `blitz.prod.local.idp.login.factors` во второй список:

```
"login" : {
  "factors" : [
    [
      ...
    ],
    [
      {
        "enabled" : false,
        "method" : "password"
      },
      ...
    ]
  ],
  ...
}
```

3. Перезапустите сервисы.

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

### Шаг 2. Настройка метода в консоли

В консоли управления настройте и активируйте метод подтверждения входа по паролю. Для этого выполните следующие действия:

1. Перейдите в раздел Аутентификация на вкладку Второй фактор. В разделе Подтверждение с помощью пароля задайте следующие настройки:
  - Разрешить пользователю пропустить смену пароля, не соответствующего парольным политикам: если флажок активен, пользователь сможет войти без смены пароля, даже если пароль не соответствует установленным политикам.
  - Всегда проверять текущий пароль пользователя на соответствие парольной политике: при включенном флажке система будет проверять пароль пользователя на соответствие парольным политикам при каждом входе.
  - Проверять при наличии HTTP-заголовка: если в HTTP-запросе присутствует указанный заголовок со значением true, пароль пользователя будет проверен на соответствие парольным политикам.
  - Кол-во неудачных попыток для временной блокировки: после превышения указанного числа неудачных попыток у пользователя будет временно заблокирован этот метод аутентификации.

- **Длительность временной блокировки:** задает продолжительность временной блокировки в минутах. После истечения этого времени пользователь сможет снова использовать данный метод.
- **Запрашивать доказательство выполнения работы:** при каждой проверке пароля браузер пользователя будет выполнять ресурсоемкую задачу. Результат этой задачи будет проверяться сервером вместе с паролем.
- **Запрашивать только при наличии HTTP-заголовка:** доказательство выполнения работы запрашивается только в случае, если HTTP-запрос содержит указанный заголовок со значением 1.
- **Показатель сложности работы:** коэффициент сложности от 1 до 160 бит. Каждый бит увеличивает сложность задачи в 2 раза.
- **Максимальный срок решения:** ограничение по времени в секундах, за которое браузер должен отправить результат выполнения задачи. Если значение не задано, по умолчанию используется 1800 секунд.

**Подтверждение с помощью пароля**

---

**Соответствие пароля парольной политике**

Разрешить пользователю пропустить смену пароля, не соответствующего парольным политикам

Всегда проверять текущий пароль пользователя на соответствие парольной политике

Проверять при наличии HTTP заголовка

Если HTTP-запрос будет содержать указанный заголовок со значением true, то текущий пароль пользователя будет проверен на соответствие парольной политике

Кол-во неудачных попыток для временной блокировки

После указанного кол-ва неудачных попыток будет установлена временная блокировка пользователю на использование данного метода аутентификации

Длительность временной блокировки

Определяет длительность временной блокировки в минутах по истечению которых пользователь снова сможет использовать данный метод аутентификации

---

**Доказательство выполнения работы**

При каждой проверке пароля на стороне браузера будет выполняться достаточно длительная работа. Результат выполнения работы будет проверяться сервером одновременно с проверкой пароля.

Запрашивать доказательство выполнения работы

Запрашивать только при наличии HTTP заголовка

Доказательство выполнения работ будет запрашиваться только при наличии в запросе HTTP заголовка с значением 1

Показатель сложности работы

Коэффициент от 1 до 160 бит. Каждый бит увеличивает сложность в 2 раза.

Максимальный срок решения

Максимальное время в секундах, за которое браузер должен прислать результат работы. Если значение не задано, то решение задачи ожидается за 1800 секунд.

## 2.2.29 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.

Добавление внешнего метода аутентификации

Идентификатор   
 Уникальное название (идентификатор) внешнего метода аутентификации. Будет использоваться в том числе и в аудите

URL сервиса   
 Адрес основного сервиса внешнего метода аутентификации. Принимает на вход текущую информацию о процессе аутентификации и возвращает HTTP-ответ, который отображается пользователю

Названия утверждений   
 Названия утверждений, которые сервис может установить пользователю

Передаваемые cookie   
 Названия cookies, которые будут пробрасываться при вызове сервиса метода

Передаваемые заголовки   
 Названия заголовков, которые будут пробрасываться при вызове сервиса 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   
 Перечислите утверждения, которые необходимо отправить внешнему методу аутентификации. Если перечень не задан, то отправляются все имеющиеся утверждения

Дополнительные параметры   
 Укажите дополнительные параметры в формате json, которые должны быть переданы в запросе к внешнему методу аутентификации

После сохранения включить метод

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.

## 2.2.30 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».

## 2.3 Внешние поставщики идентификации

Данный раздел посвящен настройке входа через внешние поставщики идентификации.

### 2.3.1 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 93) 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.

Подключенные внешние поставщики идентификации			
Название поставщика	Уникальное название	Тип поставщика	
Google	google_1	google	
Apple ID	apple_1	apple	
Яндекс	yandex_1	yandex	
Mail ID	mail_1	mail	
Facebook*	facebook_1	facebook	
Tinkoff ID	tcs_1	tcs	
VK	vk_1	vk	
Одноклассники	ok_1	ok	
ЕСИА	esia_1	esia	
Сбер ID	sbrf_1	sbrf	
Цифровой профиль ЕСИА	esiadp_1	esiadp	

[Добавить поставщика](#)
[Google](#)
[Apple](#)
[Яндекс](#)
[Mail ID](#)
[Facebook](#)
[Тинькофф ID](#)
[ВКонтакте](#)
[Одноклассники](#)
[ЕСИА](#)
[ЦП ЕСИА](#)
[Сбер ID](#)
[Blitz Identity Provider](#)
[СУДИР](#)

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 170) внешнего поставщика идентификации и учетной записи Blitz Identity Provider. Эти настройки не имеют специфики, зависящей от типа поставщика.

## 2.3.2 Отечественные поставщики

### Яндекс

Для конфигурирования входа через Яндекс необходимо выполнить следующие шаги:

1. Перейти в приложение Яндекс.OAuth<sup>35</sup>, 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.

<sup>35</sup> <https://oauth.yandex.ru/>

The screenshot shows the management interface for a provider named "Blitz-DEV", created on November 7, 2018. The interface is in Russian and includes a sidebar with navigation options like "Мои приложения", "Создать приложение", "Документация", and "Про Яндекс ID". The main content area displays the "Следующий шаг" (Next step) as "Получите данные пользователя" (Get user data), with sub-steps for getting a token and exchanging it for user data. Below this, there are several configuration fields: "Почта для связи" (Contact email), "Виджет «Мгновенный вход»" (Instant login widget), "Запрашиваемые права" (Requested permissions) which includes API ID, birth date, email, and login info, and "Время жизни токена" (Token lifetime) set to 1 year. On the right side, there are fields for "ClientID" (bf75d3b303ae474b9573b787b2633c81), "Client secret" (with an "Обновить" button), and "Redirect URI для веб-сервисов" (with two example URLs and a "Показать ещё" button). A "Задать вопрос" (Ask question) button is located at the bottom right.

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;
- 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.

## Безопасность

Для заполнения используйте данные из приложения [Яндекс.OAuth](#). Не забудьте сохранить в настройках приложения Яндекс.OAuth указанные URI перенаправления, а также отметить в разделе [Доступы/API Яндекс.Паспорта](#) данные, которые необходимо получать от Яндекса.

URI перенаправления (Callback URI)

Эти ссылки должны быть прописаны в параметре Callback URI приложения Яндекс.OAuth для корректной обработки результатов аутентификации пользователя. Используйте схему https, если вы используете защищенное соединение.

URL для авторизации

URL для получения и обновления маркера

Запоминать маркеры

URL для получения данных

ID приложения

Пароль приложения [Изменить значение](#)

## Разрешения

Запрашиваемые разрешения

Для добавления разрешения введите его имя и нажмите Enter

Укажите перечень разрешений (scope), которые должны быть получены при обращении к поставщику идентификации. [Перечень доступных для приложения разрешений Яндекс](#)

## ВКонтакте

Для конфигурирования входа через ВКонтакте необходимо выполнить следующие шаги:

1. Перейти в панель VK для разработчиков<sup>36</sup>, 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).

<sup>36</sup> <https://new.vk.com/dev>

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;
  - 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<sup>37</sup>.
4. Настроить правила связывания.
  5. В разделе Аутентификация консоли управления включить использование метода аутентификации с использованием поставщика идентификации ВКонтакте.

**Настройки поставщика идентификации ВКонтакте**

**Безопасность**

Используйте раздел "Мои приложения" панели [VK для разработчиков](#) для заполнения указанных ниже параметров. Не забудьте сохранить в панели ВКонтакте указанные URI перенаправления

Версия

Доверенные redirect URI

Эти ссылки должны быть прописаны в настройках поставщика идентификации для корректной обработки результатов аутентификации пользователя. Используйте схему https, если вы используете защищенное соединение.

URL для авторизации

URL для получения и обновления маркера

Запоминать маркеры

URL для получения данных

ID приложения

Защищенный ключ [Изменить значение](#)

---

**Разрешения**

Запрашиваемые разрешения

Для добавления разрешения введите его имя и нажмите Enter

Укажите перечень разрешений (scope), которые должны быть получены при обращении к поставщику идентификации. [Перечень доступных разрешений ВКонтакте](#)

### Одноклассники

Для конфигурирования входа через Одноклассники необходимо выполнить следующие шаги:

1. Перейти на страницу OAuth авторизация<sup>38</sup>, 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:

<sup>37</sup> <https://new.vk.com/dev/permissions>

<sup>38</sup> <https://apiok.ru/ext/oauth/>

```
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 Одно-классники.

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;

**i** **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.

## Настройки поставщика идентификации Одноклассники

## Безопасность

Используйте раздел "Как начать использовать OAuth" страницы [OAuth авторизация](#) для заполнения указанных ниже параметров. Не забудьте сохранить в настройках приложения Одноклассники указанные ниже разрешенные `redirect_uri`

Разрешённые `redirect_uri` `https://bip-dev1.reaxoft.ru/blitz/login/externalIdps/callback/ok/ok_1/false https://bip-dev1.reaxoft.ru/blitz/profile/social/externalIdps/callbackPopup/ok/ok_1`

Эти ссылки должны быть прописаны в Список разрешённых `redirect_uri` приложения Одноклассники для корректной обработки результатов аутентификации пользователя. Используйте схему `https`, если вы используете защищенное соединение.

URL для авторизации

URL для получения и обновления маркера

Запоминать маркеры

URL для получения данных

Название приложения (Application ID)

Секретный ключ приложения [Изменить значение](#)

Публичный ключ приложения [Изменить значение](#)

## Разрешения

Запрашиваемые разрешения

Для добавления разрешения введите его имя и нажмите Enter

Укажите перечень разрешений (`scope`), которые должны быть получены при обращении к поставщику идентификации. [Перечень доступных разрешений Одноклассники](#)

## Mail ID

Для конфигурирования входа через Mail ID необходимо выполнить следующие шаги:

1. Перейти на страницу Создание приложения<sup>39</sup> 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.

<sup>39</sup> <https://help.mail.ru/developers/oauth/app>




### Редактирование приложения

ID Приложения / Client ID

Секрет / Client Secret

Название проекта

Добавить фото 

Изображение размером 96x96 (\*.png)

Все redirect\_uri

Введите в столбик все redirect URI, которые будут использоваться на вашем сайте или iOS/Android приложении (по одной ссылке в строке)

#### Платформы

Проставьте галочки, чтобы отметить, на каких платформах будет установлено ваше приложение.

Web

iOS

Android

#### Дополнительные возможности

Доступ к почтовому ящику по IMAP, POP и SMTP

Пройдите модерацию и получите доступ к большему числу возможностей

- One Tap Sign In авторизация вдвое увеличивает количество регистрирующихся пользователей.

#### Права доступа

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.

**Настройки поставщика идентификации Mail ID**

**Безопасность**

Для заполнения используйте данные из приложения [oauth@mail.ru](mailto:oauth@mail.ru). Не забудьте сохранить в настройках приложения OAUTH@MAIL.RU указанные URI перенаправления и в качестве используемой платформы выбрать Web.

URI перенаправления (*redirect\_uri*)

Эти ссылки должны быть прописаны в параметре *redirect\_uri* приложения [oauth@mail.ru](mailto:oauth@mail.ru) для корректной обработки результатов аутентификации пользователя. Используйте схему [https](https://), если вы используете защищенное соединение.

URL для авторизации

URL для получения и обновления маркера

Запоминать маркеры

URL для получения данных

ID приложения

Секрет приложения [Изменить значение](#)

---

**Разрешения**

Запрашиваемые разрешения

Для добавления разрешения введите его имя и нажмите Enter

**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.

#### Информация о приложении

ID приложения	Платформа
51818493	Web
Состояние приложения ?	
Приложение включено и видно всем	
Название приложения ?	
Identity Blitz	



Изображение ?

Заменить

Удалить

#### Ключи доступа

Защищённый ключ ?	Сервисный ключ доступа ?
.....	5ed98a2f5ed98a2f5ed98a2fd25dcf25d255ed95ed98: ↻

#### Подключение авторизации

Базовый домен ?

bip-dev1.reaxoft.ru

+ Добавить базовый домен

Доверенный Redirect URL ?

https://bip-dev1.reaxoft.ru/blitz/login/externalIdps/callback/vkid/vkid\_640/false

https://bip-dev1.reaxoft.ru/blitz/profile/social/externalIdps/callbackPopup/vkid/vkid\_640

+ Добавить доверенный Redirect URL

Сохранено

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.

**Настройки поставщика идентификации VK ID**

**Безопасность**

Используйте раздел "Приложения" панели VK ID для разработчиков для заполнения указанных ниже параметров. Не забудьте сохранить в панели VK ID указанные URI перенаправления

Версия

Доверенные redirect URI

Эти ссылки должны быть прописаны в настройках поставщика идентификации для корректной обработки результатов аутентификации пользователя. Используйте схему https, если вы используете защищенное соединение.

ID приложения

Сервисный ключ доступа [Изменить значение](#)

5. *Настройте* (страница 170) правила связывания.
6. В разделе Аутентификация консоли управления *включите* (страница 93) 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 329) 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 portalу ЕСИА.

2. Зарегистрируйте систему организации на Технологическом портале ЕСИА<sup>40</sup>:
  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.

<sup>40</sup> <https://esia.gosuslugi.ru/console/tech/>

Данные информационной системы
✕

### ОСНОВНЫЕ ДАННЫЕ СИСТЕМЫ

Название системы	<input type="text" value="Проверочная система"/>
Отображаемое название	<input type="text" value="Проверочная система"/>
Мнемоника системы	<input type="text" value="CHECKSYS"/>
Информация о системе	<input type="text" value="Проверочная система"/>
URL системы	<input type="text" value="http://identityblitz.ru"/> <span style="background-color: #4f81bd; color: white; padding: 2px 5px; border-radius: 3px;">+</span>
Алгоритм формирования электронной подписи	<input type="text" value="RS256"/>
URL для отправки push сообщений	<input type="text"/>

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».

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.

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.

Выберите криптографический алгоритм формирования электронной подписи, который будет использоваться при выпуске маркеров доступа, маркеров идентификации, маркеров обновления, кода авторизации

Введите адрес (с префиксом "https://"), который будет использоваться ЕСИА для отправки в ИС сообщений - уведомлений (push-сообщений)

### КАТЕГОРИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Категория информационной системы	<input type="text" value="По умолчанию"/>
Время жизни access token, мин	<input type="text" value="60"/>
Время жизни refresh token, мин	<input type="text" value="1051200"/>

min: 10; max: 180

min: 60; max: 1051200

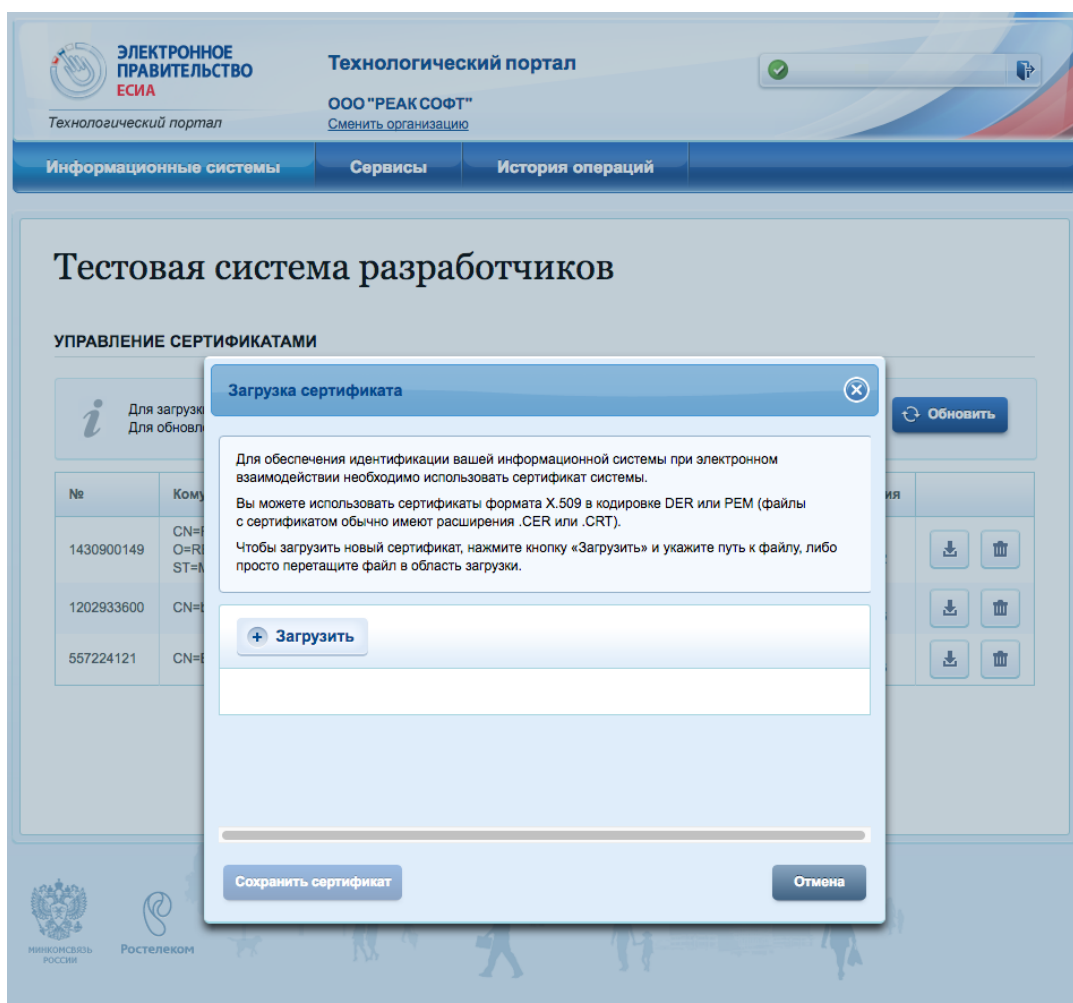
### ОТВЕТСТВЕННЫЙ ЗА ЭКСПЛУАТАЦИЮ СИСТЕМЫ

ФИО	<input type="text" value="Иванов Иван Иванович"/>
Адрес электронной почты	<input type="text"/>
Номер телефона	<input type="text"/>

Введите имя ответственного сотрудника вашей организации и выберите его из выпадающего списка. Пользователь должен быть присоединен к учетной записи вашей организации.

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 Техно-

логическом портале ЕСИА.

#### Настройки поставщика идентификации ЕСИА

##### Безопасность

Заполните данные для корректного взаимодействия Blitz Identity Provider с ЕСИА.

URL для авторизации

URL для получения и обновления маркера

Запоминать маркеры

URL для получения данных

Мнемоника системы (client\_id)

Версия протокола

Сервис подписи

После заполнения этих данных не забудьте перейти в [Технологический портал ЕСИА](#), где должна быть зарегистрирована информационная система с указанной мнемоникой и сертификатом ключа электронной подписи.

- 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.

#### Разрешения и данные пользователя

Выберите разрешения из доступного списка

[Доступные разрешения](#)

##### Запрашиваемые разрешения

openid  fullname  mobile  usr\_org  email  id\_doc  snils   
 addresses

Для добавления разрешения введите его имя и нажмите Enter

Укажите перечень разрешений (scope), которые должны быть получены при обращении к поставщику идентификации.

Запрашиваемые данные пользователя  Основные данные  Документы  Адреса  Контакты

Отмеченные ранее разрешения (scope) должны позволять получать указанные данные

#### ⓘ 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](#)<sup>41</sup>.

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.

<sup>41</sup> <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 329) 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 portalу ЕСИА.

2. Зарегистрируйте систему организации на Технологическом портале ЕСИА<sup>42</sup>:
  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.

<sup>42</sup> <https://esia.gosuslugi.ru/console/tech/>

Данные информационной системы
✕

**ОСНОВНЫЕ ДАННЫЕ СИСТЕМЫ**

Название системы

Отображаемое название   
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   
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

URL системы  +  
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   
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

Время жизни access token, мин   
min: 10; max: 180

Время жизни refresh token, мин   
min: 60; max: 1051200

**ОТВЕТСТВЕННЫЙ ЗА ЭКСПЛУАТАЦИЮ СИСТЕМЫ**

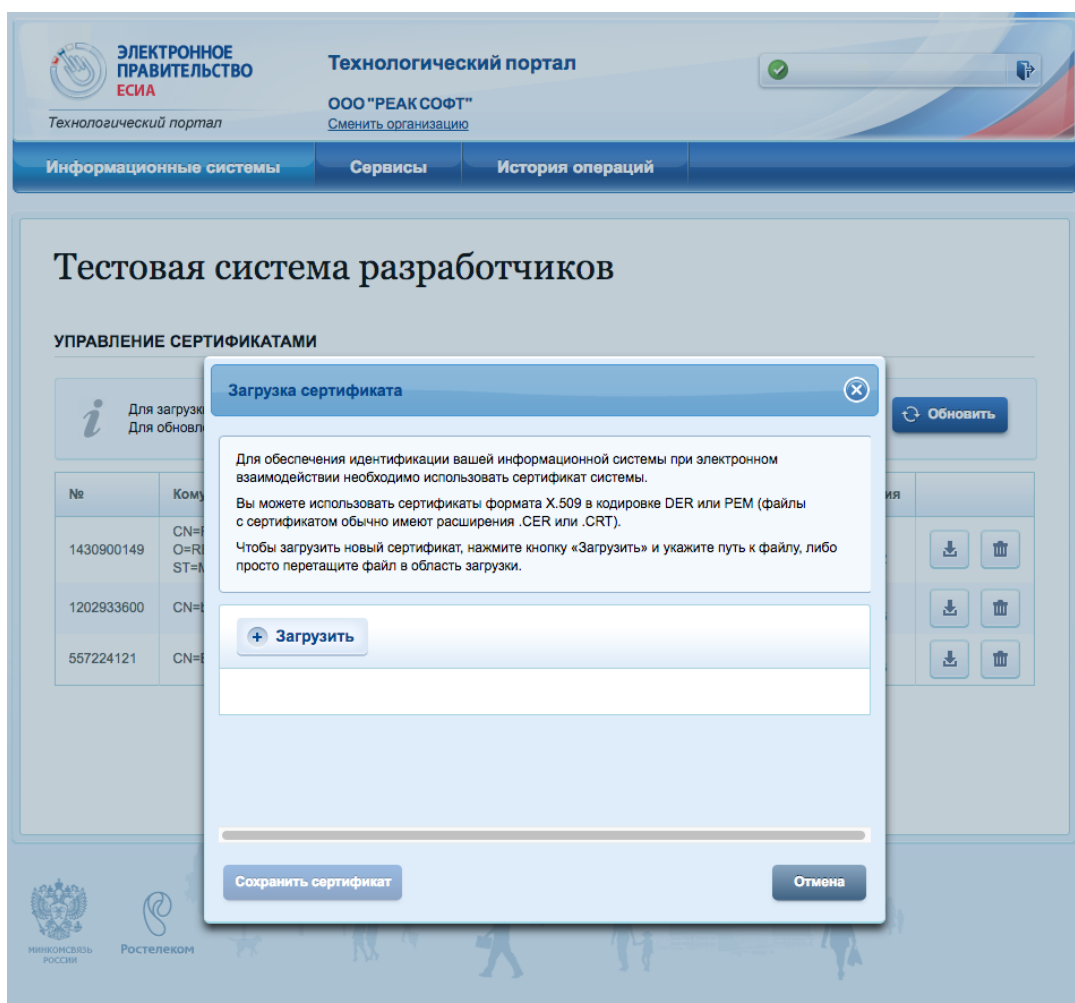
ФИО   
Введите имя ответственного сотрудника вашей организации и выберите его из выпадающего списка. Пользователь должен быть присоединен к учетной записи вашей организации.

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.

**Настройки поставщика идентификации Цифровой профиль ЕСИА**

**Безопасность**

Заполните данные для корректного взаимодействия Blitz Identity Provider с ЕСИА.

URL для авторизации

URL для получения и обновления маркера

Запоминать маркеры

URL для получения данных

Мнемоника системы (client\_id)

Версия протокола

Сервис подписи

Тип хранилища ключа электронной подписи

Идентификатор (alias) ключа электронной подписи

Предварительно ключ электронной подписи должен быть загружен в хранилище (keystore).

Пароль доступа к ключу электронной подписи [Изменить значение](#)

После заполнения этих данных не забудьте перейти в [Технологический портал ЕСИА](#), где должна быть зарегистрирована информационная система с указанной мнемоникой и сертификатом ключа электронной подписи.

- 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.

**Разрешения и данные пользователя**

Выберите разрешения из доступного списка

[Доступные разрешения](#)

Запрашиваемые разрешения

Для добавления разрешения введите его имя и нажмите Enter  
Укажите перечень разрешений (scope), которые должны быть получены при обращении к поставщику идентификации.

Тип согласия

Обычно совпадает с целью, разрешенной организации для запроса данных Цифрового профиля

Срок действия согласия

Количество минут, на которое запрашивается согласие. Не может превышать максимальный срок действия, определенный для данного типа согласия.

Ответственное лицо

Сотрудник организации или организация, осуществляющие обработку данных (строка с ФИО или другой информацией)

Запрашиваемые данные пользователя  Основные данные  Паспорт гражданина РФ

Отмеченные ранее разрешения (scope) должны позволять получать указанные данные

**Важно**

Чтобы вход через Цифровой профиль ЕСИА заработал, необходимо получить официальное разрешение на проведение идентификации и аутентификации пользователей с помощью зарегистрированной системы и получить доступ к тестовой / промышленной среде ЕСИА с доступом к цифровому профилю. Подробнее об этом можно прочитать здесь<sup>43</sup>.

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<sup>44</sup>. 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`);

<sup>43</sup> <https://identityblitz.ru/products/esia-bridge/digital-profile/>

<sup>44</sup> <https://developers.sber.ru/docs/ru/sberid/overview>

- Секрет клиента (`Client Secret`);
  - 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.

### Настройки поставщика идентификации Сбер ID

#### Безопасность

Для заполнения используйте данные, полученные при регистрации своей системы в Банке.

**URI перенаправления (Redirect URI)** `https://bip-dev1.reaxoft.ru/blitz/login/externalldps/callback/sbrf/sbrf_1/false https://bip-dev1.reaxoft.ru/blitz/profile/social/externalldps/callbackPopup/sbrf/sbrf_1`

Адрес страницы, на которую будет перенаправлен ответ после успешной аутентификации в системе Банка.

**URL для авторизации** `https://online.sberbank.ru/CSAFront/oidc/sberbank_id/authorize.do`

**URL для получения и обновления маркера** `https://sec.api.sberbank.ru/ru/prod/tokens/v2/oidc`

Запоминать маркеры

**URL для получения данных** `https://sec.api.sberbank.ru/ru/prod/sberbankid/v2.1/userinfo`

**Client ID** `d`

**Client Secret** [Изменить значение](#)

#### Группы данных

Запрашиваемые группы данных

`name x` `email x` `openid x` `birthdate x` `maindoc x` `inn x` `mobile x`  
`test_scope x` `snils x` `gender x`

Для добавления группы данных введите ее имя и нажмите Enter

### 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<sup>45</sup>. 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;

<sup>45</sup> <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;

**i** **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.



**Настройки поставщика идентификации T-ID**

**Безопасность**

Для заполнения используйте данные, полученные при регистрации своей системы в Банке.

URI перенаправления (Redirect URI)

Адрес страницы, на которую будет перенаправлен ответ после успешной аутентификации в системе Банка.

URL для авторизации

URL для получения и обновления маркера

Запоминать маркеры

URL для получения данных

URL для получения паспортных данных

URL для получения данных о СНИЛС

URL для получения данных о подтвержденности пользователя

Client ID

Client Secret [Изменить значение](#)

Запрашиваемые данные пользователя  Основные данные  Паспорт  СНИЛС  Признак идентификации

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<sup>46</sup>. 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 иницироваться 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:

- при входе пользователя;

<sup>46</sup> <https://developer.vtb.ru/connection>

- 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.

#### Настройки поставщика идентификации ВТБ ID

##### Безопасность

Для заполнения используйте данные, полученные при регистрации своей системы в Банке.

**URI перенаправления (Redirect URI)** `https://bip-dev1.reaxoft.ru/blitz/login/externalIdps/callback/vtb/vtb_1/false` `https://bip-dev1.reaxoft.ru/blitz/profile/social/externalIdps/callbackPopup/vtb/vtb_1`

Адрес страницы, на которую будет перенаправлен ответ после успешной аутентификации в системе Банка.

**URL для авторизации**

**URL для получения и обновления маркера**

Запоминать маркеры

**URL для получения данных**

Запрос на получение данных должен происходить через прокси-сервер

**Client ID**

**Client Secret** [Изменить значение](#)

##### Группы данных

Запрашиваемые группы данных

name  email  patronymic  surname  rPassport  mainMobilePhone  birthDate  snils  gender

Для добавления группы данных введите ее имя и нажмите Enter

## СберБизнес ID

Для конфигурирования входа через СберБизнес ID необходимо выполнить следующие шаги:

1. Зарегистрировать приложение в системе СберБизнес ID<sup>47</sup>. 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.

<sup>47</sup> [https://api.developer.sber.ru/how-to-use/sberbusiness\\_id](https://api.developer.sber.ru/how-to-use/sberbusiness_id)

## Настройки поставщика идентификации СберБизнес ID

## Безопасность

Для заполнения используйте данные, полученные при регистрации своей системы в Банке.

**URI перенаправления (Redirect URI)** `https://bip-dev1.reaxoft.ru/blitz/login/externalldps/callback/sbb/sbb_1/false`  
`https://bip-dev1.reaxoft.ru/blitz/profile/social/externalldps/callbackPopup/sbb/sbb_1`

Адрес страницы, на которую будет перенаправлен ответ после успешной аутентификации в системе Банка.

**URL для авторизации**

**URL для получения и обновления маркера**

Запрос на получение маркера происходит через прокси-сервер

Запоминать маркеры

**URL для получения данных**

Запрос на получение данных должен происходить через прокси-сервер

**Client ID**

**Client Secret** [Изменить значение](#)

## Группы данных

Запрашиваемые группы данных

```
OrgName x | orgJuridicalAddress x | email x
orgOktmo x | orgOgrn x | orgKpp x
orgActualAddress x | openid x | orgLawFormShort x
individualExecutiveAgency x | orgLawForm x | inn x
offerExpirationDate x | terBank x | accounts x | name x
phone_number x | orgFullName x | userPosition x
```

Для добавления группы данных введите ее имя и нажмите Enter

## Альфа ID

Для конфигурирования входа через Альфа ID необходимо выполнить следующие настройки:

1. Зарегистрировать приложение в системе Альфа ID<sup>48</sup>. 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 Альфа ID.

<sup>48</sup> <https://alfabank.ru/sme/alfaid/>

### 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.

## Безопасность

Для заполнения используйте данные, полученные при регистрации своей системы в Банке.

**URI перенаправления (Redirect URI)** `https://bip-dev1.reaxoft.ru/blitz/login/externaldps/callback/alfa/alfa_1/false` `https://bip-dev1.reaxoft.ru/blitz/profile/social/externaldps/callbackPopup/alfa/alfa_1`

Адрес страницы, на которую будет перенаправлен ответ после успешной аутентификации в системе Банка.

**URL для авторизации**

**URL для получения и обновления маркера**

Запрос на получение маркера происходит через прокси-сервер

Запоминать маркеры

**URL для получения данных**

Запрос на получение данных должен происходить через прокси-сервер

**Client ID**

**Client Secret** [Изменить значение](#)

## Группы данных

Запрашиваемые группы данных

identitydocument x email x openid x identitydocumentdetails x inn x  
birthplace x phone x addressfl x profile x

Для добавления группы данных введите ее имя и нажмите Enter

## 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.

## Настройки поставщика идентификации СУДИР

## Безопасность

Заполните данные для корректного взаимодействия Blitz Identity Provider с системой управления доступом города Москвы. Подробнее о получении доступа и настройках подключения к portalу Москвы (mos.ru) можно посмотреть [здесь](#). Информация о работе с внутреннем контуром СУДИР размещена [здесь](#).

Предопределенные ссылки возврата (redirect\_uri) `https://bip-dev1.reaxoft.ru/blitz/login/externalIdps/callback/mos/mos_1/false`  
`https://bip-dev1.reaxoft.ru/blitz/profile/social/externalIdps/callbackPopup/mos/mos_1`

Эти ссылки должны быть прописаны в настройках поставщика идентификации для корректной обработки результатов аутентификации пользователя. Используйте схему https, если вы используете защищенное соединение.

URL для авторизации

URL для получения и обновления маркера

Запоминать маркеры

URL для получения данных

Идентификатор (client\_id)

Секрет (client\_secret) [Изменить значение](#)

## Разрешения

Запрашиваемые разрешения

Для добавления разрешения введите его имя и нажмите Enter

Укажите перечень разрешений (scope), которые должны быть получены при обращении к поставщику идентификации. Обратитесь к администратору внешнего поставщика идентификации Blitz Identity Provider, чтобы получить перечень доступных разрешений

## Идентификация учетных записей

Укажите уникальный атрибут внешнего поставщика идентификации, который будет использоваться для связи учетной записи в Blitz Identity Provider.

Идентификатор

### 2.3.3 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.

### Базовые настройки Blitz Identity Provider

Идентификатор поставщика   
Уникальный идентификатор поставщика. Используется только внутри Blitz Identity Provider

Название поставщика   
Отображаемое в консоли имя поставщика. Используется только внутри Blitz Identity Provider

---

### Настройки поставщика идентификации Blitz Identity Provider

#### Безопасность

Для заполнения указанных параметров обратитесь к администратору внешнего поставщика идентификации Blitz Identity Provider. Необходимая информация размещена в свойствах подключаемого приложения (по протоколу OAuth 2.0). Также передайте администратору приведенные ниже URI перенаправления.

Предопределенные ссылки возврата (redirect\_uri)   

  
Эти ссылки должны быть прописаны в настройках поставщика идентификации для корректной обработки результатов аутентификации пользователя. Используйте схему https, если вы используете защищенное соединение.

URL для авторизации

URL для получения и обновления маркера

URL для получения данных

Идентификатор (client\_id)

Секрет (client\_secret) [Изменить значение](#)

---

#### Разрешения

Запрашиваемые разрешения

Для добавления разрешения введите его имя и нажмите Enter

Укажите перечень разрешений (scope), которые должны быть получены при обращении к поставщику идентификации. Обратитесь к администратору внешнего поставщика идентификации Blitz Identity Provider, чтобы получить перечень доступных разрешений

---

#### Идентификация учетных записей

Укажите уникальный атрибут внешнего поставщика идентификации, который будет использоваться для связи учетной записи в Blitz Identity Provider.

Идентификатор

### 2.3.4 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 регистрации 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 сопоставляться 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).

**Связывание учетных записей**

Базовая настройка | Расширенная настройка

**Идентификация учетных записей**

Укажите правила соответствия учетных записей Blitz Identity Provider и поставщика идентификации. При первом входе пользователя через поставщика идентификации с помощью этих правил будет осуществляться поиск учетной записи в Blitz Identity Provider для ее последующего связывания с учетной записью поставщика идентификации.

Для создания правила используйте строки подстановки `${attr_name}`, где `attr_name` - это имя атрибута, получаемого от поставщика идентификации. Вы можете указывать в одном правиле несколько атрибутов. Например, правило `CN=${name} ${surname}` означает, что атрибут CN будет формироваться из двух атрибутов - `name` и `surname` через пробел.

Разрешить привязывать одну учетную запись поставщика идентификации ко многим аккаунтам Blitz Identity Provider

Предлагать пользователю аутентифицироваться для привязки учетной записи, если учетная запись не была найдена в Blitz Identity Provider по результатам поиска

Разрешить регистрацию пользователя

Для привязки должна быть найдена только одна учетная запись по заданным правилам соответствия

Требовать аутентификацию пользователя, если учетная запись была найдена в Blitz Identity Provider по результатам поиска

esialid = \${oid} ✖

[+ добавить условие](#)

[+ 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.

#### Атрибуты

Укажите, каким образом должны формироваться атрибуты, используемые в Blitz Identity Provider, на основе данных, получаемых от поставщика идентификации. Для формирования каждого атрибута должно быть создано свое правило.

Для создания правила используйте обозначение `${attr_name}`, где `attr_name` - это имя атрибута, получаемого от поставщика идентификации. Вы можете указывать в одном правиле несколько атрибутов. Например, правило `CN=${name} ${surname}` означает, что атрибут CN будет формироваться из двух атрибутов - `name` и `surname` через пробел. Правило можно использовать для задания константного или вычисляемого значения. Например, правило `uid=BIP-${&random(4)}` позволит присвоить атрибуту `uid` значение `BIP-XXXXXX`, где `XXXXXX` - случайно сгенерированная величина (набор цифр и букв латинского алфавита).

[Доступные атрибуты для маппинга](#)

Атрибут	Правило	Мастер	
esialid	= \${oid}	<input type="checkbox"/>	<span style="color: red; font-weight: bold;">✖</span>

[+ Добавить атрибут](#)

- 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 `*.`

#### Выбор пользователя

Выбор пользователя возникает, если под критерии связывания подходят несколько аккаунтов или учетная запись связана с несколькими пользователями

Имя пользователя

Строка подстановки для отображения имени пользователя

Идентификатор пользователя

Строка подстановки для отображения идентификатора пользователя

- 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.

#### Привязанная учетная запись

Информация о привязанной учетной записи будет отображаться пользователю в карточке внешнего поставщика

Имя учетной записи

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.

## 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_IEsia`. 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.

```

1 package com.identityblitz.idp.federation.matching.dynamic;
2
3 import java.lang.*;
4 import java.util.*;
5 import java.text.*;
6 import java.time.*;
7 import java.math.*;
8 import java.security.*;
9 import javax.crypto.*;
10 import org.slf4j.LoggerFactory;
11 import org.slf4j.Logger;
12 import com.identityblitz.idp.federation.*;
13 import com.identityblitz.idp.federation.matching.*;
14 import com.identityblitz.idp.flow.common.api.*;
15 import com.identityblitz.idp.flow.common.model.*;
16 import com.identityblitz.idp.federation.matching.dynamic.*;
17 import java.util.function.Consumer;
18 import java.util.stream.Stream;
19 import java.util.stream.Collectors;
20 import org.codehaus.jackson.map.ObjectMapper;
21 import org.codehaus.jackson.type.TypeReference;
22 import com.identityblitz.idp.extensions.types.JsonObject;
23
24 import com.identityblitz.idp.federation.matching.*;
25 import com.identityblitz.idp.flow.common.api.HttpFactory;
26
27 /**
28  * Класс наследуется от MatchingBlock и для корректного инстантирования должен иметь default конструктор.
29  * Текущая сгенерированная реализация обеспечивает стратегию при которой пользователи не сопоставляются и не обновляются.
30  */
31 public class Esia_IEsia extends MatchingBlock {
32     private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.federation.matching.dynamic");
33
34     /**
35      * Итеративная функция определяющая соответствие внутренних УЗ и УЗ поставщика идентификации.
36      * На каждой итерации функция может выполнить операцию find (найденные пользователи будут переданы в следующей итерации)
37      * или завершить операцию со следующими решениями:
38      * matched – соответствующие пользователи найдены;
39      * matchError – ошибка определения соответствия пользователей;
40      * matchByLogin – осуществить связь с пользователем, который успешно аутентифицируется;
41      * refine – получит список пользователей, запрашивает пароль и осуществляет связь с тем пользователем, для которого введен корректный пароль;
42      * @param ctx – контекст процедур со следующими полями:
43      * iteration – номер итерации процедуры;
44      * extAttrs – атрибуты пользователя, полученные от поставщика идентификации;
45      * sid – уникальный идентификатор внешней УЗ.
46      * @param users – пользователи.
47      * @return – одно из перечисленных решений
48      */
49     @Override public MatchResult match(MatchingContext ctx, List<MatchingUser> users){
50         return matchError(ctx, new MatchingError("not_matched", "User not matched"));
51     };
52
53     /**
54      * Возвращает обновляемые и удаляемые атрибуты.
55      * @param extAttrs – атрибуты пользователя, полученные от поставщика идентификации.
56      * @param user – внутренний пользователь.
57      * @param justMatched – признак того, что связь внутренних УЗ с УЗ внешнего поставщика установлена впервые.
58      * @return – кортеж с изменяемыми и удаляемыми атрибутами. Например: change(JsonObject.empty(), Collections.<String>emptySet())
59      */
60     @Override public Tuple2<JsonObject, Set<String>> update(JsonObject extAttrs, MatchingUser user, Boolean justMatched, HttpFactory httpFactory){
61         return change(JsonObject.empty(), Collections.<String>emptySet());
62     };
63 }
64
65

```

Отмена

Сохранить

 **См. также**

*Процедуры привязки аккаунтов внешних поставщиков (страница 268)*

## 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.

### 2.4.1 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.

### Сервисы самообслуживания

Регистрация

Самостоятельная регистрация пользователей.

[Перейти к настройкам](#)

Восстановление доступа

Самостоятельное восстановление доступа посредством отправки ссылки на адрес электронной почты или кода подтверждения в SMS-сообщении.

[Перейти к настройкам](#)

Личный кабинет

Возможность редактировать свои данные, включить усиленную аутентификацию, изменить настройки безопасности.

[Перейти к настройкам](#)

---

### Общие настройки

Задайте параметры кодов подтверждения, отправляемых по SMS и электронной почте. Эти коды используются при регистрации пользователей, для восстановления доступа к учетной записи, а также при изменении номера мобильного телефона / адреса электронной почты через Личный кабинет.

---

**Параметры кода подтверждения, отправляемого в SMS**

Длина кода  Число символов в коде

Время действия кода (в секундах)  Время, после которого код перестает действовать

---

**Параметры кода подтверждения, отправляемого по электронной почте**

Длина кода  Число символов в коде

Время действия кода (в секундах)  Время, после которого код перестает действовать

[Сохранить](#)

В подразделах осуществляется настройка каждого сервиса самообслуживания в отдельности.

## 2.4.2 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<sup>49</sup>. 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"))` – отобра-

<sup>49</sup> <https://github.com/playframework/twirl>



жает 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 выполнении 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:



### Регистрация в Личный кабинет

Фамилия

Имя

Номер мобильного телефона

Адрес электронной почты

Придумайте пароль

Повторите пароль, чтобы не ошибиться

Пароль должен состоять не менее чем из 8 символов.  
Рекомендуется, чтобы пароль состоял из прописных и строчных букв и имел хотя бы одну цифру. Не применяйте пароли, используемые для других сайтов, и пароли, которые можно легко подобрать.

Нажимая на кнопку «Зарегистрироваться» вы соглашаетесь с [условиями использования](#)

[Зарегистрироваться](#)

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 294) 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.

### 2.4.3 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.

The screenshot shows the 'Blitz Profile' user interface. At the top, there is a navigation bar with the user's email 'ivanov@company.com' and a menu icon. Below this is a secondary navigation bar with three tabs: 'Основные данные' (Active), 'Безопасность', and 'События'. The main content area is titled 'Личный кабинет / Основные данные' and contains a section 'Данные учетной записи' (Account Data) with the following table:

Фамилия	<a href="#">Иванов</a>
Имя	<a href="#">Иван</a>
Электронная почта	<a href="mailto:ivanov@company.com">ivanov@company.com</a>
Мобильный телефон	<a href="#">+7 (000) 000-00-00</a>

Отображение данных пользователя определяется HTML-шаблоном. Шаблон представляет собой текстовый файл, который компилируется с помощью шаблонизатора Twirl<sup>50</sup>. 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 294). Рекомендуется использовать файлы сообщений при необходимости поддержки мультиязычности.

Примеры функций:

Листинг 3: Отображение атрибута email

```
@editAsText("email", "Электронная почта")
```

Листинг 4: Отображение атрибута phone\_number с возможностью его редактировать

```
@editAsText("phone_number", "Мобильный телефон", "Ошибка")
```

Листинг 5: Отображение булевого атрибута info с возможностью его редактировать

```
@editAsBoolean("info", "Подписка")
```

<sup>50</sup> <https://github.com/playframework/twirl>

Листинг 6: 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:

### Отображение массива

[Акции и бонусные программы](#)

[Новости компании](#)

[Дайджест событий за месяц](#)

### Редактирование массива

Акции и бонусные программы

Новости компании

Дайджест событий за месяц

### 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} ${given_name}` позволит отобразить фамилию и имя пользователя;
- URL для перехода после успешного выхода из Личного кабинета;
- период отображаемых пользователям событий аудита (в календарных месяцах от текущей даты);
- шаблон отображения геоданных в событиях (см. *База геоданных* (страница 324)). Шаблон можно составить из следующих элементов, содержащих сведения о стране, регионе, городе и координатах: `${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.

**Настройки**

**Шаблон приветствия**

Отображается в правом верхнем углу Личного кабинета. Используйте строки подстановки для формирования приветствия. Например, "Привет, \${name}"

**URL для перехода после успешного выхода**

URL, на который пользователь будет перенаправлен в случае инициирования процедуры выхода из приложения после успешного выхода

**Глубина просмотра аудита**

Кол-во полных календарных месяцев просмотра событий

**Шаблон отображения геоданных в событиях**

Отображается в событиях. Используйте строки подстановки для формирования приветствия. Например, "\${ip\_ctr}, \${ip\_st}, \${ip\_ct}"

---

**Доступные пользователям функции**

- Смена пароля
- Настройка контрольного вопроса
- Управление ключами безопасности
- Просмотр и привязка социальных сетей
- Просмотр устройств доступа
- Просмотр и отзыв разрешений
- Просмотр событий
- Привязка НОТР-генераторов
- Привязка ТОТР-генераторов
- Настройка входа по коду подтверждения
- Настройка push-аутентификации
- Привязка ключей безопасности

## 2.4.4 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.



### Восстановление доступа

---

**Поиск учетной записи**

Возможные атрибуты для поиска

Задайте список атрибутов пользователя, по которым будет осуществляться поиск пользователя

Атрибуты для сверки

Задайте список атрибутов пользователя, значения которых будут запрошены у пользователя для сверки

Проверять наличие пользователей, имеющих право менять пароль в найденной учетной записи

---

**Способы восстановления доступа**

Возможные контакты восстановления доступа

Задайте список атрибутов пользователя, соответствующих возможным контактам пользователя

Общее количество попыток

Общее число отправок кодов подтверждения и попыток ввода кода подтверждения, после которого способ аутентификации будет временно заблокирован

Время блокировки при превышении попыток, в мин.

В течение указанного времени способ аутентификации будет недоступен пользователю

**Дополнительные проверки для восстановления доступа**

Необходимость дополнительной проверки

Список методов

Задайте список методов, которые могут быть использованы для дополнительной проверки

---

**Операции после восстановления доступа**

Снимать блокировку по неактивности после восстановления доступа

### Тексты формы

После определения набора *атрибутов для сверки* (страница 183) необходимо задать соответствующие им тексты в форме восстановления доступа. Для этого используйте *стандартный алгоритм* (страница 294). 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.

Листинг 7: 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
```

## 2.5 Администрирование пользователей

Настоящий раздел посвящен администрированию пользователей в Blitz Identity Provider.

### 2.5.1 Управление учетными записями

В разделе Пользователи консоли управления администратор Blitz Identity Provider может осуществлять следующие операции:

- поиск учетных записей пользователей (страница 187);
- добавление учетной записи пользователя (страница 187);
- просмотр и редактирование атрибутов учетной записи пользователя (страница 189);
- сброс сессий пользователя, принудительная смена пароля и сброс блокировки метода (страница 200);
- изменение пароля учетной записи пользователя (страница 190);
- просмотр и отвязка учетных записей внешних поставщиков идентификации (страница 190);
- привязка устройств для проведения двухфакторной аутентификации (страница 190);
- просмотр групп, в которые включен пользователь, управление членством пользователя в группах (страница 192);
- просмотр, привязка, удаление ключей безопасности пользователя (страница 195);
- просмотр прав учетной записи пользователя, назначение и отзыв прав (страница 193);
- просмотр разрешений, выданных пользователем приложениям (страница 196);
- просмотр и удаление запомненных устройств (страница 195);
- удаление учетной записи пользователя.

Общий вид страницы управления данными пользователей представлен на рисунке.

Пользователи

Иванов

[Создать учетную запись пользователя...](#)

Учетные записи пользователей

389-ds: ВР.\*\*\*\*\*  
Иванов Иван Иванович, ivanov@company.com

Данные пользователя

sub	ВР.*****
family_name	Иванов
given_name	Иван
middle_name	Иванович
email	ivanov@company.com
phone_number	+7(000)0000000
locked	Нет

## 2.5.2 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.

## 2.5.3 Добавление пользователя

Для добавления новой учетной записи требуется нажать на ссылку «Создать учетную запись пользователя...». В открывшемся окне:

- указать хранилище, в котором 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» (\*)).

Пользователи

Иванов Найти

Укажите атрибуты пользователя.

Хранилище  internal  Idap

sub\*

family\_name

given\_name

middle\_name

email

phone\_number

locked Нет

Пароль

Создать Отмена

#### 2.5.4 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.

Пользователи

Иванов Найти

[Создать учетную запись пользователя...](#)

Учетные записи пользователей

389-ds: ВР-\*\*\*\*\*  
Иванов Иван Иванович, ivanov@company.com

Данные пользователя События безопасности

sub ВР-\*\*\*\*\*

family\_name Иванов

given\_name Иван

middle\_name Иванович

email ivanov@company.com

phone\_number +7(000)0000000

locked Нет

Сохранить

Смена пароля

Новый пароль   Сгенерировать пароль

Изменить

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.

### 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.

Генератор паролей на основе секрета (HOTP)

Серийный номер   
Серийный номер устройства генерации разовых паролей

Пароль 1

Пароль 2

Генератор паролей на основе времени (TOTP)


Название генератора

Алгоритм шифрования

Длина пароля   
Число символов, из которых будет состоять разовый пароль

Время обновления пароля   
Время (в секундах), в течение которого будет обновляться разовый пароль

Секрет   
Секрет закодирован в Base32 кодировке

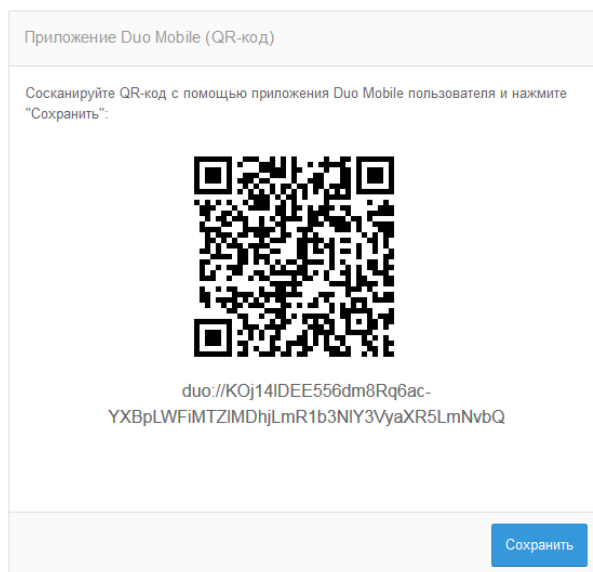


### Привязка Duo Mobile

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 будут отображены следующие данные:

- идентификатор группы;
- значения атрибутов группы.

Членство в группах	
Идентификатор	Данные группы
admins	name: Администраторы

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».

Членство в группах		
Идентификатор	Данные группы	
admins	name: Администраторы	
power_users		
Идентификатор	Имя	
power_users/roles	Power Users	




### 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.

Права субъектов в отношении пользователя			
Идентификатор	Имя	Право	
test-system	test-system	Назначать права	

[Назначить права](#)

Права пользователя в отношении объектов			
Идентификатор	Имя	Право	
_blitz_profile	custom.app.name._blitz_profile	Менять пароль	
_blitz_profile	custom.app.name._blitz_profile	rights.right.SUPPORT	
isergeev@domain.com	Сергеев Иван Петрович	rights.right.TEST	
_blitz_console	custom.app.name._blitz_console	Назначать права	

[Назначить права](#)

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.

Права субъектов в отношении пользователя

Права отсутствуют

Право: Назначать права

Тип субъекта: Приложение

Поиск субъекта: test-app Найти

Идентификатор	Имя
test-app	test-app

Добавить Отмена

Права пользователя в отношении объектов

Идентификатор	Имя	Право	
_blitz_profile	custom.app.name._blitz_profile	Менять пароль	<span>✕</span>
_blitz_profile	custom.app.name._blitz_profile	rights.right.SUPPORT	<span>✕</span>
isergeev@domain.com	Сергеев Иван Петрович	rights.right.TEST	<span>✕</span>
_blitz_console	custom.app.name._blitz_console	Назначать права	<span>✕</span>

Право: Менять пароль

Тип объекта: Пользователь

Поиск объекта: Иванов Найти







Идентификатор	Имя
6c02de61-909f-49d6-9bc4-4edb7d021c18	Иванов Александр
854436f6-af58-4a3f-8cb7-c2c441eb4a76	Иванов Сергей

Добавить Отмена

### 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 не 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 устройство 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.

Устройства пользователя				
Устройство	Браузер	Последний вход	Последний IP адрес	
 macOS 10.15.7	Chrome 100	04.05.2022 16:05	172.25.0.1	
 macOS 10.15.7	Yandex 22	27.04.2022 16:04	37.144.36.99	
 macOS 10.15.7	Chrome 100	25.04.2022 20:04	212.46.18.101	

### 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.

Ключи безопасности пользователя				
Имя ключа	Добавлен	Область применения	Последнее использование	
Face ID на iPhone	28.10.2022 16:10	Для входа Для подтверждения входа	01.11.2022 14:11	

[Добавить ключ](#)

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.

Укажите название нового ключа

Имя ключа

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.

Разрешения приложений

**oauth2\_test\_app** ✕

scopes: openid, profile

date: 09.09.2019

## 2.5.5 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",
```

(продолжается на следующей странице)

(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).

### Группы

Профиль	Атрибут	Значение	
grps	id	newusers	<a href="#">Найти</a>

[Создать группу...](#)

#### Группы пользователей

newusers

[Данные группы](#)

name

[Сохранить](#)

#### Члены группы

Идентификатор	Имя пользователя	
6647dc35-0c4f-4054-a7e7-fae41b011b4f	Петров Иван	✕
18539368-f59f-4ef1-8f34-3389272fa8bd	Васильев Дмитрий	✕

[Добавить пользователя...](#)

[Удалить группу](#)

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:

Члены группы

Идентификатор	Имя пользователя	
6647dc35-0c4f-4054-a7e7-fae41b011b4f	Петров Иван	
18539368-f59f-4ef1-8f34-3389272fa8bd	Васильев Дмитрий	

Иванов

Идентификатор	Имя пользователя
a9072d2b-9e89-45a5-9447-d0d126ba0332	Иванов Александр
854436f6-af58-4a3f-8cb7-c2c441eb4a76	Иванов Сергей

Профиль  grps

Идентификатор группы

name

## 2.5.6 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.

Справочник прав доступа

Задайте права доступа, которые могут быть назначены пользователям, группам или приложениям.

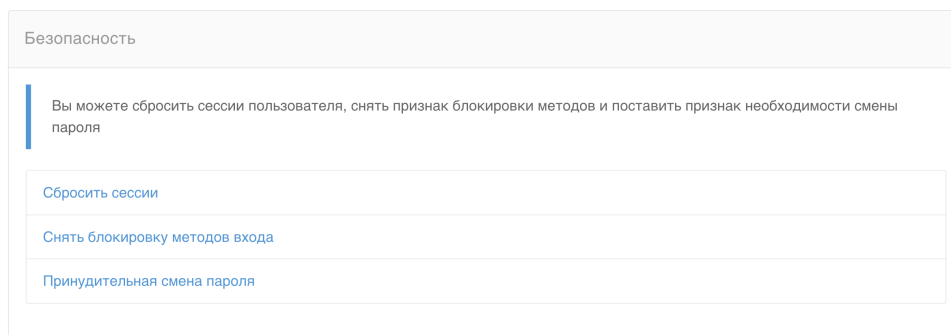
Название	Описание	
<input type="text" value="SUPPORT"/>	<input type="text"/>	
<input type="text" value="ADMIN"/>	<input type="text"/>	
<input type="text" value="TEST"/>	Проверочное право доступа	
<input type="text" value="USER"/>	<input type="text"/>	

[+ Добавить право доступа](#)

## 2.5.7 Безопасность

В блоке Безопасность доступны три кнопки:

- Сбросить сессии для сброса сессий пользователя
- Снять блокировку методов входа для снятия блокировки методов входа
- Принудительная смена пароля для принудительной смены пароля



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.

При снятии блокировки методов входа выполняются следующие действия:

- Если у пользователя был заблокирован один или несколько методов входа (например, если пользователь вводил неверные коды подтверждения и был превышен лимит на количество неверных кодов), то при снятии блокировки он сможет воспользоваться данным методом.

При принудительной смене пароля выполняются следующие действия:

- Устанавливается признак необходимости сменить пароль при следующем входе пользователя.

В случае необходимости смены пароля пользователя администратором обратитесь к разделу *Смена пароля* (страница 190).

## 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;
  - изменение ответа на контрольный вопрос.

## Параметры каналов оповещений

SMS-сообщения

Настройка сервиса отправки SMS-сообщений

Push-уведомления

Настройка сервиса отправки push-уведомлений

Email-сообщения

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	<input type="checkbox"/> Электронная почта
Смена пароля	<input checked="" type="checkbox"/> Электронная почта <input checked="" type="checkbox"/> SMS
Смена пароля в зависимой учетной записи	<input checked="" type="checkbox"/> Электронная почта
Восстановление доступа	<input checked="" type="checkbox"/> Электронная почта
Восстановление доступа в зависимой учетной записи	<input checked="" type="checkbox"/> Электронная почта
Привязка учетной записи социальной сети	<input checked="" type="checkbox"/> Электронная почта
Отвязывание учетной записи социальной сети	<input checked="" type="checkbox"/> Электронная почта
Настройка метода двухфакторной аутентификации	<input checked="" type="checkbox"/> Электронная почта
Изменение режима подтверждения входа	<input checked="" type="checkbox"/> Электронная почта
Получение права менять пароль в зависимой учетной записи	<input checked="" type="checkbox"/> Электронная почта
Предоставление права менять пароль	<input checked="" type="checkbox"/> Электронная почта
Отзыв права менять пароль в зависимой учетной записи	<input checked="" type="checkbox"/> Электронная почта
Отзыв предоставленного права менять пароль	<input checked="" type="checkbox"/> Электронная почта
Регистрация учетной записи	<input checked="" type="checkbox"/> Электронная почта
Добавление ключа безопасности	<input checked="" type="checkbox"/> Электронная почта
Удаление ключа безопасности	<input checked="" type="checkbox"/> Электронная почта
Изменение ответа на контрольный вопрос	<input checked="" type="checkbox"/> Электронная почта

## 2.6.1 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*».

Настройка сервиса отправки SMS

Протокол доставки: HTTP-GET  
Протокол доставки сообщений

При формировании URL и заголовков HTTP-запроса используйте строки подстановки:

- `${login}` - логин для доступа к сервису
- `${password}` - пароль для доступа к сервису
- `${message}` - сообщение (обязательный параметр)
- `${mobile}` - номер мобильного телефона (обязательный параметр)
- `${userId}` - идентификатор пользователя
- `${processId}` - идентификатор запроса (обязательный параметр)
- `${sessionId}` - идентификатор сессии

URL:

Логин:   
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:   
Регулярное выражение, определяющее успешную отправку сообщения. Например, `^OK.+`

Шаблон ответа при ошибке:   
Регулярное выражение, определяющее наличие ошибки при отправке сообщения. Например, `^ERROR.+`

Шаблон идентификатора успешной отправки:   
Регулярное выражение, извлекающее идентификатор успешной отправки. Например, `'gateway_id'~'\s*:\s*'[[\w]+]`

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;
- шаблон идентификатора успешной отправки. Задается в виде регулярного выражения, извлекающего идентификатор успешной отправки.

#### **Примечание**

Например, если ответ от СМС шлюза (smsc.ru) выглядит следующим образом:

```
OK - 1 SMS, ID - 3535
```

Тогда выражение для извлечения ID сообщения должно иметь следующий вид:

```
^.*ID - (.*)$
```

## 2.6.2 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    
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

Данные

Данные передаваемые в теле HTTP-запроса

Логин   
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   
Регулярное выражение, определяющее успешную отправку сообщения. Например, `^OK.+`

Шаблон ответа при ошибке   
Регулярное выражение, определяющее наличие ошибки при отправке сообщения. Например, `^ERROR.+`

### 2.6.3 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».

**Настройка сервера отправки Email-сообщений**

Хост

Порт

Отправитель   
email-адрес отправителя

Логин   
Логин учетной записи для соединения с SMTP-сервером  Совпадает с адресом отправителя

Пароль [Изменить значение](#)

Настройки

Расширенные параметры конфигурации SMTP-сервера. Каждый параметр описывается в отдельной строке. Название и значение параметра должны быть разделены символом `:`. Смотрите: [здесь](#).

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 чек-бокс);
- 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<sup>51</sup>.

<sup>51</sup> <https://javaee.github.io/javamail/docs/api/com/sun/mail/smtp/package-summary.html>

## Глава 3

# Доступ в приложения и сетевые службы

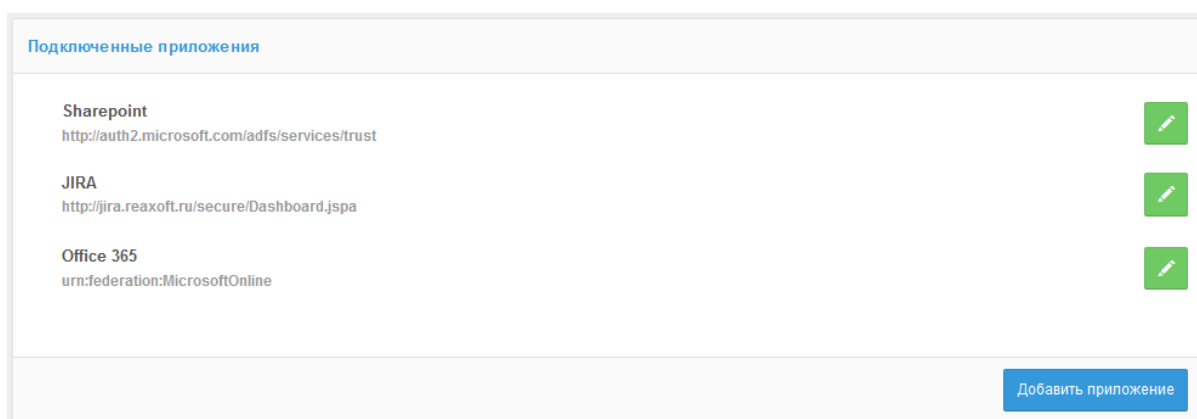
### 3.1 Регистрация приложений в Blitz Identity Provider

#### 3.1.1 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.



#### 3.1.2 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.

Новое приложение

Идентификатор (entityID, client_id)	<input type="text" value="Укажите идентификатор приложения"/>
	<small>Идентификатор приложения. Используется для идентификации приложения при доступе по протоколу SAML (соответствует entityID) и OAuth (соответствует client_id).</small>
Название	<input type="text" value="Укажите название приложения"/>
	<small>Отображаемое пользователям имя приложения. Используется только внутри Blitz Identity Provider.</small>
Домен	<input type="text" value="Укажите домен приложения"/>

## Шаг 2. Задание стартовой страницы приложения и выбор шаблона страницы входа

В поле **Стартовая страница приложения** рекомендуется задать ссылку на вход в приложение, иницирующую 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 283).

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 266).



**Параметры приложения**

**Идентификатор (entityID или client\_id)**   
Идентификатор приложения. Используется для идентификации приложения при доступе по протоколу SAML (соответствует entityID) и OAuth 2.0 (соответствует client\_id).

**Название**   
Отображаемое пользователям имя приложения. Используется только внутри Blitz Identity Provider

**Домен**   
Ссылка на стартовую страницу приложения, например, http://testdomain.ru/. При TLS-аутентификации приложения проверяется, что в сертификате приложения указан именно этот домен

**Стартовая страница приложения**   
Ссылка на стартовую страницу приложения, например, http://testdomain.ru/private. При входе по SAML используется как ссылка перехода в приложение, если открывать страницу входа из истории браузера

**Ключ шифрования идентификаторов**   
Если ключ задан, то идентификатор пользователя для приложения будет зашифрован с использованием данного ключа. Значение ключа можно выбрать из списка. Также можно назначить новый ключ, для этого введите его в строке поиска и нажмите Enter

**Шаблон страниц**   
Шаблон страниц определяет внешний вид страниц входа. Если шаблон не указан, то используется шаблон по умолчанию.

**Метки приложения**   
Позволяют пометить приложения определенными признаками. И использовать их при настройке логики работы с данным приложением, например, анализировать в процедуре входа

### 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.

**Контроль доступа**

Правила доступа к приложению не заданы и по умолчанию доступ к приложению не ограничен.  
 Для добавления правила воспользуйтесь [конфигуратором](#) или [создайте правило вручную](#).

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](#)

**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](#)

**Текущие утверждения в сессии пользователя**  
Любое утверждение может быть проверено на соответствие значению  
Пример: `age>18`

**Вхождение пользователя в группы**  
Проверяет, что пользователь входит в какую-то из указанных групп  
Пример: `contains(grps,GRP1,GRP2 ...)`

**Наличие у пользователя полномочий**  
Проверяет, что у пользователя присутствуют хотя бы одного указанное полномочие к объекту  
Объекты могут быть приложением (its), группой (grps) и пользователем (пусто)  
Пример проверки полномочий к приложению tech\_portal: `contains(rights.its.tech_portal,CHANGE_CONFIG,VIEW_CONFIG ...)`

Правило	Название	Описание	Активировано
			<input checked="" type="checkbox"/> <span style="margin-left: 20px;"><span style="color: red;">✖</span></span>

[Конфигуратор](#)    [+ Добавить правило вручную](#)

[Сохранить](#)

#### Шаг 4. Настройки протоколов подключения

Необходимо настроить один или несколько протоколов подключения приложения к Blitz Identity Provider.

**Протоколы**

SAML    **OAuth 2.0**    Simple    REST    RADIUS

---

Протокол OAuth 2.0 не сконфигурирован. [Сконфигурировать](#)

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 проху-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.

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.

## 3.2 Схемы работы технологий SSO

В данном разделе приведены схемы работы таких распространенных технологий единого входа, как OAuth 2.0 и SAML.

### 3.2.1 Подключение веб-приложения по OIDC

Взаимодействие веб-приложения с Blitz Identity Provider по OIDC включает в себя следующие этапы:

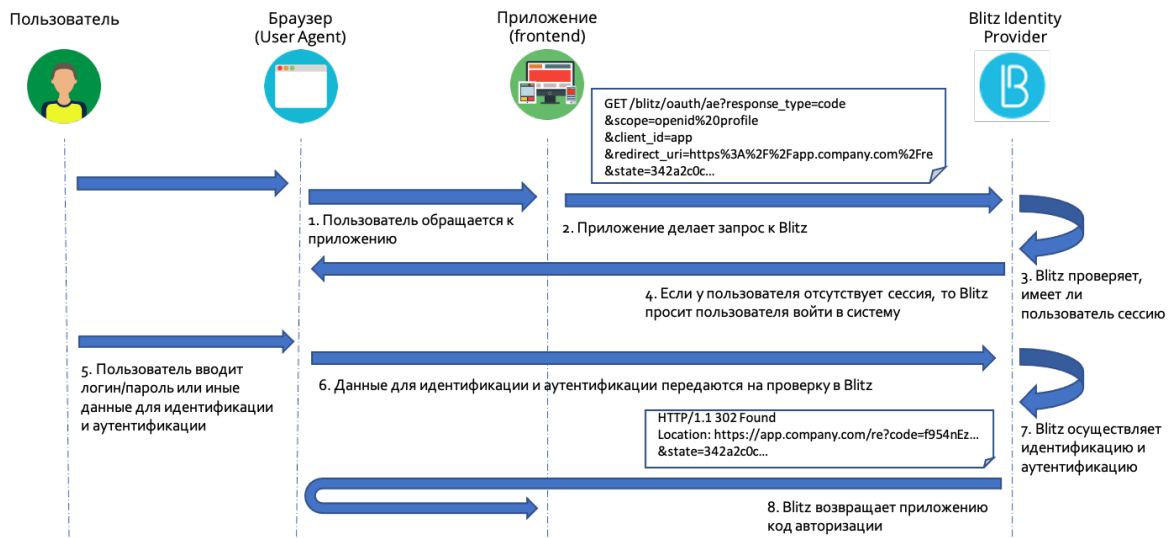
#### Примечание

Данный процесс совпадает с моделью авторизации приложения Authorization Code Grant, предусмотренной спецификацией OAuth 2.0.

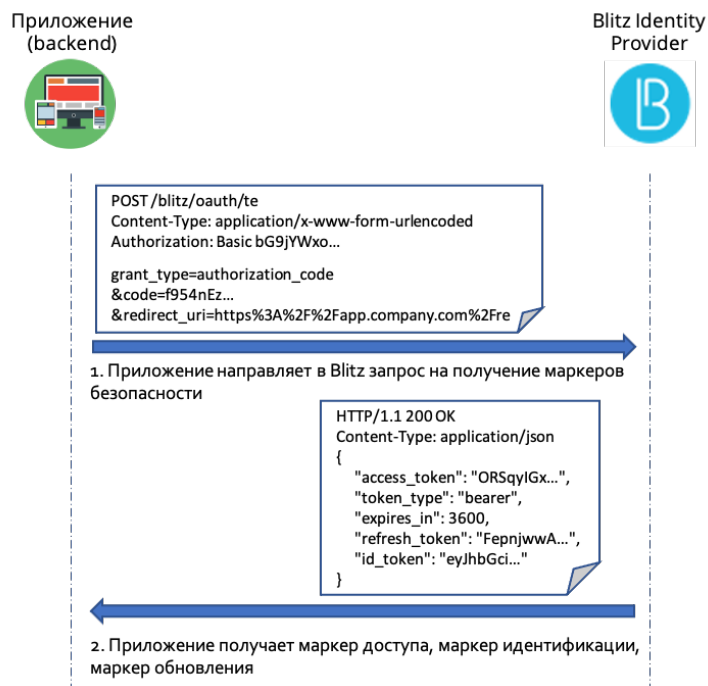
1. Приложение через веб-браузер отправляет запрос на идентификацию и аутентификацию пользователя в адрес Blitz Identity Provider.
2. Blitz Identity Provider проводит идентификацию/аутентифицирует пользователя.
3. Blitz Identity Provider получает согласие пользователя на передачу информации о нем в приложение (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.

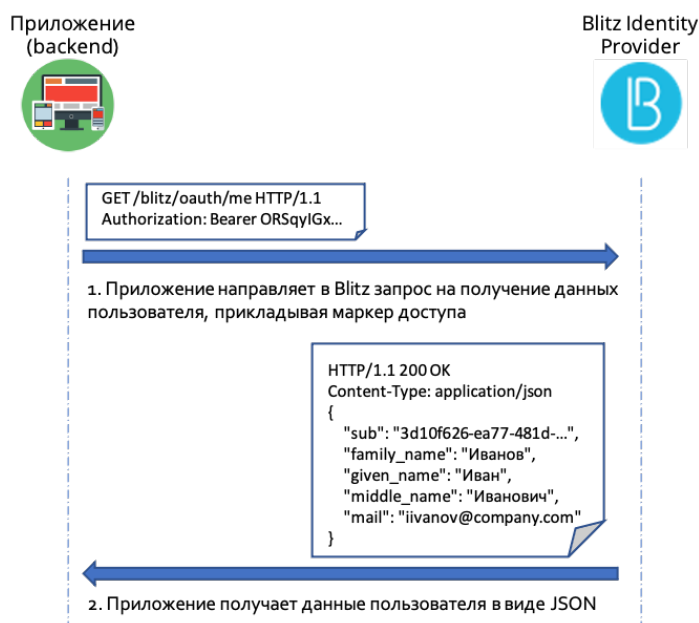
На рисунках представлены процессы получения кода авторизации, маркеров, данных пользователя.

## Получение кода авторизации:



## Получение маркеров безопасности:



**Получение данных пользователя:****3.2.2 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<sup>52</sup>,
- RFC 7592 OAuth 2.0 Dynamic Client Registration Management Protocol<sup>53</sup>.

Взаимодействие мобильного приложения с Blitz Identity Provider включает следующие этапы:

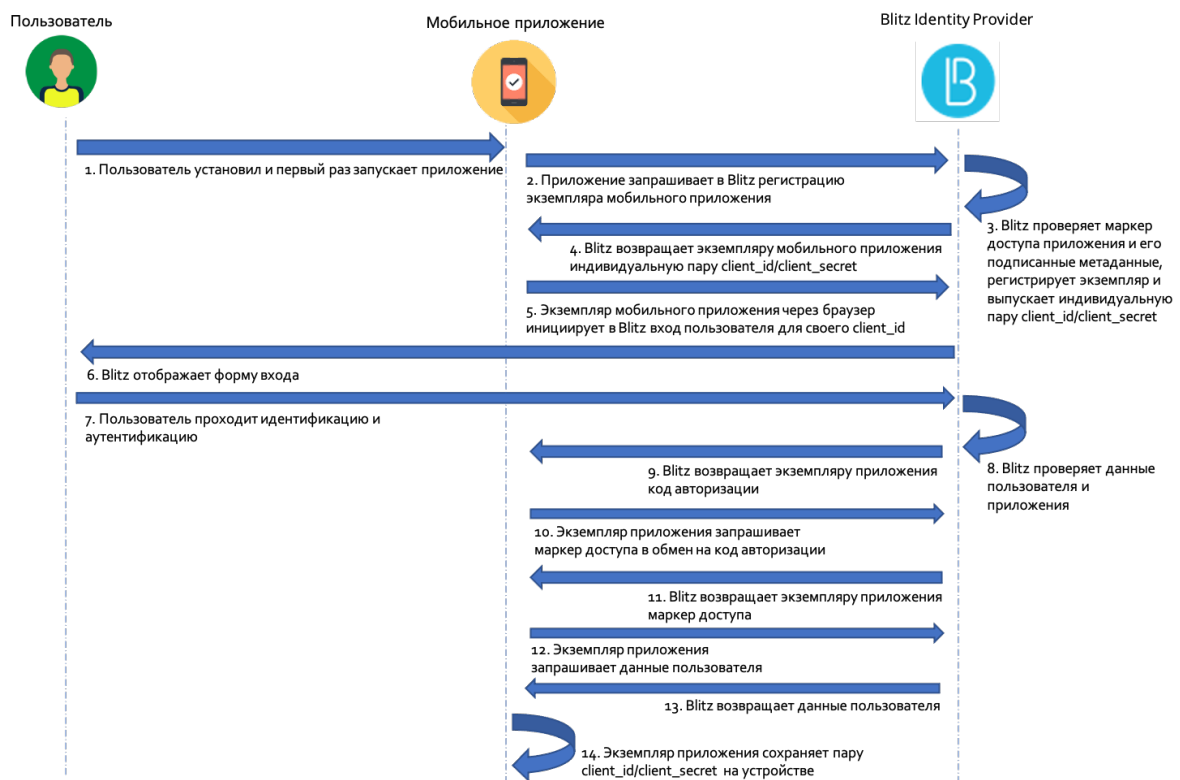
1. Динамическая регистрация в Blitz Identity Provider экземпляра мобильного приложения. Получение экземпляром приложения от Blitz Identity Provider уникальной пары `client_id / client_secret`.
2. Первичный вход пользователя в мобильное приложение с помощью Blitz Identity Provider. Установка пользователем ПИН-кода или Touch ID/Face ID. Сохранение на устройстве зашифрованной 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 зашифрованной 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.

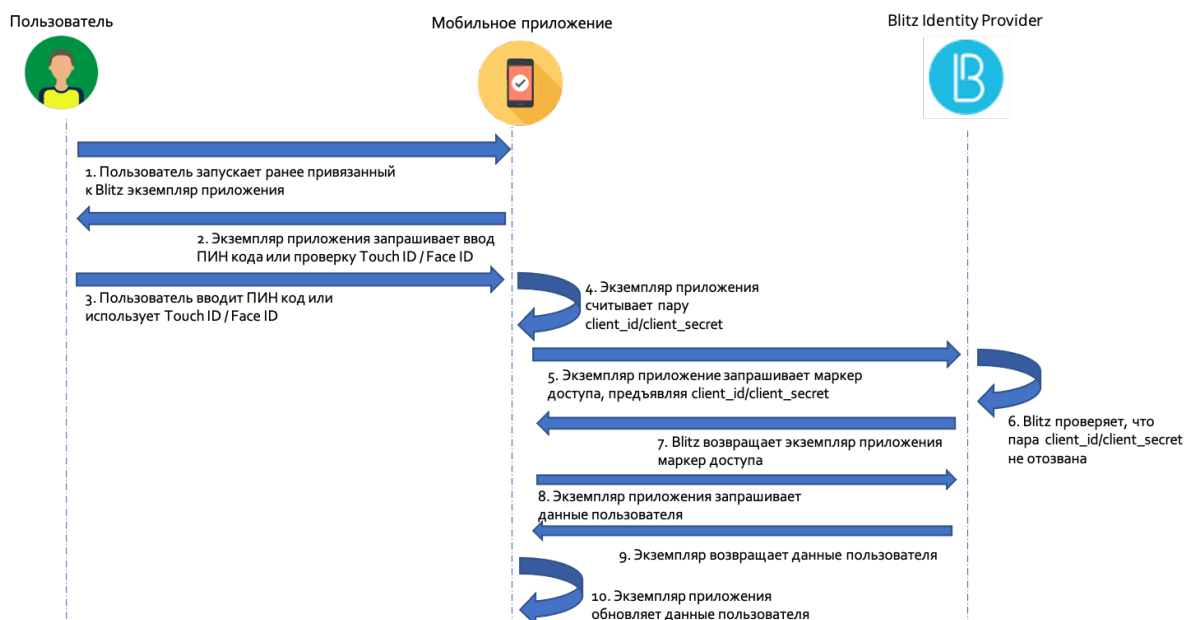
<sup>52</sup> <https://tools.ietf.org/html/rfc7591>

<sup>53</sup> <https://tools.ietf.org/html/rfc7592>

## Первый вход пользователя в мобильное приложение:



## Повторные входы пользователя в мобильное приложение:



## 3.2.3 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 че-

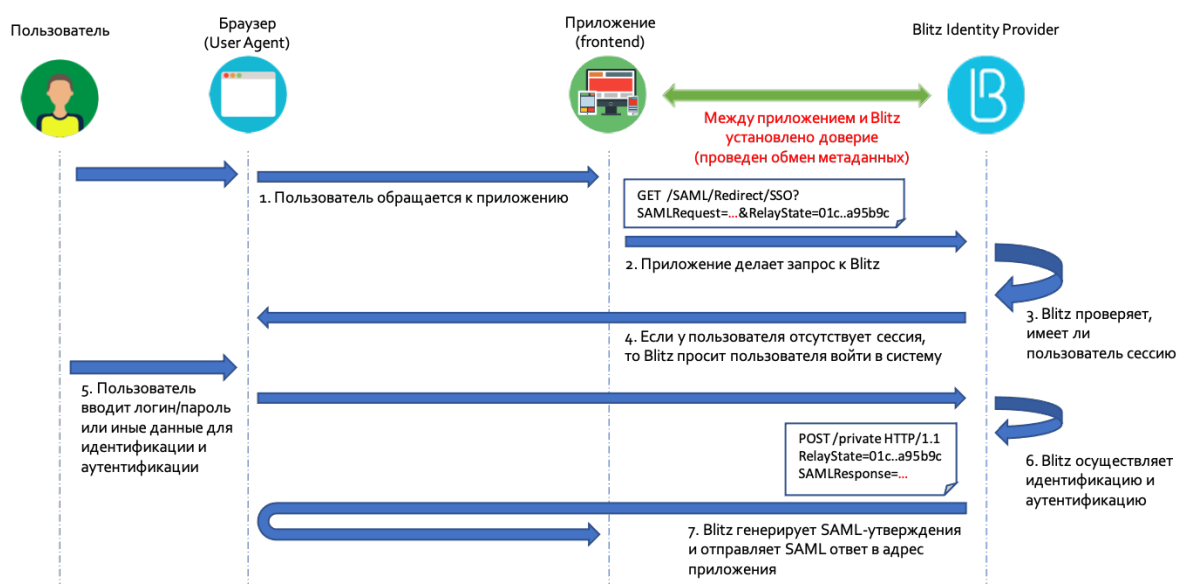
рез HTTP Redirect. Это означает, что запрос от приложения к Blitz Identity Provider осуществляется опосредованно, через браузер пользователя, и прямое сетевое взаимодействие между приложением и Blitz Identity Provider при использовании SAML не требуется.

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



## 3.3 Настройка SAML и WS-Federation

### 3.3.1 Подключение по SAML 1.0/1.1/2.0

При подключении приложения по SAML необходимо задать следующие настройки:

- *загрузить SAML-метаданные подключаемого приложения* (страница 218);
- 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 219) в разделе SAML консоли управления.

SAML профиль SAML 2.0 Web SSO Profile WS-Federation Passive Requestor Profile

Подписывать утверждения  ▼  
Правило подписи SAML-утверждений (Sign assertions)

Шифровать утверждения  ▼  
Правило шифрования SAML-утверждений (Encrypt assertions)

Шифровать идентификаторы (NameIds)  ▼  
Правило шифрования идентификаторов (Encrypt NameIds)

Включить передачу SAML-утверждений о пользователе в специальном блоке Attribute Statement

#### Атрибуты пользователя

Определите, какие атрибуты пользователя должны передаваться в приложения и с какими названиями

SAML-атрибут	Передавать	
<input type="text" value="urn:blitz:username"/> ▼	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="transientId"/> ▼	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="urn:blitz:surname"/> ▼	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="urn:blitz:mail"/> ▼	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="urn:blitz:name"/> ▼	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="urn:blitz:username"/> ▼	<input checked="" type="checkbox"/>	<input type="checkbox"/>

[+ Добавить](#)

[Сохранить](#)

### 3.3.2 Подключение по WS-Federation

При подключении приложения по WS-Federation необходимо задать следующие настройки:

- загрузить SAML-метаданные подключаемого приложения (страница 218);
- 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<sup>54</sup>, 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 219) в разделе SAML консоли управления.

<sup>54</sup> <http://www.ifap.ru/library/gost/86012001.pdf>

**SAML профиль** SAML 2.0 Web SSO Profile **WS-Federation Passive Requestor Profile**

Подписывать утверждения  ▼  
Правило подписи SAML-утверждений (Sign assertions)

Время жизни утверждений   
Время жизни утверждений в ответе в формате ISO 8601. Например, PT5M - 5 минут.

Включить передачу SAML-утверждений о пользователе в специальном блоке Attribute Statement

**Атрибуты пользователя**  
 Определите, какие атрибуты пользователя должны передаваться в приложения и с какими названиями

SAML-атрибут	Передавать	
<input type="text" value="urn:blitz:username"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="transientid"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="urn:blitz:surname"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="urn:blitz:mail"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="urn:blitz:name"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="urn:blitz:username"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

[+ Добавить](#)

[Сохранить](#)

### 3.3.3 Загрузка SAML-метаданных

Для загрузки SAML-метаданных приложения можно использовать любой из способов:

- Для загрузки готового XML-файла нажмите Открыть с файловой системы.

**Протоколы**

[SAML](#) [OAuth 2.0](#) [Simple](#) [REST](#) [RADIUS](#)

Метаданные [Открыть с файловой системы](#) [Сгенерировать метаданные](#)

```

1 <?xml version="1.0" encoding="UTF-8"?><md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" entityID="saml-dev">
2 <md:SPSSODescriptor AuthnRequestsSigned="true" WantAssertionsSigned="true" protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:
3
4 <md:KeyDescriptor use="signing">
5 <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
6 <ds:X509Data>
7 <ds:X509Certificate>
8 MIIID5zCCAs+gAwIBAgIUL60pUXeS92AF0ypIA59i1sM/NDcwDQYJKoZIhvcNAQEL
9 wgYIxCAzAJBgNVBAYTAnJlMQ8wDQYDVQQIDAZNb3Njb3cxZDZANBgNVBACMBk1v
10 vdzEQMA4GA1UECgwHUmVheG9mdEMMAoGA1UECwwDRGV2MREwDwYDVQDDAhz
11 sXZRldjEeMBwGCsqGS1b3D0EJARYPaW5mb0ByZWY4b2Z0LnJlMjB4XDTIyMDMy
12 0MjkyMVoXDTIzMDMyNTE0MjkyMVoYIxCzAJBgNVBAYTAnJlMQ8wDQYDVQOI
13 Nb3Njb3cxZDZANBgNVBACMBk1vZ2NvdzEQMA4GA1UECgwHUmVheG9mdEMMAoG
14 ECwwDRGV2MREwDwYDVQDDAhzYw1sX2RldjEeMBwGCsqGS1b3D0EJARYPaW5m
15 yZWY4b2Z0LnJlMjB4XDTIyMDMyNTE0MjkyMVoYIeMBwGCsqGS1b3D0EJARYPaW5m

```

- Для использования конструктора метаданных нажмите Сгенерировать метаданные. Введите следующие данные:
  - URL сервиса обработчика утверждений (AssertionConsumerService),
  - URL сервиса единого логгута (SingleLogoutService),
  - Сертификат подписи,
  - Сертификат шифрования.

Протоколы

SAML OAuth 2.0 Simple REST RADIUS

Задайте параметры для формирования метаданных

URL сервиса обработчика утверждений (AssertionConsumerService)

URL сервиса единого логгута (SingleLogoutService)

Сертификат подписи

Сертификат шифрования

Сгенерировать Отменить

Нажмите Сгенерировать. В результате файл метаданных будет автоматически сгенерирован на основании введенных данных.

### 3.3.4 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).

Атрибуты

Определите, какие атрибуты пользователя из хранилища могут передаваться в SAML-приложения (поставщики услуг) в виде SAML-атрибутов

Поиск... Найти

UserID
UserMail
eduPersonPrincipalName
urn:blitz:email
urn:blitz:upn
urn:blitz:givenName
urn:blitz:sn
urn:blitz:role

[+ Добавить новый SAML-атрибут](#)

Свойства SAML-атрибута

Название:

Источник:

[Сохранить](#)

Кодировщик

Тип:

Название:

Короткое название:

Формат имени:

[Удалить](#)

Кодировщик

Тип:

Название:

Формат имени:

[Удалить](#)

[+ Добавить кодировщик](#)

[Удалить SAML-атрибут](#)

## 3.4 Настройка OAuth 2.0 и OpenID Connect 1.0

### 3.4.1 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 проху 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 137) таким образом, что будут сохраняться маркеры доступа пользователя от внешних поставщиков идентификации. Если приложение будет получать по REST API 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;
- при необходимости задать отображение экрана согласия на доступ к данным о себе. Если выбрано *отображать* экран согласия, то при первом входе пользователя в систему будет отображена страница согласия на предоставление данных этой системе. Если выбрано *не отображать* экран согласия, то при первом входе пользователя в систему не будет отображена страница согласия на предоставление данных этой системе. Если выбрана настройка по умолчанию, тогда берется глобальная настройка. Также имеется возможность использовать *внешнее приложение* (страница 347) для отображения экрана согласия.
- 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. Также существует метод аутентификации `none`. Он применяется для динамических клиентов, таких как мобильные приложения, которые не используют параметр `client_secret` для аутентификации. Данный метод не предназначен для веб-приложений, где возможно безопасное хранение `client_secret`. Использование метода `none` регламентируется спецификацией RFC 7591<sup>55</sup>;
- 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`;
- отметить при необходимости опцию «*Добавлять в маркер доступа параметр rights с правами пользователя на данное приложение*».
- 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 актуаль-

<sup>55</sup> <https://datatracker.ietf.org/doc/html/rfc7591>

ные 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 227));
- выбрать формат маркера доступа – можно выбрать `opaque` или `JWT`. Если параметр не задан, то берется из общих настроек из раздела «*OAuth 2.0*».

### Настройки взаимодействия с приложением

**Секрет (client\_secret)**

Секретный ключ подключаемого приложения (client\_secret). Если указан, то именно этот секрет должен использоваться подключаемым приложением при обращении к Blitz Identity Provider

**Дополнительный секрет (client\_secret)**

Дополнительный секретный ключ подключаемого приложения (client\_secret). Если указан, то может использоваться в качестве альтернативы к основному секрету

**Предопределенная ссылка возврата (redirect\_uri)**

URL, на который по умолчанию будет переадресован пользователь после прохождения авторизации (redirect\_uri)

**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.

**Разрешения по умолчанию**

Разрешения (scope), которые будут по умолчанию выданы приложению после авторизации. Если значения по умолчанию не указаны, то в запросе необходимо явно прописать требуемые разрешения.

**Отображать экран согласия на доступ к данным о себе**

Обязательное использование Proof Key for Code Exchange (RFC 7636) для Authorization code grant type

**Метод аутентификации при обращении к сервису выдачи маркеров**

Указанный метод аутентификации должен использоваться при обращении к сервису выдачи маркеров (token endpoint). При пустом значении доступны все методы

**Допустимые grant type**

Список grant type, которые будут доступны приложению. При пустом списке доступны все grant type

**Допустимые response type**

Список response type, которые будут доступны экземпляру приложения при обращении к URL авторизации (authorization endpoint). При пустом списке доступны все response type.

Добавлять в маркер доступа параметр rights с правами пользователя на данное приложение

**Время жизни маркера доступа (access\_token)**

Задается количество секунд, через которое код доступа станет недействительным. Если не задан, то берется из общих настроек.

**Режим выдачи маркеров доступа по умолчанию**

Режим выдачи маркеров доступа (access\_token), если явно не указан в запросе. При online-режиме не выдается маркер обновления (refresh\_token)

**Время жизни маркера обновления**

Задается количество секунд через которое код обновления будет не действителен. Если не задан, то берется из общих настроек.

**Добавляемые в маркер идентификации (id\_token) утверждения**

Дополнительные утверждения (claim), которые будут добавлены в маркер идентификации (id\_token).

**Формат маркера доступа**

Задаёт формат маркера доступа для данного приложения. Если формат не указан, то формат берется из общей настройки OAuth 2.0

При использовании в приложении функции логута<sup>56</sup> в блоке «Выход из приложения» необходимо задать следующие настройки:

- указать префиксы ссылок возврата при выходе. Необходимо перечислить префиксы допустимых URL

<sup>56</sup> [https://openid.net/specs/openid-connect-rpinitiated-1\\_0.html#RPLogout](https://openid.net/specs/openid-connect-rpinitiated-1_0.html#RPLogout)

страниц перенаправления пользователя после инициирования приложением логута. Допустимо задать один или несколько префиксов ссылок возврата;

- предопределенная ссылка возврата при выходе – ссылка, на которую будет перенаправлен пользователь после логута из приложения, если в параметрах вызова логута 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 логута пользователя;
- отметить при необходимости опцию «Добавлять идентификатор сессии и эмитента в ссылку очистки сессии в браузере (`Front channel`)» – в этом случае в браузерный обработчик логута 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 логута 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 логута пользователя, будет передан `logout_token`, содержащий идентификатор сессии пользователя (`sid`).

Выход из приложения

Префиксы ссылок возврата при выходе

Для добавления нового URL введите его и нажмите Enter

Список URL используется для проверки ссылок возврата (`post_logout_redirect_uri`). Если в запросе на выход указана ссылка возврата и она не соответствует ни одному из указанных префиксов, то в выходе будет отказано

Предопределенная ссылка возврата при выходе

URL, на который по умолчанию будет переадресован пользователь после успешного выхода из системы

Не показывать пользователю экран с подтверждением выхода из системы

Ссылка для очистки сессии пользователя в браузере (`Front channel`)

URL, на который будет направлен браузер для очистки сессионной информации

Добавлять идентификатор сессии и эмитента в ссылку очистки сессии в браузере (`Front channel`)

Ссылка для очистки сессии пользователя в приложении (`Back channel`)

URL, на который будет выполнен запрос для очистки сессионной информации

Добавлять идентификатор сессии и эмитента в ссылку очистки сессии в приложении (`Back channel`)

При использовании приложением авторизации по спецификации Device Authorization Grant<sup>57</sup> (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;

<sup>57</sup> <https://tools.ietf.org/html/rfc8628>



- отметить при необходимости опцию «Добавлять в URL пользовательский код». В этом случае Blitz Identity Provider при авторизации будет возвращать не только ссылку на страницу 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`).

Device Authorization Grant

Формат пользовательского кода

Формат указывается в виде шаблона на основе регулярного выражения, по которому происходит генерация пользовательского кода для привязки устройства. Например: [0-9]{2,3}-[0-9]{2,3}

Время жизни пользовательского кода

Задается количество секунд через которое пользовательский код будет не действителен. Если не задан, то берется из общих настроек.

Ссылка на страницу ввода пользовательского кода

Если ссылка не задана, то она формируется автоматически

Добавлять в URL пользовательский код

[Сохранить](#)

### 3.4.2 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.

Свойства

URL с метаданными Blitz Identity Provider

При подключении приложений по OpenID Connect в настройках этих приложений может потребоваться указать эту ссылку на файл с метаданными поставщика идентификации

URL для авторизации

На данный URL (authorization endpoint) должен быть направлен запрос на проведение авторизации пользователя

URL для получения и обновления маркера

На данный URL (token endpoint) должен быть направлен запрос на получение или обновление маркера доступа

Время жизни маркера доступа, сек

Формат маркера доступа

Время жизни маркера обновления, сек

Аутентификация систем-клиентов с использованием Proxу TLS. Для аутентификации систем по Proxу TLS должно быть настроено взаимодействие через прокси-сервер и обеспечено установление двустороннего TLS-соединения. В поле Common Name (CN) сертификата системы должен быть указан домен системы

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<sup>58</sup>). Разработчики приложений могут не прописывать все указанные ниже URL в конфигурации своего приложения, а использовать в настоящих единую ссылку на эти метаданные;

<sup>58</sup> <https://tools.ietf.org/html/draft-ietf-oauth-discovery-10>

- «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`).

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 Proxu 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.

Device Authorization Grant

Время жизни пользовательского кода, сек	<input style="width: 95%;" type="text" value="300"/>
Разрешенный интервал опроса статуса кода привязки устройства, сек	<input style="width: 95%;" type="text" value="5"/>

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).

Настройка scopes

Укажите разрешения (scope), которые могут быть запрошены системами (приложениями). При необходимости укажите, какие атрибуты пользователя из хранилища могут быть получены по этим разрешениям

Название разрешения	Описание	Атрибуты пользователя	Системный	
openid	Информация, позволяющая провести идентификацию и аутентификацию		<input type="checkbox"/>	
profile	Основные данные профиля пользователя	* sub * family_name * middle_name * given_name	<input type="checkbox"/>	
email	Электронная почта	* email	<input type="checkbox"/>	
phone_number	Номер телефона	* phone_number	<input type="checkbox"/>	

[+ Добавить scope](#)

[Сохранить](#)

#### **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 137) таким образом, что будут сохраняться маркеры доступа пользователя от внешних поставщиков идентификации. Если подключенные по OAuth 2.0 приложения будут получать по REST API 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.

### 3.4.3 Добавление атрибутов в маркер идентификации

Чтобы иметь возможность определять сессионные утверждения в процедуре входа, соответствующие утверждения также должны быть определены в конфигурационном 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 маркере 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 значений 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, SUBJECT.1.2.643.100.5 позволяет обратиться к атрибуту с O I 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"
  }
]
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

    {
      "name" : "esia_inn",
      "value" : "inn"
    }
    {
      "name" : "esia_passport",
      "value" : "passport"
    }
    {
      "name" : "esia_birthPlace",
      "value" : "birthPlace"
    }
    {
      "name" : "esia_email",
      "value" : "email"
    }
    {
      "name" : "esia_mobile",
      "value" : "mobile"
    }
  ],

```

### 3.4.4 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 225));
- в настройках OAuth 2.0 для данного приложения перейти на закладку «*Динамические клиенты*».

Протоколы

SAML **OAuth 2.0** Simple REST RADIUS

Для корректной работы пропишите эти ссылки в настройках приложения, в которое будет осуществляться вход

URL для авторизации `/blitz/oauth/ae`  
 На данный URL (authorization endpoint) должен быть направлен запрос на проведение авторизации пользователя

URL для получения и обновления маркера `/blitz/oauth/te`  
 На данный URL (token endpoint) должен быть направлен запрос на получение или обновление маркера доступа

Статический клиент **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`).
- при необходимости добавить дополнительные утверждения для метаданных приложения (`software_statement`).

Создать первичные маркеры для приложения. Первичные маркеры используются для авторизации экземпляров 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 экземпляром 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)

Используется для регистрации динамических клиентов

Допустимые к прямой передаче утверждения

Эти утверждения допускается указывать в запросе на регистрацию инстанса приложения

Добавьте дополнительные утверждения для метаданных приложения (software\_statement)

Утверждение	Значение	Описание утверждения

Указать утверждения для Blitz Key + Добавить утверждение

[Изменить](#)

## Подписание метаданных приложения

Подпишите метаданные приложения (software\_statement). Эти метаданные передаются в качестве утверждения в запросе на регистрацию инстанса приложения

Версия приложения

Версия приложения в метаданных должна соответствовать версии в первичном маркере

Префиксы ссылок возврата

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

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 инстансу приложения

Допустимые значения response type

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	
7w6TCR9oI9IsZUVT_dZEE6SiHVFmKXWRAShpH-wz6RRRl4brtrhp_QIA7PnhhulaXOhHelJg6B_n2qtwml-g	21.10.2024 14:36:48	1	<a href="#">✕</a>

Версия ПО

[Выпустить](#)

## 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 логат в подключенном по Simple приложении в случае единого логата в Blitz Identity Provider.
  - URL для перехода после успешного выхода – указывает, какой адрес должен вызвать Blitz Identity Provider для перенаправления пользователя после успешного логата, 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([encodeURIComponent('error'), encodeURIComponent(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.

SSL

Селектор формы   
 CSS селектор используется для определения расположения формы входа на странице

Селектор поля с логином   
 CSS селектор используется для определения поля ввода логина

URL выхода по умолчанию   
 URL, на который необходимо перенаправить пользователя для выхода из приложения по умолчанию

URL для перехода после успешного выхода   
 URL, на который пользователь будет перенаправлен в случае инициирования процедуры выхода из приложения после успешного выхода

JavaScript [Изменить](#)

## 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;
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

#ssl_certificate          /etc/nginx/cert/<path to SSL certificate>.pem;
#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 / {

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

    proxy_pass http://oc-web;
  }
}

```

### 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 207) 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.

The screenshot shows the configuration page for the REST API protocol. At the top, there are tabs for 'SAML', 'OAuth 2.0', 'Simple', and 'REST', with 'REST' being the active one. Below the tabs, there are two main sections:

- The first section is titled 'Для аутентификации приложения при доступе к программному интерфейсу Blitz Identity Provider с помощью Basic-аутентификации задайте пароль. В качестве логина используйте идентификатор приложения'. It contains a 'Пароль' (Password) field with a masked input and a toggle for visibility.
- The second section is titled 'Для использования TLS-аутентификации укажите допустимые значения CN из сертификата приложения. Вы можете указать несколько значений'. It contains a 'Допустимые CN' (Allowed CNs) text input field. Below the field, there is a small instruction: 'Укажите Common Name (CN) и нажмите Enter'. At the bottom of this section is a 'Сохранить' (Save) button.

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.

### 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<sup>59</sup> — 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.

<sup>59</sup> <https://datatracker.ietf.org/doc/html/rfc2865>

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 (в соответствии с *RADIUS-процедурой* (страница 272)): вход по логину и паролю, подтверждение по коду из SMS, PUSH, TOTP, HOTP, email или через Личный кабинет 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.

### 3.7.1 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.

## Конфигурация RADIUS сервера

Общие настройки

Сегменты сети

Процедуры обработки запросов

## Настройки сервера

Статус 

Сетевой адрес привязки

0.0.0.0

Обрабатываются запросы только с указанного адреса. **0.0.0.0** - обработка запросов со всех доступных сетевых интерфейсов

Сетевой порт

На данный порт 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

300

Определяет сколько секунд дается пользователю для прохождения второго фактора. Данное время должно быть согласовано с RADIUS-клиентом за счет корректной настройки timeout

Отменить изменения

Сохранить

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` и `CallingStationId` из RADIUS-запроса.

**⚠ 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 241).
- **Приложение по умолчанию:** выберите приложение, с которым будет ассоциироваться запрос из данной подсети. Если приложений несколько, оно будет выступать приложением по умолчанию.
- **Соответствие `NasId/CallingStationId` и приложений:** если предполагается, что из одной подсети запрашивать аутентификацию будет несколько приложений, задайте `NasId` или `CallingStationId`, по которым сервер RADIUS будет их идентифицировать.

Конфигурация RADIUS сервера

Общие настройки **Сегменты сети** Процедуры обработки запросов

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 или CallingStationId. Более узкие подсети имеют приоритет page.radius.segments.add=Создать новый сегмент

vpn-net: .1.1/24  
local-net: 0.183/32

[Добавить сегмент](#)

Параметры сегмента сети

Имя: vpn-net

Подсеть: .1.1/24

Общий ключ: .....

Приложение по умолчанию: RADIUS APP VPN

Соответствия NasId и приложений не сконфигурировано  
[добавить](#)

Соответствия CallingStationId и приложений не сконфигурировано  
[добавить](#)

[Удалить](#) [Отменить изменения](#) [Сохранить](#)

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.

**Конфигурация RADIUS сервера**

Общие настройки    Сегменты сети    **Процедуры обработки запросов**

---

Процедура обработки запросов

Статус

Идентификатор процедуры:   
Java класс, описывающий процедуру обработки запросов, должен иметь такое же название.

Описание:

Приложения:   
Список приложений, для которых будет применяться данная процедура.

3. Нажмите Сохранить.

4. Введите исходный код процедуры. Для управления процессом обработки RADIUS запросов необходимо написать на языке Java класс, реализующий интерфейс `RadiusFlow`. Подробную информацию по написанию процедуры с примерами вы найдете [здесь](#) (страница 272).

5. Для компиляции нажмите Сохранить.

### 3.7.2 Шаг 2. Настройка приложения

Для настройки приложения выполните следующие действия:

1. В консоли управления перейдите в раздел Приложения. *Создайте* (страница 207) приложение с базовыми настройками.
  - Идентификатор (`entityID` или `client_id`),
  - Название,
  - Домен: домен сетевой службы.

**Параметры приложения**

Идентификатор (`entityID` или `client_id`):   
Идентификатор приложения. Используется для идентификации приложения при доступе по протоколу SAML (соответствует `entityID`) и OAuth 2.0 (соответствует `client_id`).

Название:   
Отображаемое пользователям имя приложения. Используется только внутри Blitz Identity Provider

Домен:   
Ссылка на стартовую страницу приложения, например, `http://testdomain.ru/`. При TLS-аутентификации приложения проверяется, что в сертификате приложения указан именно этот домен

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 237) на сервере RADIUS процедуры.

**⚠ Внимание**

Внимательно настраивайте *интеграцию* (страница 241) 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.

Протоколы

SAML OAuth 2.0 Simple REST **RADIUS**

Блок индивидуальных настроек приложения для RADIUS протокола

Пароль проверяется приложением самостоятельно

Время ожидания второго фактора

Определяет сколько секунд дается пользователю для прохождения второго фактора. Данное время должно быть согласовано с RADIUS-клиентом за счет корректной настройки timeout. По умолчанию значение берется из общей настройки RADIUS протокола

Процедура обработки RadTest2

Если идентификатор не указан, то используется базовая процедура обработки

Сохранить

Нажмите Сохранить.

### 3.7.3 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 237), соответствующего приложению (сетевой службе) на сервере RADIUS. По данному ключу сервер будет опознавать сетевую службу и запускать выбранную для нее процедуру обработки доступа.
- `NasId`, `CallingStationId` (при необходимости).
- Время ожидания ответа от сервера RADIUS, соответствующее установленному на сервере времени ожидания второго фактора.

## Глава 4




# Кастомизация работы с помощью программирования на Java

### 4.1 Процедуры входа и их создание

#### 4.1.1 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.

Настроенные процедуры входа			
Идентификатор	Приложения	Описание	Статус
AccessByAttribute v4		If the user attribute "appList" (as an array) contains entityID (or client_id) of the application, access will be granted	Не активирована 
FFmethods v19		Limited list of first factor methods for application	Не активирована 
Require2ndFactor v5		This procedures enables 2nd factor for the application	Не активирована 

[Создать новую процедуру входа](#)

#### 4.1.2 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.

Создание новой процедуры входа

Идентификатор процесса

Идентификатор процесса должен быть корректным Java-идентификатором. Java-класс, описывающий процесс входа, будет иметь такое же название.

Описание

При необходимости укажите комментарий

Приложения

Список приложений, для которых будет применяться данная процедура входа. Если приложения не указаны, то процедура будет считаться глобальной и будет применяться для всех приложений, для которых не определена процедура. Только одна глобальная процедура может быть активирована. Не должно быть коллизий при определении процедуры входа для отдельного приложения.

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 удастся скомпилировать. Иными словами, если при редактировании активированной процедуры была выявлена ошибка, то кнопка «Сохранить» работать не будет, а при перезагрузке страницы изменения будут утеряны.

## Исходный код процедуры

Для успешной работы процедуры аутентификации необходимо написать на языке `Java` класс, реализующий интерфейс `Strategy`. Название класса должно совпадать с идентификатором процесса (`SecondFAforAll`). Класс должен иметь публичный `default` конструктор. В целях безопасности загрузка класса осуществляется отдельный `class loader` с ограниченным списком `imports`. Вся контекстная информация о пользователе, о текущем состоянии процедуры аутентификации и т.д. доступна в объекте `Context`.

Посмотреть интерфейс Strategy ▾

Посмотреть разрешенные imports ▾

Посмотреть описание Context ▾

Загрузить Blitz Development Kit

```

1 package com.identityblitz.idp.flow.dynamic;
2
3 import java.lang.*;
4 import java.util.*;
5 import java.math.*;
6 import org.slf4j.LoggerFactory;
7 import org.slf4j.Logger;
8 import com.identityblitz.idp.login.authn.flow.Context;
9 import com.identityblitz.idp.login.authn.flow.Strategy;
10 import com.identityblitz.idp.login.authn.flow.StrategyState;
11 import com.identityblitz.idp.login.authn.flow.StrategyBeginState;
12 import com.identityblitz.idp.login.authn.flow.LCookie;
13 import com.identityblitz.idp.flow.common.api.*;
14 import com.identityblitz.idp.flow.dynamic.*;
15 import java.lang.invoke.LambdaMetafactory;
16 import java.util.function.Consumer;
17
18 import static com.identityblitz.idp.login.authn.flow.StrategyState.*;
19
20 public class SecondFAforAll implements Strategy {
21
22     private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.flow.dynamic");
23
24     @Override public StrategyBeginState begin(final Context ctx) {
25         return StrategyState.LOGOUT_THEN_MORE(new String[]{});
26     }
27
28     @Override public StrategyState next(final Context ctx) {
29         if(ctx.justCompletedFactor() == 2)
30             return StrategyState.ENOUGH();
31         else
32             return StrategyState.MORE(new String[]{});
33     }
34 }

```

Компилировать

Сохранить

## 4.2 Готовые процедуры входа

В поставку входят несколько готовых процедур, которые могут быть при необходимости изменены:

- *принудительная двухфакторная аутентификация в приложение* (страница 245) (`Require2ndFactor`);
- *ограничение перечня доступных методов первого фактора при входе в приложение* (страница 245) (`FFmethods`);
- *предоставление доступа к приложению только при определенном значении атрибута* (страница 246) (`AccessByAttribute`);
- *запрет входа в приложение после истечения срока действия учетной записи* (страница 248) (`AccountExpiresCheck`);
- *разрешение входа в приложение только из определенных сетей* (страница 249) (`AllowedIPs`);
- *запрет работы в нескольких одновременных сессиях* (страница 250) (`RestrictSessions`);
- *сохранение в утверждениях `claims` перечня групп пользователя* (страница 250) (`AddGroupsToToken`);
- *отображение пользователю объявления при входе* (страница 251) (`InfoPipe`);
- *запрос ввода пользователем атрибута или актуализации телефона и email* (страница 253) (`PipeAttrActAdd`);
- *запрос ввода пользователем контрольного вопроса, если он не задан в учетной записи* (страница 256) (`PipeSecQuestion`);

- регистрация ключа безопасности *WebAuthn*, *Passkey*, *FIDO2* при входе (страница 257) (*PipeWebAuthn*).
- отображение пользователю списка выбора значений при входе (страница 259) (*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"));
```

#### 4.2.1 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 работает.

```
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[]{});
        }
    }

    @Override public StrategyState next(final Context ctx) {
        if (ctx.justCompletedFactor() == 1)
            return StrategyState.MORE(new String[]{});
        else
            return StrategyState.ENOUGH();
    }
}
```

#### 4.2.2 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[]{});
        }
    }
}
```

### 4.2.3 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)
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

        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[]{});
    }
}
}
}

```

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();
}

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

else {
    if(reqFactor == ctx.justCompletedFactor())
        return StrategyState.ENOUGH();
    else
        return StrategyState.MORE(new String[]{});
}
}

```

#### 4.2.4 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[]{});
    }
}

@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);
    }
}
}

```



## 4.2.5 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()) {
            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;
    }
}
```

## 4.2.6 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[]{});
    }
}
```

## 4.2.7 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 227)).

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);
        }
    }
}
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

    } 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[]{});
}
}

```

#### 4.2.8 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 253) в конфигурационном файле тип уведомления;
- *настроить* (страница 301) 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=" +
↪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"
    }
  ]
}
```

### 4.2.9 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 что 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 301).

```
public class PipeAttrActAdd implements Strategy {
    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

↪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");
                };
                logger.debug("User has no { } or a non-actualized { }, 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";

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

        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-actualized {}, 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;
    long now = Instant.now().getEpochSecond();
    if (ctx.user().userProps().numProp("pipes.act."+attrName+".skippedOn") != ↪
↪null) {
        skpTime = ctx.user().userProps().numProp("pipes.act."+attrName+".

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

↪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);
        }
    }
}
}
}

```

#### 4.2.10 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();
            else

```

(продолжается на следующей странице)



(продолжение с предыдущей страницы)

```

        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();
}
}
}

```

#### 4.2.11 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 что 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();
    }
}
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

        if (keys != null) {
            keyList = keys.filter(k -> deviceType.equals(k.addedOnUA().
↵deviceType()))
                .filter(k -> os.equals(k.addedOnUA().osName())).list();
        }
        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();
    }
}

```

#### 4.2.12 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 261);
- настроить в сообщениях текст уведомления и названия кнопок (страница 301).

```
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) {
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

String uri = "https://" + DOMAIN + "/blitz/pipes/choice/start?appId=" +
↳CLIENT_ID + "&pipeId=select_value&choices=" + choiceJson;
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 `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"
    }
  ]
}

```

## 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 244).

### 4.3.1 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 `IGeoData` с функциями `getCountry()` и `getCity()`.

```

public class IGeoData {
  /**
   * Get IP address country
   *
   * @return - country or null if country not specified.

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

*/
public final String getCountry();

/**
 * 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
```

Листинг 1: 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());

```

Листинг 2: 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;

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

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 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();
    }
}

```

### 4.3.2 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)

нию 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.*;
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[]{});
        }
    }
}
```

(продолжается на следующей странице)



(продолжение с предыдущей страницы)

```

@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[]{});
}
}

```

### 4.3.3 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 294), 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 130):

```

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>

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

↪singletonMap("status", "" + result.status());
        } else {
            return HttpLoop.error("wrong_http_status",
                Collections.<String, String>
↪singletonMap("status", "" + result.status());
        }
    }
}

```

#### 4.3.4 Анализ меток приложений

Blitz Identity Provider позволяет присваивать приложениям *метки* (страница 207) 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 {

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

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) {
    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();
}
}

```

## 4.4 Кастомизация логики операций с хранилищами данных

### 4.4.1 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`

### 4.4.2 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.

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

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

- 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.

Листинг 3: Кастомизация операций `searchUsers` и `bindUser`

```
{
  "logic": {
    "searchUsers": {
      "enabled": true
    },
    "bindUser": {
      "enabled": true
    }
  }
}
```

#### 4.4.3 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:

- 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).
- 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](mailto:support@idblitz.ru).

#### 4.5 Процедуры привязки аккаунтов внешних поставщиков

Помимо базовой (страница 171) 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_IESia`. 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.

```

1 package com.identityblitz.idp.federation.matching.dynamic;
2
3 import java.lang.*;
4 import java.util.*;
5 import java.text.*;
6 import java.time.*;
7 import java.math.*;
8 import java.security.*;
9 import javax.crypto.*;
10 import org.slf4j.LoggerFactory;
11 import org.slf4j.Logger;
12 import com.identityblitz.idp.federation.*;
13 import com.identityblitz.idp.federation.matching.*;
14 import com.identityblitz.idp.flow.common.api.*;
15 import com.identityblitz.idp.flow.common.model.*;
16 import com.identityblitz.idp.federation.matching.dynamic.*;
17 import java.util.function.Consumer;
18 import java.util.stream.Stream;
19 import java.util.stream.Collectors;
20 import org.codehaus.jackson.map.ObjectMapper;
21 import org.codehaus.jackson.type.TypeReference;
22 import com.identityblitz.idp.extensions.types.JsonObject;
23
24 import com.identityblitz.idp.federation.matching.*;
25 import com.identityblitz.idp.flow.common.api.HttpFactory;
26
27 /**
28  * Класс наследуется от MatchingBlock и для корректного инстантирования должен иметь default конструктор.
29  * Текущая сгенерированная реализация обеспечивает стратегию при которой пользователи не сопоставляются и не обновляются.
30  */
31 public class Esia_IESia extends MatchingBlock {
32
33     private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.federation.matching.dynamic");
34
35     /**
36      * Итеративная функция определяющая соответствие внутренних УЗ и УЗ поставщика идентификации.
37      * На каждой итерации функция может выполнить операцию find (найденные пользователи будут переданы в следующей итерации)
38      * или завершить операцию со следующими решениями:
39      *   * matched – соответствующие пользователи найдены;
40      *   * matchError – ошибка определения соответствия пользователей;
41      *   * matchByLogin – осуществить связку с пользователем, который успешно аутентифицируется;
42      *   * refine – получить список пользователей, запрашивает пароль и осуществляет связку с тем пользователем, для которого введен корректный пароль;
43      * @param ctx – контекст процедур со следующими полями:
44      *   * iteration – номер итерации процедуры;
45      *   * extAttrs – атрибуты пользователя, полученные от поставщика идентификации;
46      *   * sid – уникальный идентификатор внешней УЗ.
47      * @param users – пользователи.
48      * @return – одно из перечисленных решений
49      */
50     @Override public MatchResult match(MatchingContext ctx, List<MatchingUser> users){
51         return matchError(ctx, new MatchingError("not_matched", "User not matched"));
52     };
53
54     /**
55      * Возвращает обновляемые и удаляемые атрибуты.
56      * @param extAttrs – атрибуты пользователя, полученные от поставщика идентификации.
57      * @param user – внутренний пользователь.
58      * @param justMatched – признак того, что связь внутренних УЗ с УЗ внешнего поставщика установлена впервые.
59      * @return – кортеж с изменяемыми и удаляемыми атрибутами. Например: change(JsonObject.empty(), Collections.<String>emptySet())
60      */
61     @Override public Tuple2<JsonObject, Set<String>> update(JsonObject extAttrs, MatchingUser user, Boolean justMatched, HttpFactory httpFactory){
62         return change(JsonObject.empty(), Collections.<String>emptySet());
63     };
64 }
65

```

Отмена

Сохранить

### 4.5.1 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());
};
}
```

## 4.5.2 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.JsonObject;

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());
    };
};
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

private MatchResult tryToSearch(MatchingContext ctx) {
    return find(ctx, filter().eq("uid", "test@test.ru"));
}

@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);
    }
};
}

```

## 4.6 Процедуры обработки RADIUS-запросов

### 4.6.1 О RADIUS-процедурах

Процедуры обработки RADIUS-запросов (страница 236) - это процедуры на языке программирования Java, которые обрабатывают запросы из приложений, подключенных к Blitz Identity Provider по протоколу RADIUS<sup>60</sup>. Процедуры определяют политики доступа в сетевые ресурсы и реализуют процесс аутентификации на разных факторах. Данный раздел описывает синтаксис создания RADIUS-процедур.

### 4.6.2 Интерфейс RadiusFlow

RADIUS-процедура должна реализовать интерфейс `RadiusFlow`, входящий в состав Blitz Development Kit (Blitz SDK):

```

package com.identityblitz.idp.radius.flow;

import java.util.Map;

public interface RadiusFlow {
    String loginN12(String var1);

    RadiusResult next(RadiusContext var1);

    RadiusResult dialog(RadiusContext var1, String var2, Map<String, String> var3, ↪
↪String var4);
}

```

Интерфейс `RadiusFlow` содержит следующие методы:

<sup>60</sup> <https://datatracker.ietf.org/doc/html/rfc2865>



- `String loginN12(String var1)`: возвращает логин, введенный пользователем, и записывает в лог информацию о вводе логина.

```
package com.identityblitz.idp.radius.flow;

import org.slf4j.LoggerFactory;
import org.slf4j.Logger;
import com.identityblitz.idp.radius.flow.RadiusResult.*;
...

public class RadiusLog implements RadiusFlow {

    public String loginN12(final String login) {
        logger.debug("#### Start loginN12: {}", login);
        return login;
    }
    ...
}
```

- `RadiusResult next(RadiusContext var1)`: метод, ссылающийся на интерфейс `RadiusResult`, определяющий алгоритм аутентификации пользователя в зависимости от *контекста входа* (страница 273).
- `RadiusResult dialog(RadiusContext var1, String var2, Map<String, String> var3, String var4)`: метод, ссылающийся на интерфейс `RadiusResult`, определяющий *отображение диалогов* (страница 274) на разных стадиях аутентификации.

#### 4.6.3 Контекст входа RadiusContext

Алгоритм аутентификации и отображение диалогов определяются в процедуре исходя из контекста входа по RADIUS. Для его описания используется интерфейс `RadiusContext`, содержащий следующий набор методов:

- `int factor()`: возвращает порядковый номер фактора аутентификации.
- `List<String> passedMethod()`: возвращает список пройденных методов аутентификации.
- `String clientId()`: возвращает значение параметра `client_id` приложения.
- `String subject()`: возвращает `id` пользователя.
- `String claim(String var1)`: возвращает значение утверждения о пользователе по имени утверждения.
- `String userProp(String var1)`: возвращает значение свойства пользователя по имени свойства.
- `List<com.identityblitz.idp.radius.flow.RadiusContext.Group> groups()`: возвращает список групп пользователя.

```
package com.identityblitz.idp.radius.flow;

import java.util.List;

public interface RadiusContext {
    int factor();

    List<String> passedMethod();

    String clientId();

    String subject();
}
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
String claim(String var1);

String userProp(String var1);

List<com.identityblitz.idp.radius.flow.RadiusContext.Group> groups();
```

#### 4.6.4 Интерфейс RadiusResult

Интерфейс `RadiusResult` определяет поведение системы при проведении аутентификации, включая алгоритмы аутентификации и отображение диалогов:

```
package com.identityblitz.idp.radius.flow;

import com.identityblitz.idp.radius.ChallengeCommunication;
import java.util.Arrays;
import java.util.Collections;
import java.util.Map;

public interface RadiusResult {
    static RadiusResult rejected(String cause) {
        return new com.identityblitz.idp.radius.flow.RadiusResult.
↳RejectedResult(cause);
    }

    static RadiusResult challenge(ChallengeCommunication chlg) {
        return new com.identityblitz.idp.radius.flow.RadiusResult.
↳ChallengesResult(Collections.singletonList(chlg));
    }

    static RadiusResult challenges(ChallengeCommunication... chlgs) {
        return new com.identityblitz.idp.radius.flow.RadiusResult.
↳ChallengesResult(Arrays.asList(chlgs));
    }

    static RadiusResult dialog(String message) {
        return new com.identityblitz.idp.radius.flow.RadiusResult.
↳DialogResult(message, Collections.emptyMap());
    }

    static RadiusResult dialog(String message, Map<String, String> answers) {
        return new com.identityblitz.idp.radius.flow.RadiusResult.
↳DialogResult(message, answers);
    }

    static com.identityblitz.idp.radius.flow.RadiusResult.VendorSpecificAttributes↳
↳vendorSpecific() {
        return new com.identityblitz.idp.radius.flow.RadiusResult.
↳VendorSpecificAttributes();
    }

    static RadiusResult authenticated(String subjectId) {
        return new com.identityblitz.idp.radius.flow.RadiusResult.
↳AuthenticatedResult(subjectId, new com.identityblitz.idp.radius.flow.
↳RadiusResult.VendorSpecificAttributes());
    }
}
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

static RadiusResult authenticated(String subjectId, com.identityblitz.idp.
↪radius.flow.RadiusResult.VendorSpecificAttributes avps) {
    return new com.identityblitz.idp.radius.flow.RadiusResult.
↪AuthenticatedResult(subjectId, avps);
}
}

```

- RadiusResult rejected(String cause): вызов диалога об отказе в аутентификации на основании указанной причины cause.

### Реализация в Blitz SDK

Класс RadiusResult.RejectedResult

```

package com.identityblitz.idp.radius.flow;

public class RadiusResult$RejectedResult implements RadiusResult {
    private final String cause;

    public RadiusResult$RejectedResult(String cause) {
        this.cause = cause;
    }

    public String getCause() {
        return this.cause;
    }
}

```

Пример использования в процедуре:

```

return RadiusResult.rejected("This method is temporarily unavailable");

```

- RadiusResult challenge(ChallengeCommunication chlg) и RadiusResult challenges(ChallengeCommunication... chlgs): вызов диалогов различных методов аутентификации (SMS, push, email, TOTP, HOTP, через Личный кабинет, по паролю, вызов метода по имени).

### Реализация в Blitz SDK

Класс RadiusResult.Challenges

```

package com.identityblitz.idp.radius.flow;

import com.identityblitz.idp.radius.ChallengeCommunication;
import com.identityblitz.idp.radius.SmsChallenge;
import com.identityblitz.idp.radius.EmailChallenge;
import scala.Option;

public class RadiusResult$Challenges {
    /* Метод вызывает настроенный по умолчанию канал доставки
    разового кода SMS/push. Если канал не работает,
    код будет отправлен по второму каналу */
    public static ChallengeCommunication sms() {
        return new SmsChallenge(Option.apply((Object) null));
    }
    /* Метод явно вызывает канал доставки
    разового кода SMS/push, указанный в переменной way.

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

    Можно передать: "sms", "push".
    Канал доставки должен быть настроен, иначе метод не сработает */
public static ChallengeCommunication sms(String way) {
    return new SmsChallenge(Option.apply(way));
}
// Метод входа по email
public static ChallengeCommunication email() {
    return .MODULE$;
}
// Метод входа через TOTP
public static ChallengeCommunication totp() {
    return com.identityblitz.idp.radius.TotpChallenge..MODULE$;
}
// Метод входа через HOTP
public static ChallengeCommunication hotp() {
    return com.identityblitz.idp.radius.HotpChallenge..MODULE$;
}

// Метод входа через Личный кабинет

public static ChallengeCommunication profile() {
    return com.identityblitz.idp.radius.ProfileChallenge..MODULE$;
}
// Метод входа по паролю
public static ChallengeCommunication password() {
    return com.identityblitz.idp.radius.PasswordChallenge..MODULE$;
}

/* Метод используется для вызова переданного в пате метода входа.
    Можно передать: "sms", "push", "email", "totp", "hotp", "profile",
    "password", "mobile" (использование канала SMS/push,
    настроенного по умолчанию) */

public static ChallengeCommunication byDialogName(String name) {
    return com.identityblitz.idp.radius.ChallengeCommunication..MODULE$.
↪byDialogName(name);
}
}

```

Примеры использования в процедуре:

Листинг 4: Вызов метода аутентификации по паролю

```
return RadiusResult.challenge(Challenges.password());
```

Листинг 5: Вызов метода аутентификации по email

```
return RadiusResult.challenge(Challenges.email());
```

Листинг 6: Вызов фактора методом из константы challenge

```
final String challenge = "totp";
return RadiusResult.challenge(Challenges.byDialogName(challenge));
```

Листинг 7: Явный вызов метода аутентификации по push

```
return RadiusResult.challenge(Challenges.sms("push"))
```

Листинг 8: Явный вызов метода аутентификации по SMS

```
return RadiusResult.challenge(Challenges.sms("sms"))
```

Для доставки кода по настроенному по умолчанию каналу SMS/push можно использовать любой из следующих методов:

#### **i** Примечание

Если канал по умолчанию не работает, будет использован второй канал.

##### 1. sms ()

```
return RadiusResult.challenge(Challenges.sms());
```

##### 2. byDialogName ("mobile")

```
return RadiusResult.challenge(Challenges.byDialogName("mobile"));
```

- Перегрузки метода `RadiusResult dialog`:

- `RadiusResult dialog(String message)`: отображение сообщения `message`;
- `RadiusResult dialog(String message, Map<String, String> answers)`: вызов диалога, состоящего из инструкции по прохождению фактора `message` с возможностью выбрать нужный метод из словаря `answers`.

#### **i** Примечание

Для задания текста конкретного сообщения `message` (в примере ниже `challengeChooseNew`) см. *Тексты диалоговых сообщений* (страница 281).

Пример использования в процедуре:

```
/* Создать словарь answers, задающий соответствие между
   * ответами пользователя и методами входами */
final java.util.Map<String, String> answers = new java.util.HashMap<String,
↳String> ();
answers.put ("1", "sms");
answers.put ("2", "totp");
answers.put ("3", "prfc");

/* Вызвать метод RadiusResult.dialog,
   * который отображает диалог challengeChooseNew с вариантами
   * ответов из словаря answers и принимает ответ пользователя */
return RadiusResult.dialog("challengeChooseNew", answers);
```

- `com.identityblitz.idp.radius.flow.RadiusResult.VendorSpecificAttributes vendorSpecific()`: работа с атрибутами, поступающими от сетевого оборудования определенного вендора.

**Реализация в Blitz SDK**

Класс RadiusResult\$VendorSpecificAttributes

```

package com.identityblitz.idp.radius.flow;

import java.util.HashMap;
import java.util.Map;

public class RadiusResult$VendorSpecificAttributes {
    private final Map<Long, Map<Integer, Object>> avps = new HashMap();

    protected RadiusResult$VendorSpecificAttributes() {
    }

    public RadiusResult$VendorSpecificAttributes withAttribute(long vendorId, ↵
    ↵int vType, Object value) {
        Map<Integer, Object> byVendor = (Map)this.avps.get(vendorId);
        if (byVendor == null) {
            Map<Integer, Object> nByVendor = new HashMap();
            this.avps.put(vendorId, nByVendor);
            nByVendor.put(vType, value);
        } else {
            byVendor.put(vType, value);
        }

        return this;
    }

    protected Map<Long, Map<Integer, Object>> getAvps() {
        return this.avps;
    }
}

```

- Перегрузки метода RadiusResult authenticated:
  - RadiusResult authenticated(String subjectId): проведение аутентификации пользователя с указанным идентификатором;
  - RadiusResult authenticated(String subjectId, com.identityblitz.idp.radius.flow.RadiusResult.VendorSpecificAttributes avps): проведение аутентификации пользователя с указанным идентификатором и значениями атрибутов сетевого оборудования определенного вендора.

**Реализация в Blitz SDK**

Класс RadiusResult\$AuthenticatedResult

```

package com.identityblitz.idp.radius.flow;

import com.identityblitz.idp.radius.flow.RadiusResult.VendorSpecificAttributes;
import java.util.Map;

public class RadiusResult$AuthenticatedResult implements RadiusResult {
    private final String subjectId;
    private final VendorSpecificAttributes avps;

    public RadiusResult$AuthenticatedResult(String subjectId, ↵
    ↵VendorSpecificAttributes avps) {

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

        this.subjectId = subjectId;
        this.avps = avps;
    }

    public String getSubjectId() {
        return this.subjectId;
    }

    public Map<Long, Map<Integer, Object>> getAvps() {
        return this.avps.getAvps();
    }
}

```

Пример использования в процедуре:

```

public RadiusResult next(final RadiusContext context) {
    ...
    RadiusResult.authenticated(context.subject());
}

```

#### 4.6.5 Пример процедуры

Пример процедуры с комментариями и логированием на уровне DEBUG показан ниже. Руководствуясь данным примером и Blitz SDK, перепишите процедуру в соответствии со своими требованиями к входу в сетевые службы.

```

package com.identityblitz.idp.radius.flow;

import org.slf4j.LoggerFactory;
import org.slf4j.Logger;
import com.identityblitz.idp.radius.flow.RadiusResult.*;

public class RadFLOWFull implements RadiusFlow {

    // Создать логгер для класса com.identityblitz.idp.flow.radius
    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.
↪flow.radius");

    /* Метод возвращает введенный пользователем логин
    и логирует информацию о вводе логина */
    public String loginN12(final String login) {
        logger.debug("#### Logging in via RADIUS. User entered login: {}", login);
        return login;
    }

    /* Вызов метода, реализующего алгоритм аутентификации
    в зависимости от контекста входа */
    public RadiusResult next(final RadiusContext context) {

        /* Логирование на уровне DEBUG: id пользователя и
        порядковый номер фактора аутентификации из контекста */
        logger.debug("### Getting subject id and factor number: subject = {}, factor =
↪{}", context.subject(), context.factor());

        /* Если пользователь в данный момент проходит первый
        фактор аутентификации, запросить ввод пароля */

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

if (context.factor() == 1) {
    return RadiusResult.challenge(Challenges.password());

    /* Иначе проверить, проходит ли пользователь в данный момент второй фактор.
    В этом случае создать словарь answers, задающий соответствие между
    ответами пользователя и методами входами.
    При вводе "1" будет реализован метод входа по SMS, при вводе "2" -
    ↪подтверждение через Личный кабинет
    */
} else if (context.factor() == 2) {
    final java.util.Map<String, String> answers = new java.util.HashMap<String, ↪
    ↪String>();
    answers.put("1", "sms");
    answers.put("2", "prfc");

    /* Вызвать метод RadiusResult.dialog,
    который отображает диалог challengeChoose с вариантами
    ответа answers и принимает ответ пользователя */
    return RadiusResult.dialog("challengeChoose", answers);

    /* Если фактор не первый и не второй, значит пользователь прошел
    оба фактора. Аутентифицировать пользователя с указанным идентификатором */
} else {
    return RadiusResult.authenticated(context.subject());
}
}

/* Реализация метода RadiusResult.dialog. Метод принимает контекст,
диалоговое сообщение, словарь с вариантами ответа пользователя,
введенный ответ пользователя */
public RadiusResult dialog(final RadiusContext context,
                           final String message,
                           final java.util.Map<String, String> answers,
                           final String answer) {

    // Логировать на уровне DEBUG вывод диалога для пользователя
    logger.debug("#### Start dialog for user: {}", context.subject());

    // Логировать вывод диалогового сообщения
    logger.debug("#### Start dialog with message: {}", message);

    /* Если передается сообщение challengeChoose,
    записать в константу challenge ответ пользователя */
    if(message.equals("challengeChoose")) {
        final String challenge = answers.get(answer);

        // Логировать на уровне DEBUG ответ пользователя
        logger.debug("#### Get challenge by dialog: {}", challenge);

        // Проверить, что пользователь отправил непустой ответ
        if (challenge != null) {
            // Если ответ равен prfc, создать словарь confirm
            if (challenge.equals("prfc")) {
                final java.util.Map<String, String> confirm = new java.util.HashMap
                ↪<String, String>();

```

(продолжается на следующей странице)



(продолжение с предыдущей страницы)

```

/* Добавить в словарь confirm соответствие между нажатием Enter
и отправкой подтверждения входа в Личный кабинет */
confirm.put("OK", "prfc_configrm");

/* Отобразить сообщение confirm_profile и
принять действие пользователя (нажатие Enter) */
return RadiusResult.dialog("confirm_profile", confirm);

/* Если ответ пользователя не совпадает с prfc,
вызвать метод входа, соответствующий ответу пользователя */
} else {
return RadiusResult.challenge(Challenges.byDialogName(challenge));
}

/* Если пользователь отправил пустой ответ,
повторно отобразить диалог с вариантами ответа answers */
} else {
return RadiusResult.dialog(message, answers);
}

/* Если передается сообщение, отличное от challengeChoose,
проверить его на соответствие confirm_profile.
Если это так, вызвать метод входа через Личный кабинет */
} else if (message.equals("confirm_profile")) {
return RadiusResult.challenge(Challenges.profile());
}
// В противном случае отказать в аутентификации по передаваемой в метод причине
else {
return RadiusResult.rejected("Login method not specified");
}
}
}
}

```

#### 4.6.6 Тексты диалоговых сообщений

Для того чтобы определить текст диалоговых сообщений, вызываемых в RADIUS-процедурах, руководитесь *стандартным алгоритмом* (страница 294) внесения изменений в тексты интерфейса.

Идентификатор строки для задания диалогового сообщения `message` из методов `RadiusResult.dialog(message, answers)` и `RadiusResult.dialog(message)` должен быть в формате `radius.display.dialog.message.<message>`. Например, для реализации процедуры из *примера* (страница 279) в файл `messages` вносятся следующие изменения:

```

radius.display.dialog.message.challengeChoose=Выберите способ подтверждения входа:↵
↵\n 1 - Подтверждение кодом из SMS \n 2 - Подтверждение в Личном кабинете (https://
↵/sso.domain.ru/blitz/profile) \n Выбор:
radius.display.dialog.challenge.name.prfc=Подтверждение в профиле пользователя\n
radius.display.dialog.challenge.name.sms=Подтверждение кодом из SMS\n
radius.display.dialog.message.confirm_profile=Нажмите Enter/Ввод для подтверждения↵
↵входа в Личном кабинете (https://sso.domain.ru/blitz/profile).

```

Сообщение `radius.display.dialog.message.challengeChoose` вызывается в методе `RadiusResult.dialog("challengeChoose", answers)`, а сообщение `radius.display.dialog.message.confirm_profile` вызывается в методе `RadiusResult.dialog("confirm_profile", confirm)`.

В строках `radius.display.dialog.challenge.name.prfc` и `radius.display.dialog.challenge.`

`name.sms` перезаписываются диалоги, отображаемые по умолчанию для метода входа через Личный кабинет (`prfc`) и по коду из SMS (`sms`).

При необходимости вы можете менять следующие строки по умолчанию:

```
radius.display.vrf_code.sms=Введите код из SMS:
radius.display.vrf_code.sms.on_error=Неверный код. Попробуйте еще раз:
radius.display.vrf_code.push=Введите код из push-сообщения:
radius.display.vrf_code.push.on_error=Неверный код. Попробуйте еще раз:
radius.display.vrf_code.totp=Введите код из приложения {0}:
radius.display.vrf_code.totp.onError=Неверный код. Попробуйте еще раз:
radius.display.vrf_code.totp.noTotp=Вход по генератору разовых паролей не настроен.
↪ в учетной записи.
radius.display.vrf_code.hotp=Введите код из приложения {0}:
radius.display.vrf_code.hotp.onError=Неверный код. Попробуйте еще раз:
radius.display.vrf_code.hotp.noHotp=Вход по генератору разовых паролей не настроен.
↪ в учетной записи.
radius.display.vrf_code.email=Введите код из email:
radius.display.vrf_code.email.on_error=Неверный код. Попробуйте еще раз:
radius.display.dialog.message.challengeChoose=Выберите метод аутентификации: {0}
radius.display.dialog.challenge.name.password=пароль
radius.display.dialog.challenge.name.sms=sms
radius.display.dialog.challenge.name.push=push
radius.display.dialog.challenge.name.mobile=sms/push
radius.display.dialog.challenge.name.email=email
radius.display.dialog.challenge.name.totp=TOTP
radius.display.dialog.challenge.name.hotp=HOTP
radius.display.dialog.challenge.name.prfc=профайл
radius.display.dialog.challenge.name.trustKey=доверенное приложение
```

## Глава 5

# Дизайн и тексты интерфейса

### 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 логота.

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.

#### 5.1.1 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):

**Свойства шаблона**

Идентификатор шаблона

Название шаблона

Теги шаблона

Задайте теги, чтобы ограничить область применения темы. В качестве тегов может выступать модуль (например, "blitz-idp", "blitz-recovery", "blitz-registration") или режим аутентификации (например, "sso", "required", "extBinding", "logout"). Будет использоваться тема с наибольшим числом совпадений тегов с текущим запросом на аутентификацию

Описание

Приложения


[Удалить](#) [Сохранить](#)

Настройка внешнего вида страницы входа (внешний вид страницы входа):

**Внешний вид страницы входа**

Тема


Расположение основного блока  Слева  По центру  Справа



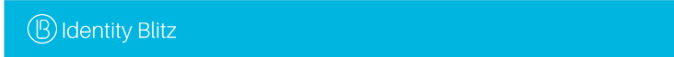
Выбор языка

Настройка внешнего вида страницы входа (логотип):

**Логотип**



Рекомендуемая высота логотипа 32px



Настройка внешнего вида страницы входа (фоновый рисунок):




Фоновый рисунок

📁 Загрузить фоновый рисунок

Рекомендуемый размер файла не более 1MB

Или выбрать из предложенных рисунков

Рисунок 1     
  Рисунок 2     
  Рисунок 3

Настройка внешнего вида страницы входа (настройка футера):

Настройка футера

Добавьте фрагмент HTML-кода для отображения в футере страницы входа.

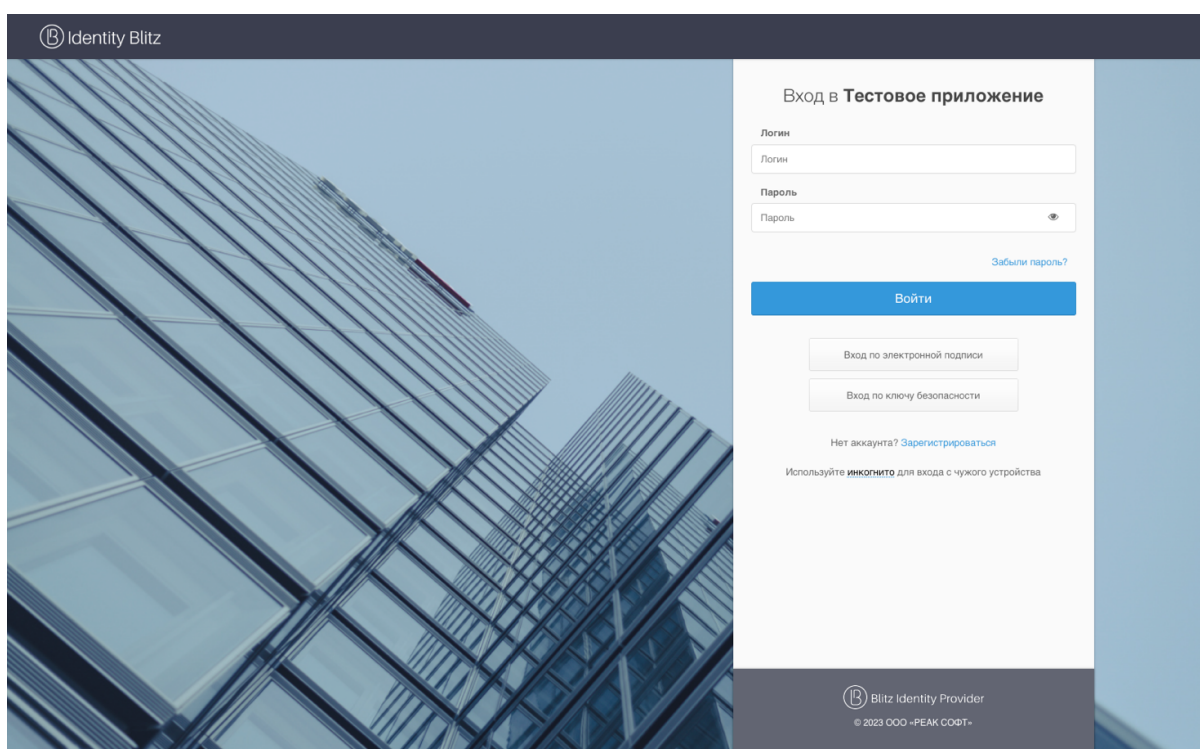
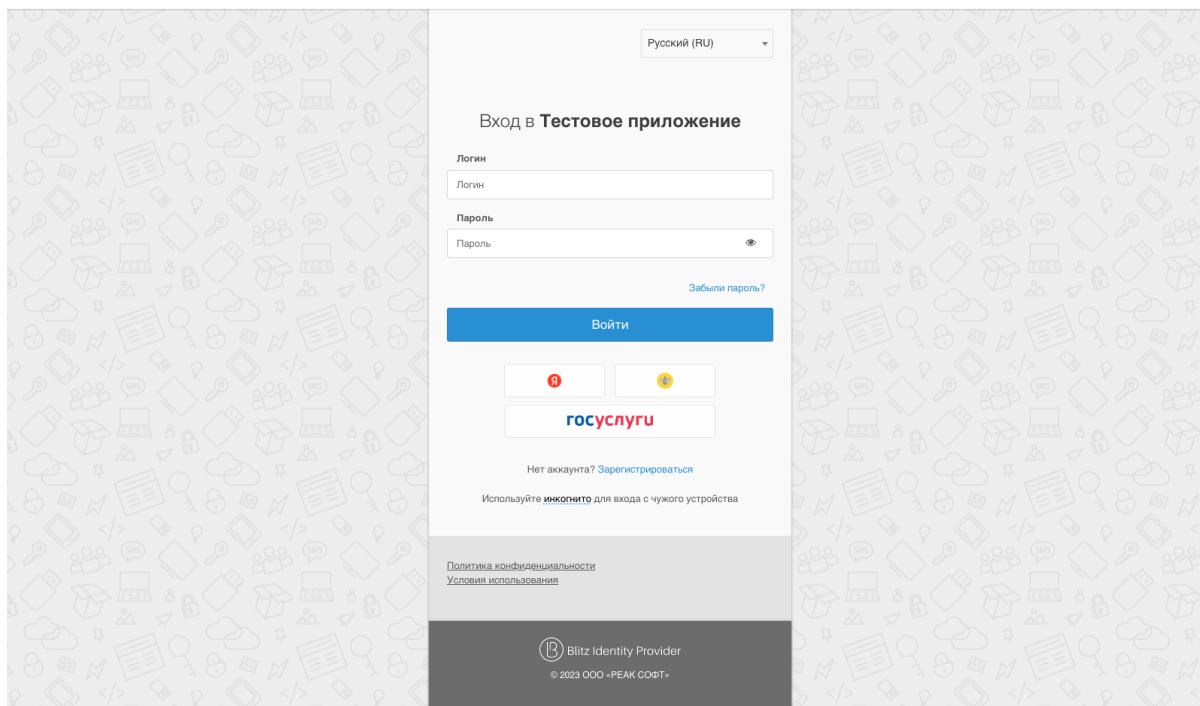
1 Скопируйте сюда фрагмент HTML кода

[Сохранить](#)

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.



### 5.1.2 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.

**Шаблоны страниц входа**

Для оформления страницы входа используются шаблоны. Вы можете отредактировать основной шаблон или создать дополнительные шаблоны, которые будут применяться для страниц. Вы можете создавать дополнительные шаблоны, сохранив копию основного шаблона, а также редактировать шаблоны вручную.

Идентификатор	Название шаблона	Тэги	Приложения	Описание	
default	default		test_app, test_app2	Generated at 1677833153	
default_1	default_1	logout, blitz-idp, blitz-recovery, blitz-registration	design_test	Generated at 1677833153	

Чтобы новый шаблон использовался при входе в некоторое приложение, необходимо в разделе «*Приложения*» перейти к редактированию нужного приложения и выбрать требуемый шаблон страниц.

**Параметры приложения**

Идентификатор (entityID или client\_id): test\_app  
Идентификатор приложения. Используется для идентификации приложения при доступе по протоколу SAML (соответствует entityID) и OAuth 2.0 (соответствует client\_id).

Название: Test app  
Отображаемое пользователям имя приложения. Используется только внутри Blitz Identity Provider

Домен: https://demo.company.com  
Ссылка на стартовую страницу приложения, например, http://testdomain.ru/. При TLS-аутентификации приложения проверяется, что в сертификате приложения указан именно этот домен

Стартовая страница приложения:   
Ссылка на стартовую страницу приложения, например, http://testdomain.ru/private. При входе по SAML используется как ссылка перехода в приложение, если открывать страницу входа из истории браузера

Ключ шифрования идентификаторов:   
Если ключ задан, то идентификатор пользователя для приложения будет зашифрован с использованием данного ключа. Значение ключа можно выбрать из списка. Также можно назначить новый ключ, для этого введите его в строке поиска и нажмите Enter

Шаблон страниц: default  
Шаблон страниц определяет внешний вид страниц входа. Если шаблон не указан, то используется шаблон по умолчанию.

### 5.1.3 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.

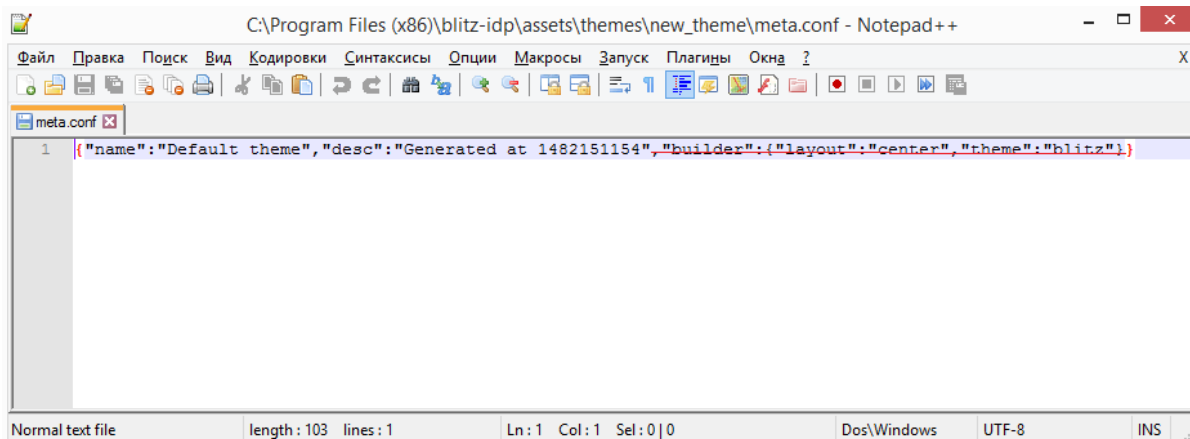
Каждый шаблон страницы входа представляет собой zip-архив. Все шаблоны размещены в директории:

```
\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<sup>61</sup>.

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.

Листинг простейшего шаблона приведен ниже:

```
@(headers: Html, fBuilder: FormBuilder, scripts: Html, path: String)(implicit_
  ↳request: RelyingPartyRequest[_], messages: Messages)

<!DOCTYPE html>
<html>

<head>
  @headers
</head>
```

(продолжается на следующей странице)

<sup>61</sup> <https://www.playframework.com/documentation/2.5.x/ScalaTemplates>



(продолжение с предыдущей страницы)

```
<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.

Вход в Личный кабинет

Логин

Пароль

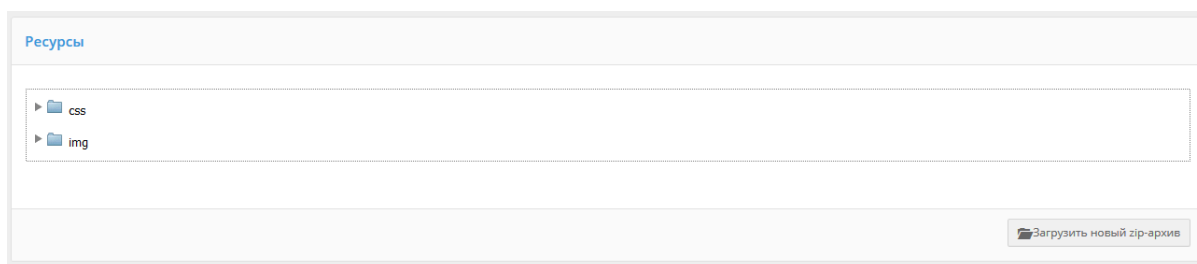
[Забыли пароль?](#)

**Войти**

[Нет аккаунта? Зарегистрироваться](#)

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 загрузить 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>
```

## 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.

### 5.2.1 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 `/usr/share/identityblitz/blitz-config/assets/public/lib/blitz-common/img/` новым файлом с тем же названием.

#### Способ 2

1. Положите в директорию `/usr/share/identityblitz/blitz-config/assets/public/lib/blitz-common/img/` 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 `/usr/share/identityblitz/blitz-config/assets/public/lib/blitz-common/img/` используйте `https://<domain>/blitz/public/img/`.

```
:root{
  --navbar-branding-img: url(https://<domain>/blitz/public/img/mylogo.png);
  ...
}
```

**Совет**

Для логотипа также можно использовать файл на внешнем URL.

### 5.2.2 Логотип в футере

Для замены логотипа в футере Личного кабинета замените файл логотипа `logo-bip.png` в директории `/usr/share/identityblitz/blitz-config/assets/public/lib/blitz-common/img/` **НОВЫМ** файлом с тем же названием.

### 5.2.3 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;
  --core-color-accent:#00bde5;
  --core-color-body:#666;
  --core-color-border:#ddd;
  --core-color-outline:#d9d9d9;
  --core-color-primary:#3498db;
  --core-color-sidebar:#f2f2f2;
  --core-color-light:#fff;
  --core-color-dark:#3b3f4f;
  --core-color-border-primary:#3498db;
  --core-color-footer:#fafafa;
  --core-color-ib-blue:#00bde5;
  --core-color-text-list:#fff;
  --core-color-text-danger:#dc4831;
  --core-color-text-footer:#898f97;
  --core-color-text-link:#666;
  --core-color-text-success:#64c556;
  --core-color-text-suggestion:#fff;
  --core-color-social-vk:#5e81a8;
  --core-color-social-vkid:#5e81a8;
  --core-color-social-esia:#0063b0;
  --core-color-social-esiadp:#0063b0;
  --core-color-social-sbrf:#00b017;
  --core-color-social-sbb:#127f8c;
  --core-color-social-alfa:#ef3124;
  --core-color-social-google:#009a22;
  --core-color-social-apple:#000;
  --core-color-social-yandex:#ff5651;
  --core-color-social-mail:#3d32d7;
  --core-color-social-blitz:#00bde5;
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

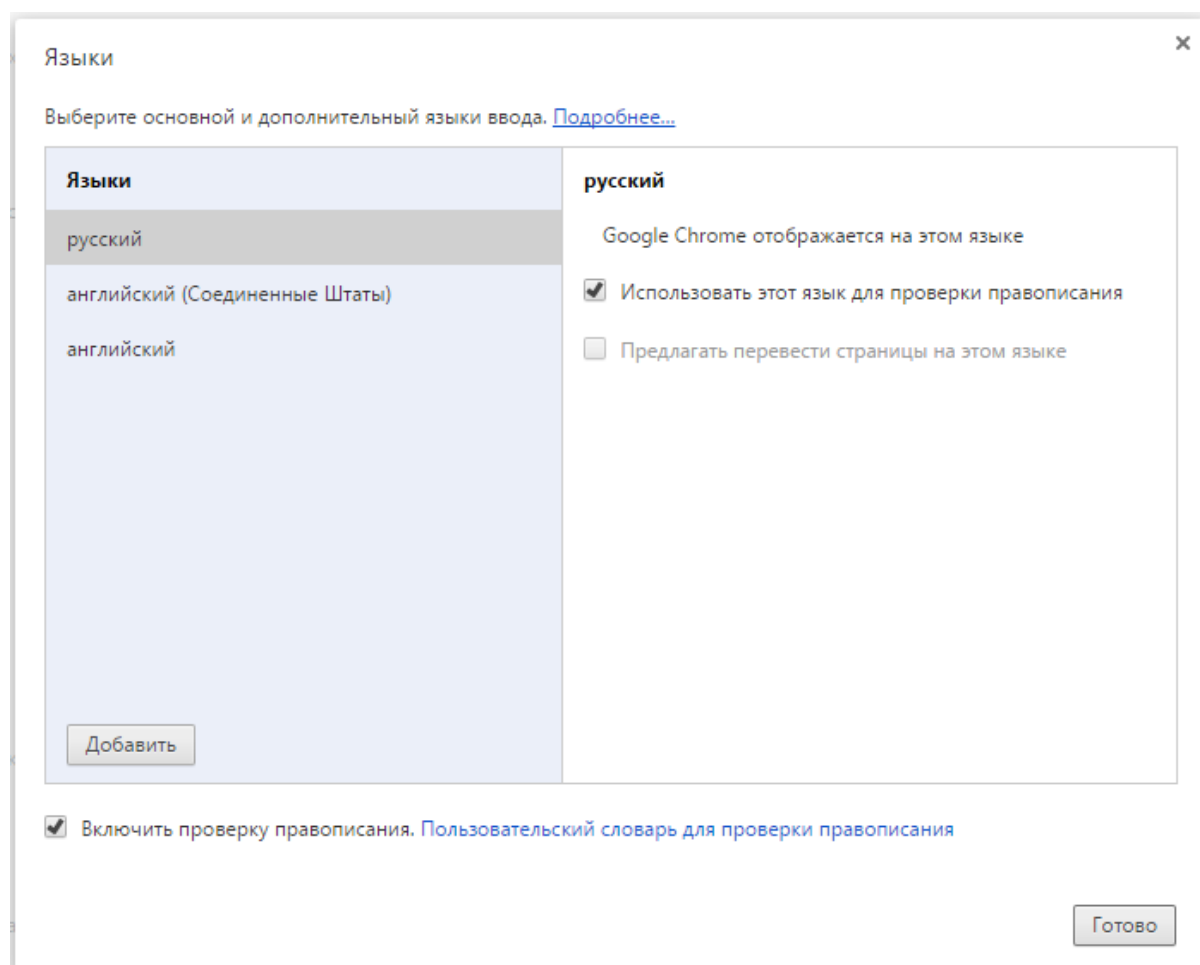
```
--core-color-social-mos: #b40031;
--core-color-social-facebook: #00cdff;
--core-color-social-tcs: #000;
--core-color-social-vtb: #0a2796;
--core-color-social-ok: #ff8100;
}
```

### 5.3 Мультиязычность

Веб-интерфейс Blitz Identity Provider поддерживает мультиязычность. По умолчанию предусмотрено два языка – русский и английский.

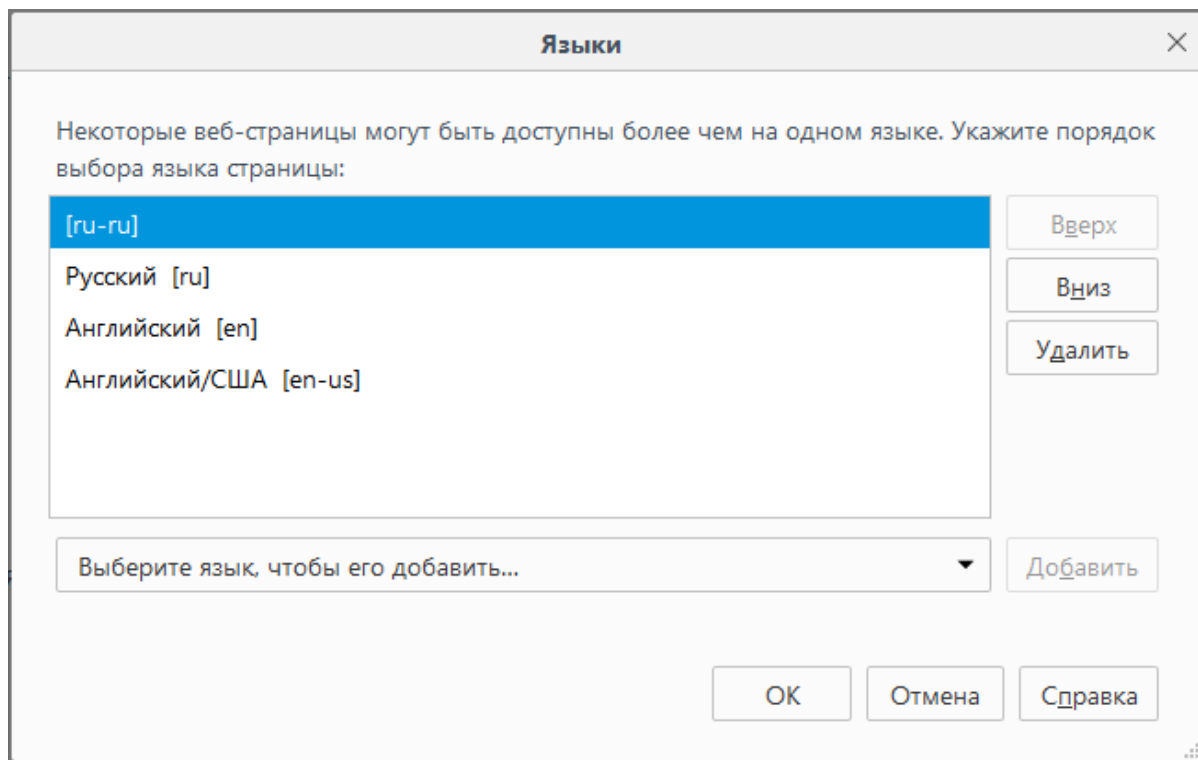
По умолчанию пользователю отображается интерфейс на том языке, который соответствует его системному языку в ОС и предпочтительному языку в браузере. В этом случае переключение языка осуществляется посредством изменения основного языка ввода (языка отображения веб-страниц) в используемом браузере. Например, 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 порталная 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 300).

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`.

## 5.4 Настройки текстов интерфейса

### 5.4.1 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.

### 5.4.2 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 г., 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 261).

### 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 назначении 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 предоставлении доступа к объекту.

Параметры:

- `blitz_right` - код права доступа;
- атрибуты субъекта;
- атрибуты объекта с префиксом `obj`.

**granted\_access\_on**

Информирование объекта о предоставлении 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.

### 5.4.3 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
```

#### 5.4.4 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 294) 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",
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

"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.

#### 5.4.5 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

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

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
↳ 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
```

## 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 137).

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
```

## Глава 6

# Настройки конфигурационных файлов

### 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 358);
- `assets/` – *настройки пользовательского интерфейса*. См.:
  - *Использование и обновление плагина* (страница 93),
  - *Внешние поставщики идентификации* (страница 137),
  - *Страница входа* (страница 283);
- `custom_messages/` – *строки пользовательского интерфейса* (страница 294);
- `devices/` – *вспомогательные каталоги для обработки загрузки HOTP и TOTP устройств* (страница 120);
- `/dynamic/idstore/` – *пользовательские процедуры для кастомизации логики операций с хранилищами данных* (страница 267);
- `flows/` – *процедуры входа* (страница 242);
- `saml/` – *настройки SAML* (страница 216);
- `simple/` – *настройки подключения приложений по протоколу Simple* (страница 207);
- `token_exchange/rules/` – *настройки правил обмена маркеров доступа*;
- `blitz.conf` – *основной файл конфигурации* (страница 305);
- `boot.conf` – *настройки путей к конфигурационным файлам*;
- `console.conf` – *настройки консоли управления* (страница 365);
- `credentials` – *учетные записи администраторов консоли управления* (страница 368);
- `play.conf` – *настройки серверов приложений*. См.:
  - *установка консоли управления `blitz-console` в Общая инструкция по установке* (страница 7),
  - *Домен Blitz Identity Provider* (страница 326);
- `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;
- `blitz-keeper-log4j.xml` – настройки журналирования событий и ошибок.

## 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 Identity Provider;
- `blitz.prod.local.idp.logger` – настройки логгеров;
- `blitz.prod.local.idp.login` – настройки методов аутентификации;
- `blitz.prod.local.idp.logout` – настройки процесса логута;
- `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.rest` – настройки добавления новых разрешений (`scopes`);
- `blitz.prod.local.idp.rights` – настройки прав доступа;

- `blitz.prod.local.idp.saml` – настройки SAML;
- `blitz.prod.local.idp.scopes-source` - настройки хранения разрешений (scopes);
- `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`.

## 6.2.1 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 системой 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",
  ...
}
```

## 6.2.2 Атрибуты

### Внешний валидатор атрибута

Если возможностей, предоставляемых *правилами преобразования* (страница 62) 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",
        "format" : "string",
        "name" : "phone_number",
        "realmed" : false,
        "required" : false,
        "searchable" : true,
        "source" : {
          "validators" : [
            {
              "className" : "validator.MobileValidator",
              "conf" : {
                "conf1" : "value1"
              }
            }
          ]
        },
        "type" : "idStore"
      }
    ]
  }
}
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

        },
        "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"
          }
        }
      ],
    },
    ...
  ]
}

```

- 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" : [
    {
      ...

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

        "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"
```

### 6.2.3 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
            },
            ...
        ],
    },
}

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

        {
            "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<sup>62</sup>.

Если необходимо, чтобы отображались только ключи подписи из реестра ОС Windows, доступные через MS CAPI, то блок настроек должен иметь следующий вид:

```

"plugin": {
    "allModulesEnabled": false,
    "capi": {
        "stores": [
            {
                "name": "My"
            }
        ]
    },
    "modules": []
}

```

<sup>62</sup> [https://identityblitz.ru/products/smart-card-plugin/documentation/?ref=p\\_pl#document-9](https://identityblitz.ru/products/smart-card-plugin/documentation/?ref=p_pl#document-9)

## 6.2.4 CAPTCHA

### reCAPTCHA

Для отображения сервиса reCAPTCHA 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-}"
      }
    },
    {
      "call": {
        "headers": [
          "accept:application/json",
          "Authorization:Bearer ${cfg.bearerToken}"
        ],
        "method": "post",
        "url": "https://captcha.example.com/captcha/1.0.0/refresh?"
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

↪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 reCAPTCHA (`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, указанным 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 следующими 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,
        "turnOn" : 30
      },
      "forced" : false
    },
    "user" : {
      "tTlInSec": 3600,
      "failedAttempts" : {
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

        "turnOn" : 5
      }
    }
  }
}

```

В случае использования в качестве CAPTCHA сервиса Google reCAPTCHA v3<sup>63</sup> 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" : "SITEKEY"
    },
    "secureParams" : {
      "secret" : "1234"
    }
  }
}

```

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"
  },
  "enabled" : true,
  "initJs" : "require(['blitz/assets/blitz-common/javascripts/recaptcha_v3.js',
↪'captcha-conf'], function(captcha, conf){ captcha(conf);});",

```

(продолжается на следующей странице)

<sup>63</sup> <https://developers.google.com/recaptcha/docs/v3>

(продолжение с предыдущей страницы)

```

    "name" : "reCAPTCHAav3"
  }

```

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" : "reCAPTCHAav3"
}

```

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" : "reCAPTCHAav3"
}

```

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" : "reCAPTCHAav3"
}

```

### Yandex SmartCaptcha

Для отображения сервиса Yandex SmartCaptcha при входе по логину и паролю, восстановлении доступа и регистрации необходимо внести изменения в конфигурационный файл.

Изменения конфигурационного файла должны быть произведены:

- в блоке настроек `blitz.prod.local.idp.captcha`. Пример записи настройки приведен ниже:

```
"captcha": {
  "yandexSmartCaptcha": {
    "operations": [
      {
        "call": {
          "body": {
            "contentType": "application/x-www-form-urlencoded",
            "params": {
              "ip": "${ctx.remoteAddress}",
              "secret": "${cfg.secret}",
              "token": "${ocp.token}"
            }
          },
          "headers": [],
          "method": "post",
          "url": "https://smartcaptcha.yandexcloud.net/validate"
        },
        "check": {
          "errRegExp": {},
          "okRegExp": {
            "status": "ok"
          }
        }
      },
      {
        "name": "verify"
      }
    ],
    "plainParams": {
      "sitekey": "ysc1"
    },
    "secureParams": {
      "secret": "1234"
    }
  }
}
```

В этом блоке в поле `secureParams.secret` укажите Ключ сервера, в поле `plainParams.sitekey` укажите Ключ клиента. Опционально в раздел `plainParams` можно добавить следующие параметры конфигурирования captcha: `sitekey`, `hl`, `test`, `webview`, `invisible`, `shieldPosition`, `hideShield`. Подробнее про эти параметры можно прочитать по ссылке<sup>64</sup>.

- в блоке `blitz.prod.local.idp.captcha` включите Yandex SmartCaptcha в методе входа по паролю, в форме регистрации и восстановления доступа. Общий принцип настройки капчи одинаков для всех методов, он представлен в следующем блоке:

```
"captcha": {
  "enabled": true,
  "initJs": "require(['./blitz/assets/javascripts/yandex_smart_captcha.js', 'captcha-
  ↪conf'], function(captcha, conf){ captcha(conf);});",
  "mode": {
    "type": "always_on"
  },
  "name": "yandexSmartCaptcha",
}
```

В этом блоке в поле `initJs` указать `require(['./blitz/assets/javascripts/yandex_smart_captcha.js', 'captcha-conf'], function(captcha, conf){ captcha(conf);`

<sup>64</sup> <https://yandex.cloud/ru/docs/smartcaptcha/concepts/widget-methods>

});, в поле name указать `yandexSmartCaptcha`.

Данный блок необходимо добавить в соответствующие разделы:

#### Вход по паролю

```
"provisioning" : {
  "password" : {
    "captcha" : {...},
```

#### Регистрация

```
"provisioning" : {
  "registration" : {
    "captcha" : {...},
```

#### Восстановление доступа

```
"provisioning" : {
  "recovery" : {
    "captcha" : {...},
```

Для получения дополнительной информации о процессе аутентификации с использованием пароля и капчи, см. раздел `auth-api-login-password`.

## 6.2.5 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" : { // optional
        "virtual": {
          "host": "testVhost"
        }
      },
      "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"
        },
        "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,
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

        "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"
    }
]
}

```

## 6.2.6 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"
    }
}

```

(продолжается на следующей странице)



(продолжение с предыдущей страницы)

```

},
...
}

```

### 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
<code>testOnBorrow</code>	<code>true</code>	Проверка состояния соединения перед отправкой данных. В случае ошибки соединение удаляется и выбирается следующее из пула.
<code>testOnCreate</code>	<code>false</code>	Проверка состояния соединения после создания в пуле.
<code>testOnReturn</code>	<code>false</code>	Проверка состояния соединения после возвращения в пул.
<code>testWhileIdle</code>	<code>false</code>	Проверка состояния соединения в состоянии <code>idle</code> . В случае ошибки соединение удаляется из пула.
<code>timeBetweenEvictionRunsMillis</code>	<code>-1</code>	Интервал в мс между запуском проверки соединения в состоянии <code>idle</code> , влияет на <code>testWhileIdle</code> .
<code>validationQuery</code>	<code>-</code>	SQL-запрос выполняемый при проверке состояния соединения из пула, при пустом значении используется <code>isValid()</code> .

```
"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 параметром `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 случаев 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 `firstSucceeded` 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`<sup>65</sup> 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 настройками 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;

<sup>65</sup> <https://www.maxmind.com/en/geoip2-databases>

- `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;
  - `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"
  ]
}

```

## 6.2.7 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 292) 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/
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

↔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/
↔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>

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

&lt;/EntityDescriptor&gt;

## 6.2.8 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 в 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 183).

### 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
  }
}
```



## 6.2.9 ЕСИА

### Контейнер ключей для ЕСИА и Цифрового профиля

Запросы на аутентификацию через внешние 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

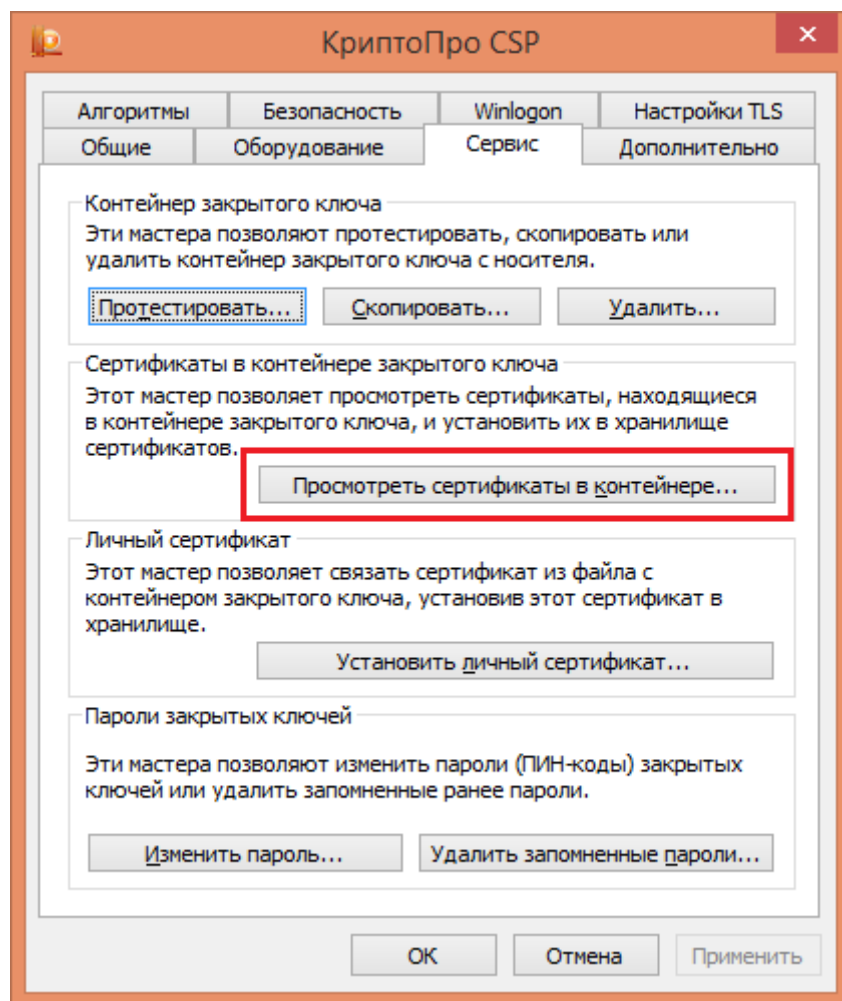
Для конвертации контейнера КриптоПро CSP в Windows понадобятся:

- ПК под управлением Windows с настроенным браузером Internet Explorer 11.
- Платная утилита P12FromGostCSP.
- Установленный на ПК криптопровайдер КриптоПро CSP (версия 4.0 и новее).

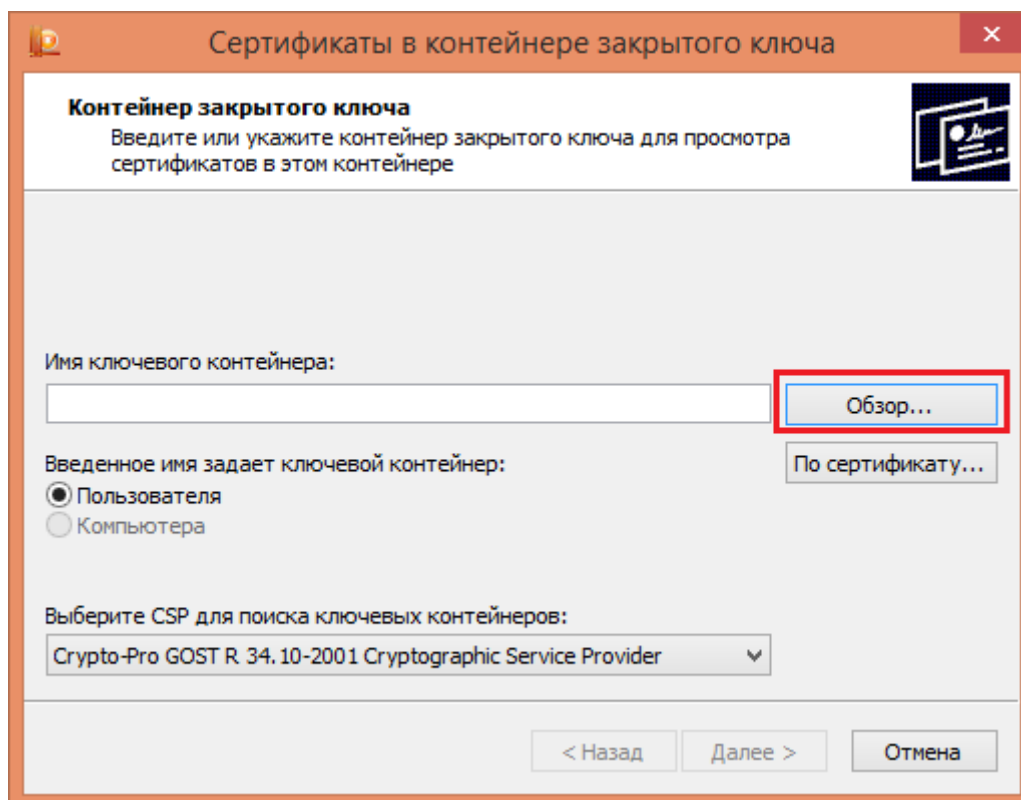
Инструкция по конвертации контейнера:

1. Установить на ПК под управлением Windows криптопровайдер КриптоПро CSP<sup>66</sup> (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:

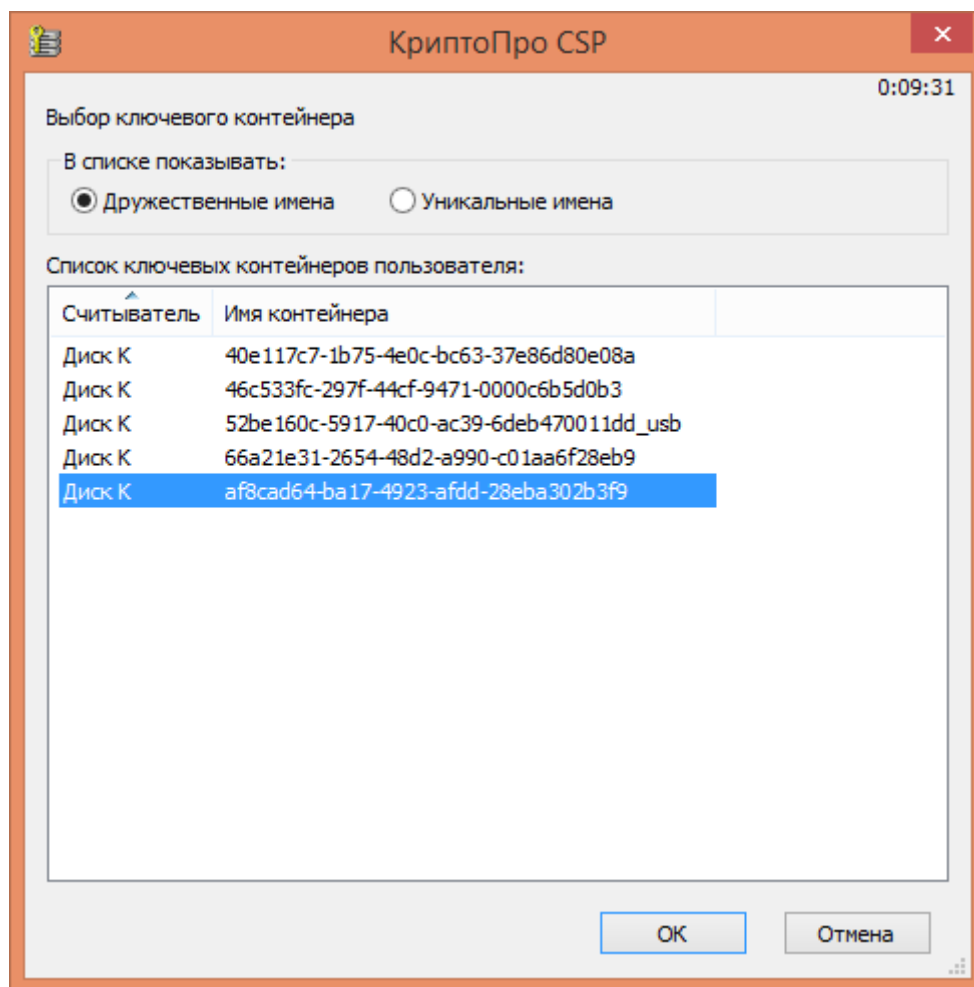
<sup>66</sup> <https://cryptopro.ru>



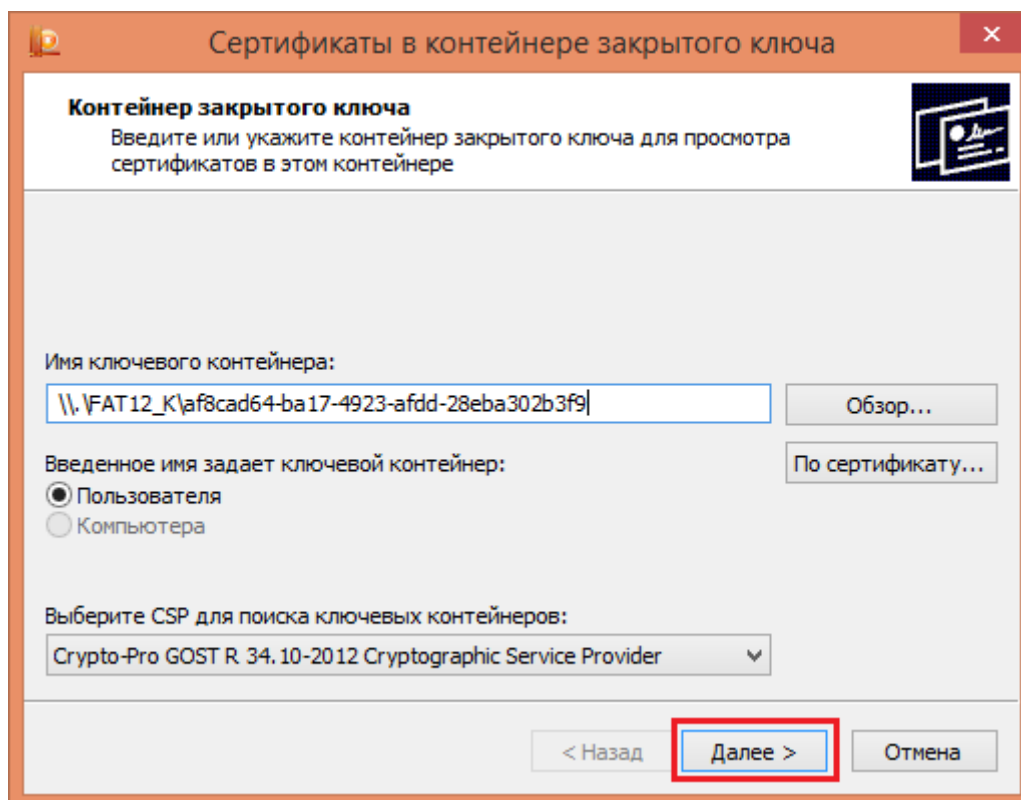
- нажать на кнопку Обзор:



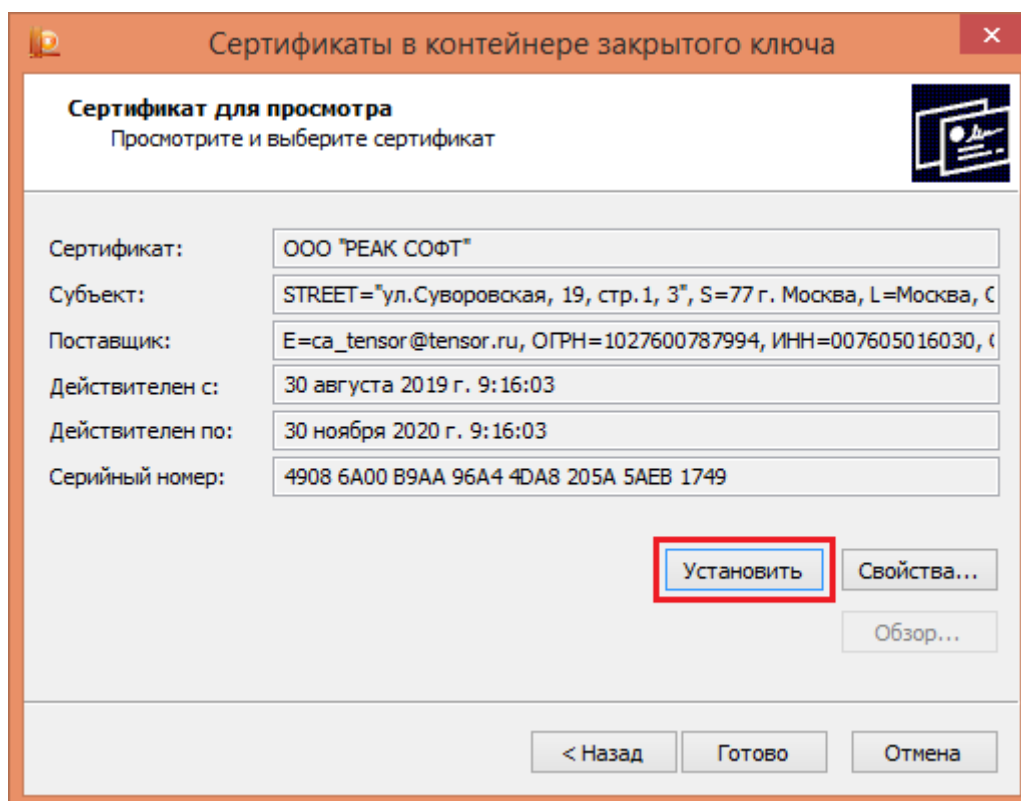
- выбрать контейнер с ключом ГОСТ Р 34.10-2012:



- в открывшемся окне убедиться, что выбран нужный ключевой контейнер и нажать далее:



- в окне со свойствами сертификата нажать на кнопку Установить:

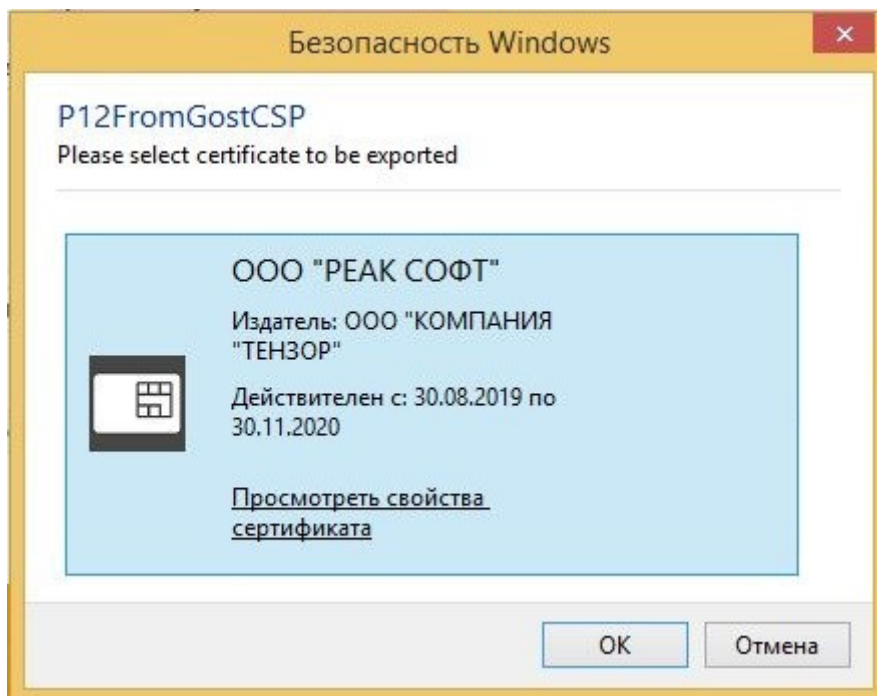


- сообщение Сертификат был установлен в хранилище «Личные» текущего пользователя свидетельствует об успешном сохранении.

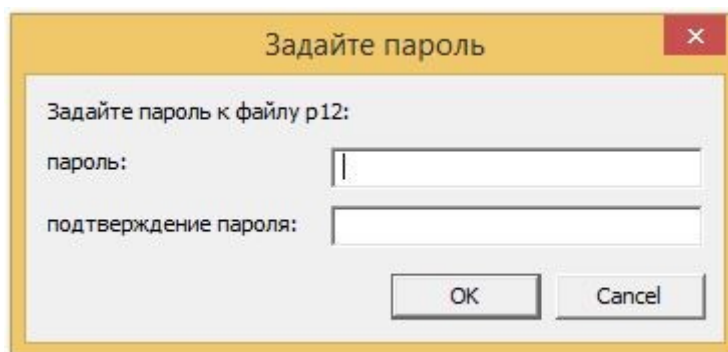
- Экспортировать ключ КристоПро из хранилища Windows в PKCS#12. Для этого приобрести и запустить утилиту P12FromGostCSP<sup>67</sup>. Нужна специальная версия утилиты – рекомендуется проконсультироваться с технической поддержкой Blitz Identity Provider.

После запуска утилиты выполнить шаги:

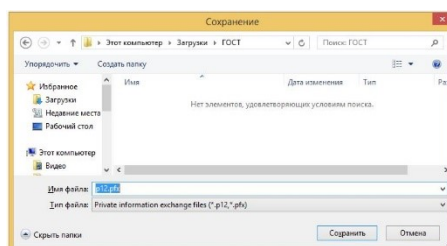
- выбрать сертификат:



- здать пароль от создаваемого контейнера PKCS#12:



- указать файл для сохранения PKCS#12. В качестве расширения обязательно указать .pfx:



- Установить на ПК окружение Java (JRE) версии 8.

<sup>67</sup> [http://soft.lissi.ru/ls\\_product/utills/p12fromcsp/](http://soft.lissi.ru/ls_product/utills/p12fromcsp/)

4. Импортировать ключ из PKCS#12 в BKS-хранилище с помощью бесплатной утилиты `gost-keytool`<sup>68</sup> и актуальной версии библиотеки `bcprov-jdk15on`<sup>69</sup>.

```
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 использования сервера рекомендуется использовать иной сервер, чем те, на которых развернут Blitz Identity Provider.
- Бесплатная утилита CryptoPro PFX Decoder by liOard.

**Важно**

Перед началом конвертации нужно сохранить контейнер ключей КриптоПро CSP в формат `pfx` (файл `sourcekey.pfx`) и сертификат открытого ключа в формат `cer` (`sourcecert.cer`).

Инструкция по конвертации контейнера:

1. Переключить ОС на использование ГОСТ в библиотеке OpenSSL согласно инструкции для используемой ОС: Astra Linux<sup>70</sup>, РЕД ОС 7.3<sup>71</sup>, Альт<sup>72</sup>. Инструкция далее приводится на примере РЕД ОС 7.3 с включенной настройкой для использования ГОСТ в OpenSSL.
2. Включить настройку ГОСТ в OpenSSL:

<sup>68</sup> <https://identityblitz.ru/wp-content/uploads/2019/11/gost-keytool.zip>

<sup>69</sup> <https://repo1.maven.org/maven2/org/bouncycastle/bcprov-jdk15on/>

<sup>70</sup> <https://wiki.astralinux.ru/pages/viewpage.action?pageId=27362269>

<sup>71</sup> <https://redos.red-soft.ru/base/manual/safe-redos/gost-in-openssl/>

<sup>72</sup> [https://www.altlinux.org/ГОСТ\\_в\\_OpenSSL](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.cyherpunks.ru/download/pyderasn-9.3.tar.zst
[fetch|wget] http://www.pyderasn.cyherpunks.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.cyherpunks.ru/pygost-5.12.tar.zst
[fetch|wget] http://www.pygost.cyherpunks.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 sourcecert.cer -inform der -out sourcecert.pem
```

7. Собрать новый `pfx`-контейнер из преобразованных ключа и сертификата:

```
openssl pkcs12 -export -in sourcecert.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`<sup>73</sup> и актуальной версии библиотеки `bcprov-jdk15on`<sup>74</sup>.

```
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
```

<sup>73</sup> <https://identityblitz.ru/wp-content/uploads/2019/11/gost-keytool.zip>

<sup>74</sup> <https://repo1.maven.org/maven2/org/bouncycastle/bcprov-jdk15on/>



**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.

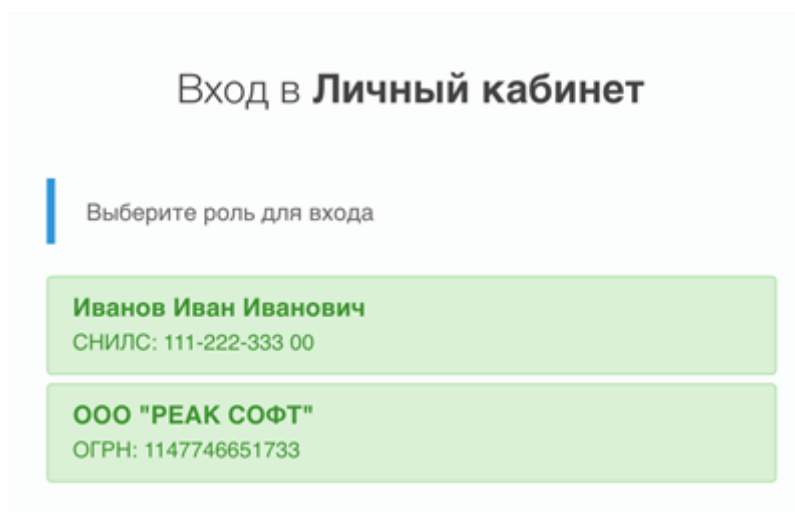
**КриптоПро JCP**

Для импорта контейнера в формате КриптоПРО JCP необходимо установить данный криптопровайдер<sup>75</sup> 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 148), 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.



- Получение из ЕСИА сведений о выбранной при входе организации, автоматическое создание на основе этих сведений в LDAP-хранилище группы пользователей с атрибутами, соответствующими орга-

<sup>75</sup> <https://cryptopro.ru/products/csp/jcp>

низации (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 196) и вход через *ЕСИА* (страница 148). 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" : "${org.inn}"
              },
              {
                "attr" : "org_fullname",
                "master" : true,
                "value" : "${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-}"
              },
            ]
          }
        }
      }
    ]
  }
}
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
{
  "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" : "${org.legalAddress-}"
}
],
"matchingRules" : [
  [
    {
      "attr" : "id",
      "value" : "${org.oid}"
    }
  ]
],
"profile" : "orgs"
},
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

    "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"
          },
          {
            "name" : "org_shortname",
            "value" : "org.shortName"
          },
          {
            "name" : "org_fullname",
            "value" : "org.fullName"
          },
          {
            "name" : "org_type",

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

        "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"
},

```

### 6.2.10 Сертификаты поставщиков WebAuthn, Passkey, FIDO2, U2F

#### ➔ См. также

- *Вход с помощью WebAuthn, Passkey, FIDO2* (страница 84)
- *Подтверждение входа с помощью WebAuthn, Passkey, FIDO2, U2F* (страница 85)

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"
    }
  ],
  ...
}
```

## 6.2.11 OAuth/OIDC, SAML и внешние поставщики идентификации

### Сервис OIDC Discovery

Blitz Identity Provider автоматически публикует сервис OIDC Discovery<sup>76</sup> 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" : {
    "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",
        ...
      },
      ...
    ]
  }
}
```

(продолжается на следующей странице)

<sup>76</sup> <https://tools.ietf.org/html/rfc8414>

(продолжение с предыдущей страницы)

```
}
}
```

### 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" : {
    ...

```

(продолжается на следующей странице)



(продолжение с предыдущей страницы)

```

"externalIdps" : {
  "idps" : [
    ...
    {
      "fedPoint" : "saml:demo-idp"
    },
    ...
  ],
  ...
},
...
}
}

```

Настроить логотип (страница 303) 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 логгута, инициированные в СУДИР из СУДИС;
- `sloServiceLocation` – адрес обработчика логгута в СУДИС при инициировании логгута 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"
        },
        ...
      ],
      ...
    },
    ...
  },
  ...
},
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

},
...
}

```

### Внешний экран согласия

Предусмотрена возможность вызова внешнего приложения, которое отображает пользователю экран согласия на доступ к его данным. Внешнее приложение может быть подключено как глобально (для всех приложений), так и для отдельных приложений. Имеется возможность настроить вызов разных внешних приложений при входе в отдельные приложения. Для использования данной возможности необходимо:

1. Добавить настройку экрана согласия, общую для всех приложений, в раздел `blitz.prod.local.idp.oauth` файла конфигурации `/usr/share/identityblitz/blitz-config/blitz.conf`.

```

"oauth":{
...
"access-token-ttl" : 1500,
...
"consent":{
  "requiredScopes":[ ],
  "autoConsent":false,
  "externalConsent" : {
    "auth" : {
      "type": "Basic",
      "userId" : "test",
      "password" : "pswd"
    },
    "url" : "https://72276dde-b44a-4d4d-8b6e-b472899bce41.mock.pstmn.io/
↪redirect"
  }
}
}
}

```

#### Описание настроек:

- `requiredScopes` - список scope, которые требуются для получения согласия пользователя. Если список пуст, scope не запрашиваются.
  - `autoConsent` - флаг, указывающий, требуется ли автоматическое согласие пользователя. Если `true`, экран согласия не отображается, и доступ предоставляется автоматически. Если `false`, означает, что экран согласия будет показан пользователю.
  - `type` - тип аутентификации, используемый для взаимодействия с внешним приложением. В данном случае используется базовая аутентификация. Аутентификация применяется при запросах от системы к внешнему приложению для передачи данных и получения ответа.
  - `userId` - идентификатор пользователя для аутентификации.
  - `password` - пароль для аутентификации.
  - `url` - URL внешнего приложения, которое отображает экран согласия.
2. При необходимости можно настроить экран согласия на выдачу разрешений отдельно для каждого приложения. Для этого необходимо добавить настройку экран согласия в конфигурационный файл в раздел `blitz.prod.local.idp.apps`. Индивидуальные настройки приложений имеют приоритет над общей настройкой.

```

"apps" : {
  "app_name" : {
    "oauth":{

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

    "consent":{
      "requiredScopes":[ ],
      "autoConsent":false,
      "activeExternalConsent":true,
      "externalConsent" : {
        "auth" : {
          "type": "Basic",
          "userId" : "test",
          "password" : "pswd"
        },
        "url" : "https://72276dde-b44a-4d4d-8b6e-b472899bce41.mock.
↳pstmn.io/redirect"
      }
    }
  }
}
}
}

```

**Описание настроек:**

- `requiredScopes` - список scope, которые требуются для получения согласия пользователя. Если список пуст, scope не запрашиваются.
  - `autoConsent` - флаг, указывающий, требуется ли автоматическое согласие пользователя. Если `true`, экран согласия не отображается, и доступ предоставляется автоматически. Если `false` означает, что экран согласия будет показан пользователю.
  - `activeExternalConsent` - флаг, указывающий, используется ли внешнее приложение для отображения экрана согласия. Если `true`, система передает управление отображением экрана согласия внешнему приложению. Если `false`, система использует встроенный механизм отображения экрана согласия, игнорируя внешнее приложение
  - `type` - тип аутентификации, используемый для взаимодействия с внешним приложением. В данном случае используется базовая аутентификация. Однако система поддерживает возможность настройки других методов аутентификации, если это требуется. Аутентификация применяется при запросах от системы к внешнему приложению для передачи данных и получения ответа.
  - `userId` - идентификатор пользователя для аутентификации.
  - `password` - пароль для аутентификации.
  - `url` - URL внешнего приложения, которое отображает экран согласия.
3. Для настройки интеграции с внешним приложением, ответственным за отображение экрана согласия, используйте параметры, описанные в разделе `invoke-consent-app`.

**6.2.12 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`.

### 6.2.13 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",
  }
}
```

- `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<sup>77</sup>.

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": [
    {
      "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",
```

(продолжается на следующей странице)

<sup>77</sup> <https://kafka.apache.org/documentation/#producerconfigs>

(продолжение с предыдущей страницы)

```

        "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<sup>78</sup>). 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%n%Exception{20}</
↪pattern>
        </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 логге:

```

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

<sup>78</sup> <https://logback.qos.ch/documentation.html>



- `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
<code>id</code>	Идентификатор записи о событии безопасности
<code>type</code>	Тип события безопасности
<code>alt_pswd_cause</code>	Причина, по которой пользователя просили сменить пароль. Возможные значения: <ul style="list-style-type: none"> <li>• <code>password_expired</code> – пароль просрочен</li> <li>• <code>password_reset</code> – пароль нужно сменить при первом входе</li> <li>• <code>password_policy_violated</code> – пароль не соответствует парольной политике</li> </ul>
<code>attr_name</code>	Имя установленного, удаленного или измененного атрибута

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Атрибут	Назначение и возможные значения
auth_methods	<p>Содержит список пройденных пользователем методов аутентификации. Возможные значения:</p> <ul style="list-style-type: none"> <li>• password – парольная аутентификация</li> <li>• spnego – вход с помощью сеанса ОС</li> <li>• x.509 – вход с помощью средства электронной подписи</li> <li>• qrCode – вход по QR-коду</li> <li>• tls – вход с помощью HTTP-заголовков прокси-сервера</li> <li>• webAuthn – вход или подтверждение входа с помощью ключей безопасности</li> <li>• css – автоматический вход по результатам регистрации пользователя или восстановления пароля</li> <li>• sms – одноразовый пароль по SMS</li> <li>• email – одноразовый пароль по email</li> <li>• hotp – второй фактор аутентификации с помощью аппаратного брелока</li> <li>• totp – второй фактор аутентификации с помощью программного TOTP-генератора кодов подтверждения</li> <li>• externalIdps:&lt;type&gt;:&lt;name&gt; – вход с помощью внешнего поставщика идентификации (соцсети и пр.)</li> <li>• userApp – вторичная аутентификация в мобильном приложении</li> <li>• outside_%NAME% – внешний метод входа с именем %NAME%</li> </ul> <p>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:</p>
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

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Атрибут	Назначение и возможные значения
client_auth_method	Способ аутентификации вызвавшего Blitz Identity Provider приложения: <ul style="list-style-type: none"> <li>• <code>internal</code> – для событий, вызванных внутренними приложениями Blitz Identity Provider</li> <li>• <code>x.509</code> – для событий, вызванных SAML-приложениями, при условии, что SAML-запрос пришел подписанным</li> <li>• <code>Basic</code> – для приложений, вызывающих REST-сервисы, использующие Basic-авторизацию</li> <li>• <code>redirectUri</code> – для приложений, которые идентифицировали себя в URL (например, указали свой <code>client_id</code> в URL-параметре), но чья аутентификация не проводилась (достоверно не известно, что это действительно вызывает Blitz Identity Provider именно это приложение)</li> <li>• <code>Bearer</code> – использование <code>access_token</code> для аутентификации мобильным приложением с динамическими <code>client_id/client_secret</code></li> </ul>
dcId	Динамический <code>client_id</code>
device	ID устройства
deviceFingerprint	Отпечаток устройства
dTyp	Тип устройства (при динамической регистрации)
email	Адрес электронной почты
entry_point	Тип интерфейса, использованного для регистрации пользователя: <ul style="list-style-type: none"> <li>• <code>WEB</code> – при регистрации из веб-приложения Blitz Identity Provider,</li> <li>• <code>REST</code> – при регистрации через REST-сервисы Blitz Identity Provider.</li> </ul>
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 ( <code>EntityId</code> для SAML или <code>client_id</code> для OIDC), вызвавшего Blitz Identity Provider.

продолжается на следующей странице

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: <ul style="list-style-type: none"> <li>• SAML – для SAML и WS-Federation</li> <li>• OAuth – для OpenID Connect и OAuth 2.0</li> <li>• simple – для прокси-аутентификации–</li> <li>• internal – для входа в Личный кабинет (_blitz_profile)</li> </ul>
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

продолжается на следующей странице

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

### 6.2.14 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 `%`.

#### **i** Примечание

Например, для приложения с идентификатором `https://example.com` должен быть создан каталог с именем `https%##example.com`.

#### **ii** 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": {
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

    "domain": "https://company.com",
    "name": "test app",
    "oauth": {
        ...
    },
    ...
}

```

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`.

### 6.2.15 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 78) 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
```

### 6.2.16 Разрешения (scopes)

Настройки разрешений (`scope`) по умолчанию управляются через вкладку OAuth 2.0 в веб-консоли администрирования.

По умолчанию разрешения хранятся в конфигурационном файле `/usr/share/identityblitz/blitz-config/blitz.conf`. При необходимости их можно хранить в отдельном файле, это упрощает управление конфигурацией - изменения в разрешениях не затрагивают основную конфигурацию, есть возможность быстрого обновления и восстановления настроек. Для того чтобы хранить разрешения в отдельном файле, добавьте раздел `blitz.prod.local.idp.scopes-source` в конфигурационный файл `blitz.conf`. При этом необходимо указать значение `file` в параметре `type`. Путь к файлу указывается в параметре `file`. Если путь не указан, используется путь по умолчанию: `$app-data-path/scope.json`, где `$app-data-path` — это путь, заданный в конфигурационном файле `boot.conf`.

Пример настройки для хранения отдельным файлом:

```

"scopes-source" : {
  "file" : "/home/test/work/blitz-idp/conf/scopes.json",
  "type" : "file"
}

```

Если нужно обратно переключиться на основную конфигурацию, используйте значение `config` для параметра `type` в разделе конфигурационного файла `blitz.prod.local.idp.scopes-source`.

Пример настройки для хранения в основном конфигурационном файле:

```
"scopes-source" : {
  "file" : "/usr/share/identityblitz/blitz-config/blitz.conf",
  "type" : "config"
}
```

Настройки хранилища отдельным файлом:

```
{
  "scope_1" : {
    "sysname" : "scope_1",
    "names" : {
      "default" : "scope_1"
    },
    "descriptions" : {
      "default" : "desc_1"
    },
    "client-level" : false,
    "enabled" : true,
    "depricated" : false,
    "allowedResponseTypes" : [ "code", "token", "id_token", "device_code" ],
    "allowedGrantTypes" : [ "client_credentials", "password", "authorization_code",
↔ "implicit", "refresh_token", "urn:ietf:params:oauth:grant-type:token-exchange",
↔ "urn:ietf:params:oauth:grant-type:device_code" ],
    "attrs" : [ "org_id" ]
  },
  "scope_2" : {
    "sysname" : "scope_2",
    "names" : {
      "default" : "scope_2"
    },
    "descriptions" : {
      "default" : "desc_2"
    },
    "client-level" : false,
    "enabled" : true,
    "depricated" : false,
    "allowedResponseTypes" : [ "code", "token", "id_token", "device_code" ],
    "allowedGrantTypes" : [ "password", "urn:ietf:params:oauth:grant-type:device_
↔ code", "urn:ietf:params:oauth:grant-type:token-exchange", "refresh_token",
↔ "implicit", "client_credentials", "authorization_code" ],
    "attrs" : [ "testClaim" ]
  }
}
```

Свойства разрешений (scopes):

#### Базовые:

- `sysname` - системное имя scope;
- `names` - название scope. Если имя не задано, то используется `sysname`. На данный момент используется только значение `default`.
- `desc` - описание scope на разных языках. На данный момент используется только значение `default`.

#### Для OAuth:

- `clientLevel` - является ли scope системным (`true/false`);



- `allowedResponseTypes` - по умолчанию разрешено все: `code`, `token`, `id_token`, `device_code`;
- `allowedGrantTypes` - по умолчанию разрешено все: `authorization_code`, `implicit`, `password`, `client_credentials`, `refresh_token`, `urn:iETF:params:oauth:grant-type:token-exchange`, `urn:iETF:params:oauth:grant-type:device_code`

#### По доступу к данным:

- `attrs` - атрибуты пользователя доступные для `scope`.

#### По жизненному циклу:

- `enabled` - активирован ли `scope`. Если `false`, то даже по ранее выданным токенам будет отказ. По умолчанию `true`;
- `deprecated` - параметр указывает, что запрос `scope` больше недоступен, однако токены, полученные ранее с этим параметром, продолжают работать. По умолчанию `false`.

#### Примечание

Для настройки разрешений (`scope`) для доступа к сервисам REST API см. раздел *REST API* (страница 361). В данном разделе описывается процесс изменения набора разрешений по умолчанию, а также настройка доступа к атрибутам, связанными с этими разрешениями.

## 6.2.17 REST API

Blitz Identity Provider позволяет настраивать разрешения (`scopes`) для доступа к сервисам REST API. В данном разделе описывается процесс изменения набора разрешений по умолчанию, а также настройка доступа к атрибутам, связанным с этими разрешениями.

Ниже приведен список сервисов Blitz Identity Provider, системные имена сервисов и разрешения (`scope`) по умолчанию.

### Административные REST API

Сервис	Путь к сервису	Системное имя	Разрешения
Получение информации о приложении	GET /blitz/admin/api/v3/	admin.app.get	blitz_api_sys_app
Создание нового приложения	PUT /blitz/admin/api/v3/	admin.app.create	blitz_api_sys_app_chg
Изменение параметров приложения	POST /blitz/admin/api/v3/	admin.app.change	blitz_api_sys_app_chg
Удаление приложения	DELETE /blitz/admin/api/v3/	admin.app.delete	blitz_api_sys_app_chg

### Регистрация REST API v3

Сервис	Путь к сервису	Системное имя	Разрешения
Регистрация пользователя	PUT /sps/reg/api/v3/user	reg.put	blitz_api_sys_users_reg
Регистрация пользователя (подтверждение контактов)	POST /sps/reg/api/v3/user	reg.advance	blitz_api_sys_users_reg

**Аутентификация по QR-коду REST API v3**

Сервис	Путь к сервису	Системное имя	Разрешения
Показать QR-код для аутентификации по QR-коду	GET /sps/api/v3/auth/qr/	auth.qrcode.get	blitz_qr_auth
Подтвердить аутентификацию по QR-коду	POST /sps/api/v3/auth/qr/	auth.qrcode.refuse	blitz_qr_auth
Отклонить аутентификацию по QR-коду	POST /sps/api/v3/auth/qr/	auth.qrcode.confirm	blitz_qr_auth

**Работа с правами доступа REST API v3**

Сервис	Путь к сервису	Системное имя	Разрешения
Получить права субъекта	GET /sps/api/v3/rights/of GET /sps/api/v3/rights/of	rights.sbj.get	blitz_user_rights, blitz_rights_full_access
Получить права на объект	GET /sps/api/v3/rights/of GET /sps/api/v3/rights/of	rights.obj.get	blitz_rights_full_access
Изменить права	POST /sps/api/v3/rights	rights.change	blitz_user_rights, blitz_rights_full_access
Назначить права	PUT /sps/api/v3/rights	rights.grant	blitz_user_rights, blitz_rights_full_access
Отозвать права	DELETE /sps/api/v3/rights	rights.revoke	blitz_user_rights, blitz_rights_full_access

## Работа с пользователями REST API v3

Сервис	Путь к сервису	Системное имя	Разрешения
Выполнение системных команд в контексте пользователя	POST /blitz/api/v3/users/n	users.docommand	blitz_api_user_chg, blitz_api_sys_users_chg
Получение данных пользователя	GET /sps/api/v3/users/:su	users.attrs.get	blitz_api_user, blitz_api_sys_users
Изменение атрибутов пользователя	POST /sps/api/v3/users/:in	users.attrs.change	blitz_api_user_chg, blitz_api_sys_users_chg
Смена пароля	POST /sps/api/v3/users/:in	users.pswd.change	blitz_api_usec_chg, blitz_api_sys_usec_chg
Сброс сессий	POST /sps/api/v3/users/:su	users.sessions.reset	blitz_api_usec_chg, blitz_api_sys_usec_chg
Изменение аутентификационной информации	POST /sps/api/v3/users/:su	users.auth.change	blitz_api_usec_chg, blitz_api_sys_usec_chg
Изменение свойств пользователя	GET /sps/api/v3/users/:su	users.props.get	blitz_api_user, blitz_api_sys_users
Изменение признака блокировки	POST /sps/api/v3/users/:su	users.props.change	blitz_api_user_chg, blitz_api_sys_users_chg
Изменение состояния блокировки пользователя	POST /blitz/api/v3/users/:s	users.lockState.chan	blitz_api_usec_chg, blitz_api_sys_usec_chg
Получение контрольного вопроса	GET /sps/api/v3/users/:su	users.SecQuestion.ge	blitz_api_usec_chg, blitz_api_sys_usec_chg
Изменение контрольного вопроса/ответа	POST /sps/api/v3/users/:su	users.SecQuestion.cf	blitz_api_usec_chg, blitz_api_sys_usec_chg
Удаление контрольного вопроса	DELETE /sps/api/v3/users/:su	users.SecQuestion.de	blitz_api_usec_chg, blitz_api_sys_usec_chg
Проверка контрольного вопроса	POST /sps/api/v3/users/:su	users.SecQuestion.cf	blitz_api_usec, blitz_api_sys_usec

## Работа с приложениями пользователя REST API v3

Сервис	Путь к сервису	Системное имя	Разрешения
Посмотреть устройства входа	GET /sps/api/v3/users/:su	users.userAgents.get	blitz_api_uapps, blitz_api_sys_uapps
Удалить устройство входа	DELETE /sps/api/v3/users/:su	users.userAgents.del	blitz_api_uapps_chg, blitz_api_sys_uapps_chg
Посмотреть мобильные устройства пользователя	GET /sps/api/v3/users/:su	users.apps.get	blitz_api_uapps, blitz_api_sys_uapps
Удалить устройство пользователя	DELETE /sps/api/v3/users/:su	users.apps.delete	blitz_api_uapps_chg, blitz_api_sys_uapps_chg

## Работа со списками разрешений, выданных пользователем и аудит REST API v3

Сервис	Путь к сервису	Системное имя	Разрешения
Посмотреть разрешения пользователя	GET /sps/api/v3/users/:su	users.acls.get	blitz_api_usec, blitz_api_sys_usec
Удалить разрешение пользователя	DELETE /sps/api/v3/users/:su	users.acls.delete	blitz_api_usec_chg, blitz_api_sys_usec_chg
Посмотреть события пользователя	GET /sps/api/v3/users/:su	users.audit.get	blitz_api_uaud, blitz_api_sys_uaud

### Пользовательские привязки TOTP REST API v3

Сервис	Путь к сервису	Системное имя	Разрешения
Посмотреть привязки TOTP-генераторов	GET /sps/api/v3/users/:su	users.totp.get	blitz_api_usec, blitz_api_sys_usec
Показать QR-код для привязки TOTP-генератора	GET /sps/api/v3/users/:su	users.totp.qrGen	blitz_api_usec_chg, blitz_api_sys_usec_chg
Привязать TOTP-генератор	POST /sps/api/v3/users/:su	users.totp.attach	blitz_api_usec_chg, blitz_api_sys_usec_chg
Отвязать TOTP-генератор	DELETE /sps/api/v3/users/:su	users.totp.detach	blitz_api_usec_chg, blitz_api_sys_usec_chg

### Пользовательские привязки внешних поставщиков REST API v3

Сервис	Путь к сервису	Системное имя	Разрешения
Посмотреть привязки внешних поставщиков	GET /sps/api/v3/users/:su	users.fed.get	blitz_api_ufa, blitz_api_sys_ufa
Привязать внешний поставщик	POST /sps/api/v3/users/:su	users.fed.bind	blitz_api_ufa_chg, blitz_api_sys_ufa_chg
Отвязать внешний поставщик	DELETE /sps/api/v3/users/:su	users.fed.unbind	blitz_api_ufa_chg, blitz_api_sys_ufa_chg
Получить маркер доступа внешнего поставщика	GET /sps/api/v3/users/:su	users.fedToken.get	fed_tkn_any

Чтобы завести новое разрешение для одного из указанных сервисов, необходимо добавить в конфигурационный файл новый блок `blitz.prod.local.idp.rest`. В него добавить следующие параметры:

- `defaultEnabled` – принимает значение `true`, если нужно, чтобы разрешения по умолчанию продолжали работать;
- `points` – перечень сервисов (используется системное имя из таблицы выше), для которых необходимо предусмотреть новые разрешения.

Настройка разрешений для каждого сервиса определяется следующими параметрами:

- `enabled` – признак необходимости включить настройку (`true/false`);
- `access` – блок из следующих настроек:
  - `methods` – методы авторизации к сервису: `oauth` (авторизация по маркеру доступа) и `basic` (Basic-авторизация);
  - `filter` – способы фильтрации атрибутов: `disabled` (возвращаются все атрибуты), `strict` (атрибуты возвращаются при условии, что все возвращаемые атрибуты разрешены системе) и `flexible` (возвращаются только те атрибуты, приписанные данному разрешению);
  - `scopes` – список разрешений для данного сервиса;
  - `oauth` – перечень приложений, которым разрешен доступ к сервису при предъявлении маркера доступа (если параметр отсутствует, то сервис разрешен всем приложениям, которым выдан маркер доступа с данным разрешением);
  - `basic` – перечень приложений, которым разрешен доступ к сервису с помощью Basic-авторизации (если параметр отсутствует, то сервис разрешен всем приложениям).

Пример настройки разрешений (`blitz_custom_api_sys_users_chg`, `blitz_api_sys_users_chg`, `blitz_api_user_chg`) для сервиса изменения атрибутов пользователя (`users.attrs.change`):

```
"rest" : {
  "defaultEnabled" : true,
  "points" : {
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

"users.attrs.change" : {
  "access" : {
    "filter" : "flexible",
    "methods" : [
      "oauth"
    ],
    "scopes" : [
      "blitz_custom_api_sys_users_chg",
      "blitz_api_sys_users_chg",
      "blitz_api_user_chg"
    ]
  },
  "enabled" : true
}
},
}
},

```

**i Примечание**

Для получения сведений о механизмах хранения разрешений (*scope*) обратитесь к разделу *Разрешения (scopes)* (страница 359). Хранение разрешений возможно в конфигурационном или отдельном файле. Хранение в отдельном файле упрощает управление конфигурацией: изменения в разрешениях не затрагивают основную конфигурацию, есть возможность быстрого обновления и восстановления настроек.

## 6.3 Настройки консоли управления

Консоль управления настраивается с помощью файлов `console.conf` и `credentials`. Далее в подразделах описаны возможные настройки.

### 6.3.1 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 57) 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:

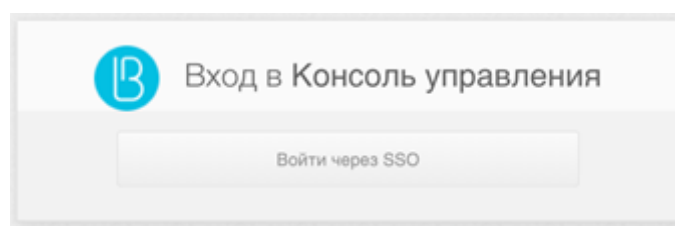
```

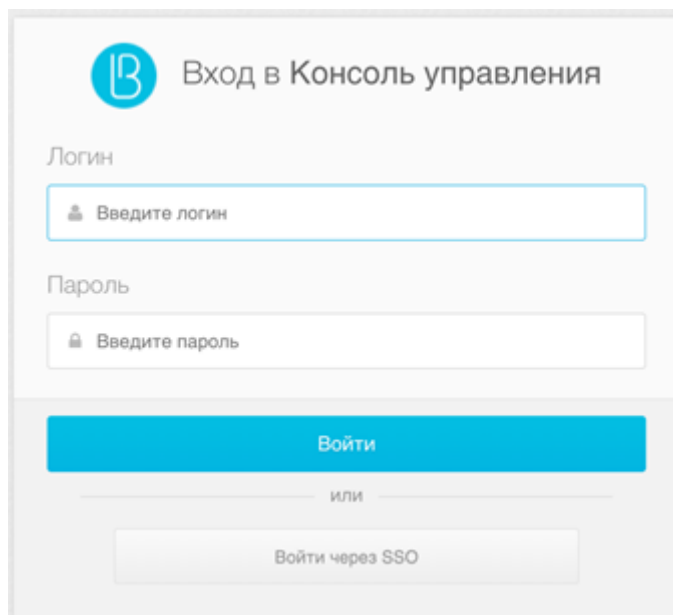
{
  "login" : {
    "fp" : {
      "authUri" : "https://idp-host.com/blitz/oauth/ae",
      "clientId" : "blitz-console",
      "clientSecret" : "client_secret_value",
      "logoutUrl" : "https://idp host.com/blitz/login/logout?post_logout_redirect_
↵uri=https://idp host.com/blitz/console",
      "scopes" : [
        "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`.

### 6.3.2 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 244).

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,
  ...
}
```

### 6.3.3 Роли и права доступа в консоль

Стандартные роли администраторов описаны в *ранее* (страница 56). 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 294). 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)

### 6.3.4 Смена пароля администратора

Для изменения пароля администратора консоли выполните следующие действия:

1. Откройте файл `/usr/share/identityblitz/blitz-config/credentials`.
2. Укажите новый пароль в параметре `pswdHash`. Пароль указывается в открытом виде без шифрования. После применения изменений система его зашифрует.

```
"users" : [
  {
    "pswdHash" : "new$password",
    "roles" : [
      "root"
    ],
    "username" : "admin"
  }
]
```

3. Перезапустите сервис `blitz-console`.

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



### 6.3.5 Работа с тегами приложений

В системе существует удобный механизм работы с тегами приложений, который позволяет администраторам эффективно управлять тегами и назначать их приложениям. Для этого предусмотрены следующие возможности:

1. Выбор из ранее заведенных тегов:
  - При назначении тега приложению можно выбирать тег из списка ранее созданных. Это исключает необходимость запоминания и ручного ввода тегов.
  - Если требуемый тег отсутствует в списке, есть возможность создать новый тег непосредственно в конфигурационном файле консоли `console.conf`.
2. Набор тегов по умолчанию:
  - В системе предусмотрен набор тегов по умолчанию, которые не назначаются автоматически ни одному приложению.
  - Существует возможность назначать эти теги приложениям через консоль.
3. Хранение тегов:
  - Теги хранятся в отдельном блоке конфигурационного файла консоли `console.conf`.
  - Редактирование общего списка тегов через интерфейс консоли недоступно. Для удаления тега из общего списка необходимо сначала удалить его у всех приложений, после чего вручную удалить его конфигурационного файла консоли `console.conf` из блока `ui`.

Для создания нового тега необходимо в конфигурационном файле консоли `console.conf` добавить блок `ui`, в котором определены настройки пользовательского интерфейса для работы с приложениями. Внутри раздела `apps` находится подраздел `commonTags`, содержащий список тегов.

Пример конфигурации:

```
"ui": {
  "apps": {
    "commonTags": ["COMMON_TAG1", "COMMON_TAG2", "COMMON_TAG3"]
  }
}
```

## 6.4 Настройка Token Exchange

Blitz Identity Provider поддерживает технологию обмена маркеров доступа OAuth 2.0 Token Exchange<sup>79</sup>. Стандартным применением данной технологии в Blitz Identity Provider является взаимодействие шлюза безопасности Blitz Keeper с сервисом авторизации для получения доступа к защищаемым сервисам.

Для настройки Token Exchange выполните описанную ниже последовательность действий.

### 6.4.1 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": {
```

(продолжается на следующей странице)

<sup>79</sup> <https://tools.ietf.org/html/rfc8693>

(продолжение с предыдущей страницы)

```

    "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": {
    "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, выдан-

ного 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"
  }
}
]
```

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 196)).

Пример 2 правила:

```
"userRights": [
{
  "rights": ["security_administrator"],
  "target": {
    "type": "grps",
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

    "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 отличие 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 225));

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.

## 6.4.2 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 369). 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`.

## Глава 7

# Безопасность, обслуживание и устранение неисправностей

### 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.

#### Просмотр событий

**Значение**

**Период**

**Группа событий**

Вход  Выход  Авторизация доступа

Изменение аутентификационных данных

Изменения учетной записи  Операции с группами

Отправка кодов подтверждения

Администрирование

**Протокол**

OAuth 2.0  SAML  Другие

Показать  Экспортировать в CSV

ID процесса	Время	Событие	Субъект	Объект	Приложение	IP-адрес
1bae0d7a...	23.12.2024 10:45:04	Выполнен вход	admin	admin	Консоль управления	172.25.0.88
0ea4cae1...	23.12.2024 10:44:20	Выполнен вход	admin	admin	Консоль управления	172.25.0.88

Также есть возможность выгрузки событий безопасности в формате CSV.

**Просмотр событий**

**Значение**

**Период**

**Группа событий**

Вход  Выход  Авторизация доступа

Изменение аутентификационных данных

Изменения учетной записи  Операции с группами

Отправка кодов подтверждения

Администрирование

**Протокол**

OAuth 2.0  SAML  Другие

Показать  Экспортировать в CSV

Количество событий

## 7.2 Мониторинг функционирования приложений

### 7.2.1 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<sup>80</sup>.

#### 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",} 9.223372036854776E18
# 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
```

(продолжается на следующей странице)

<sup>80</sup> <https://prometheus.io/>



(продолжение с предыдущей страницы)

```

# 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
# 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.
↪6869999999999998
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

```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
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
<code>uptime_seconds</code>	<code>gauge</code>	Время с момента запуска приложения (в секундах)
<code>licence_exp_seco</code>	<code>gauge</code>	Время до истечения срока действия лицензии (в секундах)
<code>config_mtime</code>	<code>gauge</code>	Время последнего изменения файла конфигурации
<code>datasource_laten</code>	<code>histogr</code>	Задержки ответа от хранилища УЗ по операциям чтения и записи (могут быть типы <code>ldap</code> , <code>jdbс</code> , <code>couch</code> )
<code>mq_connections</code>	<code>gauge</code>	Количество коннектов к MQ ( <code>rmq</code> , <code>kafka</code> )
<code>mq_latency</code>	<code>histogr</code>	Задержки ответа от MQ ( <code>rmq</code> , <code>kafka</code> )
<code>authn_method_a</code>	<code>countei</code>	Количество успешных и неуспешных аутентификаций каждым методом входа в различные приложения
<code>authn_method_t</code>	<code>countei</code>	Общее кол-во успешных и нет аутентификаций разными методами
<code>authn_method_la</code>	<code>histogr</code>	Длительность аутентификации по разным методам входа
<code>datasource_conn</code>	<code>gauge</code>	Кол-во коннектов к хранилищам
<code>version</code>	<code>gauge</code>	Версия приложения
<code>notify_user_total</code>	<code>countei</code>	Кол-во направленных сообщений по разным каналам
<code>authn_method_app_created</code>	<code>counter</code>	Эти метрики (с суффиксом <code>_created</code> ) генерируются в связи с особенностями Prometheus и содержат время в <code>unix timestamp</code> момента создания метрики
<code>authn_method_created</code>	<code>counter</code>	
<code>authn_method_latency_created</code>	<code>counter</code>	
<code>datasource_latency_created</code>	<code>counter</code>	
<code>mq_latency_created</code>	<code>counter</code>	
<code>notify_user_created</code>	<code>counter</code>	
<code>notify_user_created</code>	<code>counter</code>	

## 7.2.2 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.

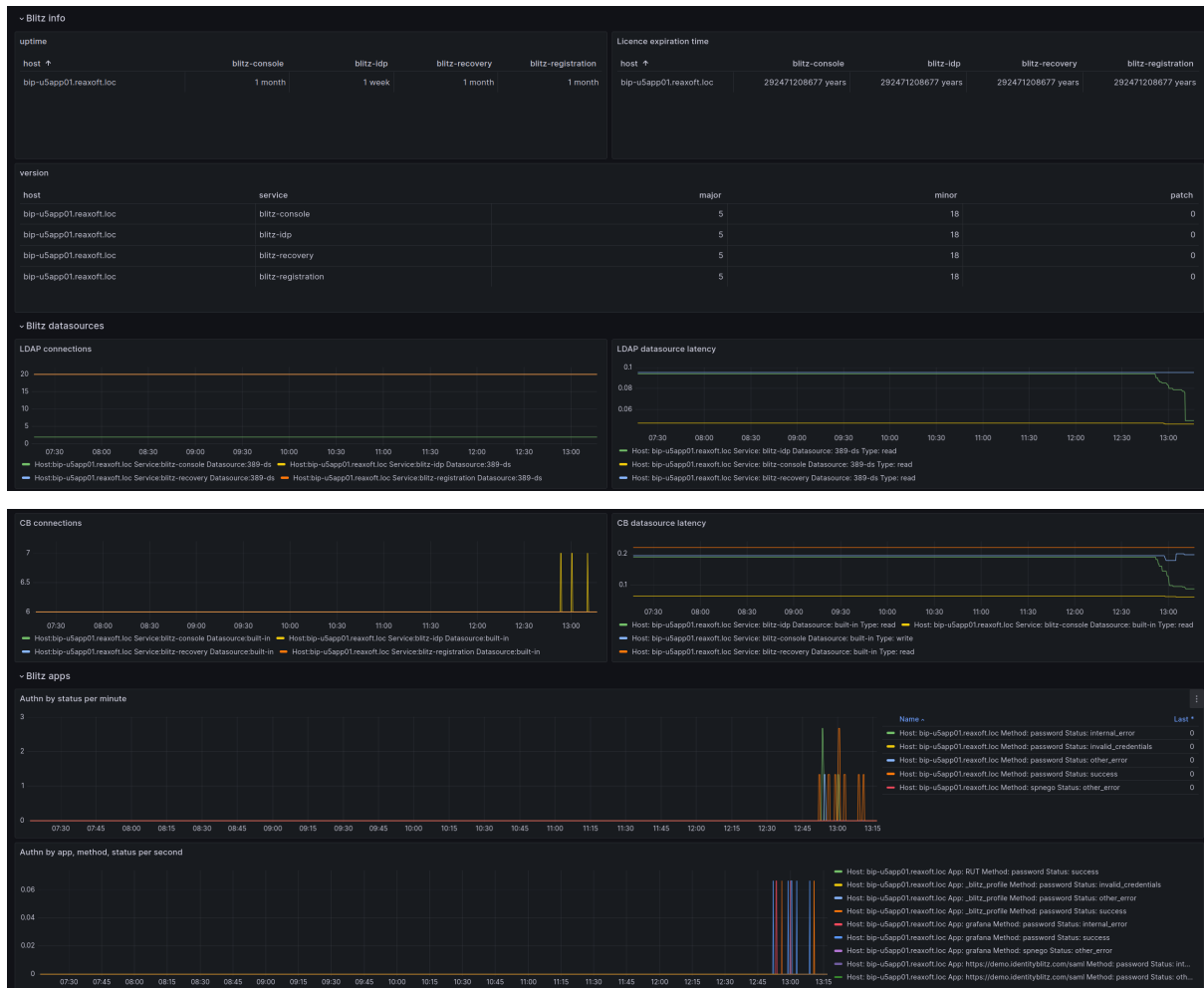
Для настройки визуализации выполните следующие действия:

1. Модифицируйте job-задание `prometheus.yaml` в соответствии с конфигурацией своей системы и добавьте его в Prometheus<sup>81</sup>.

<sup>81</sup> <https://prometheus.io>

2. Модифицируйте шаблон дашборда `blitz-dashboard.json`. Настройте Grafana и добавьте дашборд<sup>82</sup>.

Примеры визуализации данных в Grafana:



## 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:

<sup>82</sup> <https://prometheus.io/docs/visualization/grafana/>

```

"levels" : {
  "ROOT" : "TRACE",
  "application" : "TRACE",
  "com.couchbase.client" : "INFO",
  "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: 0:0:0:0:0:0:1
# user agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
#####
↪#####

```

## 7.4 Шлюз безопасности

В качестве шлюза безопасности используется отдельно устанавливаемый модуль Blitz Keeper.

## Глава 8

# 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 проху 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

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Группа функций	Функции
	Поддержка единого SSO с приложениями IBM, использующими для единого входа Ltpa2Token
<b>Идентификация и аутентификация</b>	
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 логута
	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): <ul style="list-style-type: none"> <li>• проверка CAPTCHA (reCAPTCHA или иной сервис, выбранный Заказчиком)</li> <li>• временное блокирование входа по паролю учетной записи при выявленных попытках перебора</li> <li>• замедление входа пользователя (задержка входа, решение браузером вычислительно сложной задачи – Proof of Work)</li> </ul>
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

продолжается на следующей странице

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: <ul style="list-style-type: none"> <li>• Яндекс, ВКонтакте, Одноклассники, Mail ID, VK ID, Apple ID, Google, Facebook<sup>1</sup></li> <li>• Вход через ЕСИА в режиме физического лица</li> <li>• Вход через ЕСИА в режиме представителя организации (с выбором организации при входе)</li> <li>• Вход через ЕСИА в режиме цифрового профиля</li> <li>• Вход через банки Сбер ID, Т-ID, ВТБ ID, Сбер-Бизнес ID, Альфа ID</li> <li>• Вход через Mos ID (СУДИР)</li> <li>• Вход через СУДИС</li> </ul>
	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 платформу-независимых ключей безопасности FIDO2

продолжается на следующей странице

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	<p>Вход с помощью средств квалифицированной электронной подписи</p> <p>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</p> <p>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</p> <p>Поддерживаемые браузеры: Internet Explorer 11, Chrome, Firefox, Yandex, Спутник</p> <p>Возможность сопоставления/регистрации учетных записей в процессе первичного входа на основе данных из сертификата квалифицированной электронной подписи</p> <p>Возможность проверки действительности подписи/сертификата встроенными возможностями ПО</p> <p>Возможность проверки действительности подписи/сертификата через вызов внешнего сервиса проверки</p>
Двухфакторная аутентификация	<p>Подтверждение входа разовым паролем из SMS (SMS-шлюз предоставляет Заказчик)</p> <p>Подтверждение входа разовым паролем из email</p> <p>Подтверждение входа разовым паролем TOTP-приложения (RFC 6238 “TOTP: Time-Based One-Time Password Algorithm”)</p> <p>Подтверждение входа разовым паролем из аппаратного брелока. Поддержка брелоков HOTP (RFC 4226 “HOTP: An HMAC-Based One-Time Password Algorithm”). Брелоки предоставляет Заказчик</p> <p>Подтверждение входа ключом безопасности WebAuthn, Passkey, FIDO2</p> <p>Подтверждение входа ключом безопасности U2F</p> <p>Подтверждения входа разовым паролем в push-уведомлении в мобильном приложении Заказчика (сервис для отправки push и мобильное приложение предоставляет Заказчик)</p> <p>Подтверждение входа по входящему звонку (Flash Call)</p>
Другое	<p>Возможность Заказчику самостоятельно добавить собственный метод аутентификации</p> <p>Возможность Заказчику самостоятельно настроить внешний вид страницы входа отдельно для каждого приложения, в которое осуществляется вход</p> <p>Предоставление API, позволяющее мобильным приложениям зарегистрировать событие входа и получить маркеры безопасности при входах с использованием ПИН-кода, Touch ID, Face ID</p>

продолжается на следующей странице



Таблица 1 – продолжение с предыдущей страницы

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

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Группа функций	Функции
Настройки безопасности учетной записи	<p>Веб-приложение, позволяющее пользователю управлять настройками безопасности его учетной записи:</p> <ul style="list-style-type: none"> <li>• возможность самостоятельно сменить пароль;</li> <li>• 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;</li> <li>• возможность настроить двухфакторную аутентификацию для своей учетной записи;</li> <li>• возможность посмотреть/отредактировать список запомненных устройств, привязанных учетных записей внешних поставщиков входа;</li> <li>• возможность посмотреть события безопасности со своей учетной записью.</li> </ul>
	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

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Группа функций	Функции
	<p>Возможность запросить у пользователя заполнение текстового атрибута</p> <p>Возможность в момент входа задать контрольный вопрос</p> <p>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)</p>
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	<p>Задание правил входа пользователей в приложения посредством процедур входа и регистрации</p> <p>Кастомизация операций с хранилищами данных</p>
Мониторинг и аудит	
Оповещения пользователей о событиях безопасности	<p>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</p> <p>Возможность настроить набор событий оповещения и тексты оповещений для SMS и для email</p>
Регистрация событий безопасности	<p>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</p> <p>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</p> <p>Сопоставление IP-адресам геоданных в событиях и уведомлениях (БД в формате mmdb с геоданными предоставляет Заказчик)</p> <p>Интерфейс администратора для поиска/просмотра событий безопасности</p> <p>Запись событий безопасности: в БД, в лог-файл, в Kafka</p>
Мониторинг	Возможность в момент входа пользователя вызывать системы сбора метрик и статистики, антифрод системы

продолжается на следующей странице

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	

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Группа функций	Функции
Администрирование	<p>Веб-приложение администрирования:</p> <ul style="list-style-type: none"> <li>• задание настроек подключенных приложений (параметры приложений, разрешенные режимы взаимодействия, правила контроля доступа)</li> <li>• настройка атрибутов пользователей и сопоставление атрибутов хранилищам учетных записей</li> <li>• настройка подключения к хранилищам учетных записей на основе LDAP</li> <li>• настройка подключения к произвольным хранилищам (через предоставленный Заказчиком сервис)</li> <li>• поддержка работы одновременно с несколькими хранилищами учетных записей</li> <li>• настройка методов идентификации/аутентификации и внешних поставщиков входа</li> <li>• настройка подключения к SMTP-службе и к SMS-шлюзу</li> <li>• поддержка ролевого доступа для входа в веб-приложение администратора. Возможность для разных пользователей задать разный набор доступных действий</li> <li>• управление настройками веб-приложений регистрации, управления настройками безопасности, восстановления пароля</li> <li>• 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)</li> <li>• администрирование групп пользователей, управления членством пользователей в группах</li> <li>• настройка внешнего вида страниц входа в приложения</li> <li>• просмотр и фильтрация зарегистрированных событий безопасности</li> <li>• возможность входа в веб-приложение администрирования через SSO</li> </ul> <p>Интерфейс администратора на русском и английском языках</p> <p>Возможность добавления переводов на дополнительные языки</p>

<sup>1</sup> Meta признана экстремистской организацией и запрещена в России, деятельность принадлежащих ей соц.сетей Facebook и Instagram также запрещена в РФ.

## Глава 9

# Рекомендации ФСТЭК

Ниже приведены рекомендации по обеспечению защиты информации согласно требованиям ФСТЭК.

### 9.1 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 76).
- Для администраторов консоли управления *настроить вход через Blitz* (страница 365).
- Настроить через процедуры входа для администраторов и пользователей *требования к прохождению двухфакторной аутентификации* (страница 244).
- *Задать* (страница 207) 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 63), который будет использоваться в качестве идентификатора учетной записи.
- Настроить запрет на *повторное использование идентификатора после удаления учетной записи* (страница 328) и *блокирование неактивных учетных записей* (страница 328).

#### 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 304) парольную политику.
- *Управлять* (страница 186) 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 148), *СУДИР* (страница 166), *другая установка продукта* (страница 168).

## 9.2 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 59) атрибут с типом учетной записи и *настроить* (страница 244) политику доступа пользователей в приложении.
- Для временных учетных записей *настроить* (страница 367) правила блокирования входа по истечении срока действия записей.
- Для использования функций работы с группами пользователей *настроить* (страница 196) группы пользователей.
- *Управлять* (страница 186) 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 304), *разрешения* (страница 225).
- Установить *разрешения приложений* (страница 220).
- Настроить шлюз безопасности.
- Настроить *атрибуты пользователей* (страница 59) и *группы пользователей* (страница 196), *процедуры входа в приложения* (страница 244).

**упд.4**

Разделение полномочий (ролей) пользователей, администраторов и лиц, обеспечивающих функционирование информационной системы

Рекомендация по настройке:

- Использовать 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 56) роли администраторов.
- *Управлять* (страница 186) атрибутами пользователей.
- Назначать права доступа с использованием сервисов.

**упд.6**

Ограничение неуспешных попыток входа в информационную систему (доступа к информационной системе)

Рекомендация по настройке:

- *Настроить* (страница 87) 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 220), *Общие настройки OAuth 2.0* (страница 225), *Настройки текстов интерфейса* (страница 294).

**упд.8**

Оповещение пользователя после успешного входа в информационную систему о его предыдущем входе в информационную систему)

Рекомендация по настройке:

- *Настроить* (страница 182) 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 367) 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 76) период неактивности.



## 9.3 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 375) 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 379) журналов событий на предмет возникновения ошибок.

### РСБ.5

Мониторинг (просмотр, анализ) результатов регистрации событий безопасности и реагирование на них

Рекомендация по настройке:

- *Просмотр* (страница 375) 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 9) резервное копирование СУБД

### РСБ.8

Обеспечение возможности просмотра и анализа информации о действиях отдельных пользователей в информационной системе

Рекомендация по настройке:

- *Просмотр* (страница 375) событий безопасности периодически осуществлять в консоли управления.