ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «BLITZ IDENTITY PROVIDER»

Версия 5.21

РУКОВОДСТВО АДМИНИСТРАТОРА

1147746651733.62.01.000.001.И1

Москва, 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ **7**

1. ПОДГОТОВКА К УСТАНОВКЕ **8**

1.1. Минимальные требования к развертыванию **8**

1.2. Рекомендуемые требования к развертыванию в кластере **10**

2. УСТАНОВКА **14**

2.1. Общая инструкция по установке **14**

2.1.1. Установка JDK 14
2.1.2. Установка memcached 16
2.1.3. Установка и настройка СУБД 16
2.1.4. Установка и настройка сервера очередей RabbitMQ 20
2.1.5. Установка приложений Blitz Identity Provider 20
2.1.6. Настройка опций запуска приложений Blitz Identity Provider 23
2.1.7. Настройка синхронизации файлов конфигурации 26
2.1.8. Настройка веб-сервера 28
2.1.9. Установка и настройка LDAP-каталога 29
2.1.10. Вход в консоль управления 30
2.1.11. Установка лицензионного ключа 32
2.1.12. Управление учетными записями администраторов 33
2.1.13. Перезапуск приложений Blitz Identity Provider 34
2.1.14. Рекомендуемые действия после первого запуска Blitz Identity Provider 34

2.2. Экспресс-инструкции по установке **35**

2.2.1. Astra Linux Special Edition 1.7 35
2.2.2. Альт 8 СП Сервер 38
2.2.3. Альт Сервер 10 40
2.2.4. ОСнова 2.5.1 43
2.2.5. Red OS 7.3 46
2.2.6. Rocky Linux 8, AlmaLinux 8, Oracle Linux 8, RHEL 8 48
2.2.7. Rocky Linux 9, AlmaLinux 9, Oracle Linux 9, RHEL 9 52

3. НАСТРОЙКА АТРИБУТОВ УЧЕТНЫХ ЗАПИСЕЙ **56**

3.1. Конфигурирование доступных атрибутов **56**

3.1.1. Настройка хранимых атрибутов 56
3.1.2. Настройка вычисляемых атрибутов 58
3.1.3. Настройка правил преобразования входных значений 59
3.1.4. Настройка правил преобразования выходных значений 60
3.1.5. Настройка назначения атрибутов 60

3.2. Подключение хранилищ атрибутов **62**

3.2.1. Типы хранилищ 62
3.2.2. Подключение хранилища по протоколу LDAP 63
3.2.3. Подключение к хранилищу по REST 67
3.2.4. Настройка внутреннего хранилища 74

4. НАСТРОЙКИ АУТЕНТИФИКАЦИИ **75**

4.1. Настройка парольной политики **77**

4.2. Настройка ключей безопасности **78**

4.3. Настройка доступных методов аутентификации **80**

4.4. Настройка входа по логину и паролю **82**

4.5. Настройка входа с помощью средства электронной подписи **87**

4.5.1. Настройка метода аутентификации в консоли управления 87
4.5.2. Использование и обновление плагина 89

4.6. Настройка входа через внешние сервисы идентификации ... 89
4.7. Настройка входа с помощью прокси-аутентификации ... 90
4.8. Настройка входа с помощью сеанса операционной системы ... 91
4.8.1. Настройки контроллера домена (Kerberos-сервера) ... 92
4.8.2. Настройки в консоли управления Blitz Identity Provider ... 94
4.8.3. Настройки браузеров пользователей ... 96
4.8.4. Настройки запуска приложений Blitz Identity Provider ... 98
4.8.5. Настройки веб-сервера ... 98
4.8.6. Отладка проблем с входом по сеансу операционной системы ... 98
4.9. Настройка входа с помощью кодов подтверждения ... 99
4.10. Настройка входа с известного устройства ... 101
4.11. Вход по разовой ссылке ... 101
4.12. Вход по QR-коду ... 102
4.13. Вход с помощью ключей безопасности (WebAuthn, Passkey, FIDO2) ... 103
4.14. Автоматическая идентификация пользователя по свойствам сессии ... 105
4.15. Подтверждение входа разовым паролем на основе состояния (HOTP) ... 108
4.16. Подтверждение входа разовым паролем основе времени (TOTP) ... 110
4.17. Привязка устройств к учетным записям пользователей ... 111
4.17.1. Привязка аппаратных брелоков ... 111
4.17.2. Привязка мобильного приложения ... 113
4.18. Коды подтверждения, отправляемые в SMS и push-уведомлениях ... 114
4.19. Коды подтверждения, отправляемые по электронной почте ... 116
4.20. Подтверждение входа с помощью Duo Mobile ... 117
4.21. Повторное подтверждение при входе с известного устройства ... 120
4.22. Подтверждение с помощью ключей безопасности WebAuthn, Passkey, FIDO2, U2F .. 120
4.23. Подтверждение ответом на контрольный вопрос ... 121
4.24. Подтверждение по входящему звонку ... 122
4.25. Настройка внешнего метода аутентификации ... 125
4.26. Настройка процедуры имперсонификации ... 127
5. РЕГИСТРАЦИЯ ПРИЛОЖЕНИЙ И СЕТЕВЫХ СЛУЖБ ... 129
5.1. Создание учетной записи нового приложения ... 129
5.2. Настройка SAML и WS-Federation ... 134
5.2.1. Подключение по SAML 1.0/1.1/2.0 ... 134
5.2.2. Подключение по WS-Federation ... 135
5.2.3. Настройка SAML-атрибутов ... 137
5.3. Настройка OAuth 2.0 и OpenID Connect 1.0 ... 138
5.3.1. Настройка приложения ... 138
5.3.2. Общие настройки OAuth 2.0 ... 143
5.3.3. Добавление атрибутов в маркер идентификации ... 146
5.3.4. Настройка динамической регистрации клиентов OAuth 2.0 ... 149
5.4. Настройка Simple ... 151
5.5. Настройка клиента REST-сервисов Blitz Identity Provider ... 154
5.6. Доступ к сетевым службам по RADIUS ... 155
5.6.1. Конфигурирование сервера RADIUS ... 155
5.6.2. Настройка приложения ... 160

5.6.3. Настройка на стороне сетевой службы 162

6. КАСТОМИЗАЦИЯ РАБОТЫ BLITZ IDENTITY PROVIDER ПОСРЕДСТВОМ ПРОГРАММИРОВАНИЯ НА JAVA 163

6.1. Создание процедур входа 163

6.2. Примеры процедур входа 165

6.2.1. Принудительная двухфакторная аутентификация в приложение 166
6.2.2. Ограничение перечня доступных методов первого фактора 166
6.2.3. Разрешить вход в приложение только при определенном значении атрибута у пользователя 167
6.2.4. Запрет входа в приложение после истечения срока действия учетной записи 168
6.2.5. Разрешение входа в приложение только из определенных сетей 169
6.2.6. Запрет работы в нескольких одновременных сессиях 170
6.2.7. Сохранение в утверждениях (claims) перечня групп пользователя 170
6.2.8. Отображение пользователю объявления при входе 171
6.2.9. Запрос ввода пользователем атрибута или актуализации телефона и email 172
6.2.10. Запрос ввода пользователем контрольного вопроса 174
6.2.11. Регистрация ключа безопасности (WebAuthn, Passkey, FIDO2) при входе 175
6.2.12. Отображение пользователю списка выбора значений при входе 177
6.2.13. Получение геоданных пользователя 178

6.3. Кастомизация логики операций с хранилищами данных 179

6.3.1. Принцип кастомизации 179
6.3.2. Конфигурация 179
6.3.3. Написание пользовательской процедуры 180

7. НАСТРОЙКА СЕРВИСОВ САМООБСЛУЖИВАНИЯ ПОЛЬЗОВАТЕЛЕЙ 181

7.1. Общие настройки 181

7.2. Регистрация пользователей 183

7.2.1. Форма регистрации 183
7.2.2. Настройки сервиса регистрации 185
7.2.3. Процедура регистрации 186
7.2.4. Изменение текста условий использования 186

7.3. Личный кабинет 187

7.3.1. Отображение атрибутов пользователя 187
7.3.2. Дополнительные параметры 189

7.4. Восстановление доступа 190

8. ВХОД ЧЕРЕЗ ВНЕШНИЕ ПОСТАВЩИКИ ИДЕНТИФИКАЦИИ 194

8.1. Вход через Apple ID 195

8.2. Вход через Google 199

8.3. Вход через Яндекс 201

8.4. Вход через Facebook 203

8.5. Вход через ВКонтакте 205

8.6. Вход через Одноклассники 207

8.7. Вход через Mail ID 209

8.8. Вход через VK ID 211

8.9. Вход через Единую систему идентификации и аутентификации (ЕСИА) 213

8.10. Вход через Цифровой профиль ЕСИА 217

8.11. Вход через Сбер ID 222

8.12. Вход через Tinkoff ID 223

8.13. Вход через систему идентификации ВТБ ID 225

8.14. Вход через систему идентификации СберБизнес ID ... 227
8.15. Вход через систему идентификации Альфа ID ... 229
8.16. Вход через Mos ID (СУДИР) ... 231
8.17. Вход через другую установку Blitz Identity Provider ... 234
8.18. Настройки связывания учетных записей ... 236
9. УПРАВЛЕНИЕ УЧЕТНЫМИ ЗАПИСЯМИ ПОЛЬЗОВАТЕЛЕЙ ... 240
9.1. Поиск учетных записей пользователей ... 240
9.2. Добавление учетной записи пользователя ... 241
9.3. Просмотр и изменение атрибутов пользователя ... 242
9.3.1. Редактирование атрибутов пользователя ... 243
9.3.2. Сброс сессий пользователя ... 243
9.3.3. Смена пароля пользователя ... 244
9.3.4. Просмотр и отвязка привязанных учетных записей внешних поставщиков идентификации ... 244
9.3.5. Привязка устройств для проведения двухфакторной аутентификации по разовому паролю ... 245
9.3.6. Привязка мобильного приложения Duo Mobile ... 246
9.3.7. Просмотр групп, в которые включен пользователь, управление членством пользователя в группах 247
9.3.8. Просмотр прав, назначение и отзыв прав ... 248
9.3.9. Просмотр и удаление запомненных устройств и браузеров ... 250
9.3.10. Управление ключами безопасности ... 250
9.3.11. Просмотр и удаление выданных приложениям разрешений ... 251
10. УПРАВЛЕНИЕ ГРУППАМИ ПОЛЬЗОВАТЕЛЕЙ ... 253
11. УПРАВЛЕНИЕ ПРАВАМИ ДОСТУПА ... 255
12. ПРОСМОТР СОБЫТИЙ БЕЗОПАСНОСТИ ... 256
13. НАСТРОЙКА УВЕДОМЛЕНИЙ И ОТПРАВКИ СООБЩЕНИЙ ... 257
13.1. Настройка подключения к SMS-шлюзу ... 258
13.2. Настройка подключения к сервису отправки push-уведомлений ... 260
13.3. Настройка подключения к SMTP-шлюзу ... 262
14. НАСТРОЙКА ВНЕШНЕГО ВИДА СТРАНИЦЫ ВХОДА ... 264
14.1. Редактирование шаблона по умолчанию ... 264
14.2. Создание и изменение новых шаблонов с помощью конструктора ... 268
14.3. Создание и изменение новых шаблонов в ручном режиме ... 269
15. НАСТРОЙКИ ШЛЮЗА БЕЗОПАСНОСТИ ... 272
15.1. Настройка blitz-keeper ... 273
15.2. Создание правил доступа к сервисам ... 275
15.3. Настройка правил обмена маркеров доступа ... 279
16. НАСТРОЙКИ КОНФИГУРАЦИОННЫХ ФАЙЛОВ ... 280
16.1. Файл настроек blitz.conf ... 280
16.1.1. Ограничение количества одновременных проверок пароля пользователя ... 282
16.1.2. Отключение функции смены пароля при входе ... 282
16.1.3. Настройка внешнего валидатора атрибута ... 282
16.1.4. Настройка транслятора атрибута ... 283
16.1.5. Настройка вызова внешнего сервиса проверки электронной подписи ... 284
16.1.6. Настройка вызова плагина электронной подписи ... 284
16.1.7. Настройка CAPTCHA ... 285

16.1.8. Настройка отправки событий в сервер очередей 289
16.1.9. Настройка использования сервера очередей в качестве брокера сообщений 291
16.1.10. Настройка хранения объектов в Couchbase Server 292
16.1.11. Настройка времени хранения объектов в базе данных 293
16.1.12. Настройка домена Blitz Identity Provider 293
16.1.13. Расширенные настройки подключения к хранилищам 295
16.1.14. Блокирование неактивных пользователей 297
16.1.15. Запрет повторного использования идентификатора удаленного пользователя 298
16.1.16. Настройка групп пользователей 298
16.1.17. Настройка контейнера ключей для работы с ЕСИА и Цифровым профилем 299
16.1.18. Вход через ЕСИА в режиме выбора сотрудника организации 306
16.1.19. Настройка доверенных сертификатов поставщиков ключей безопасности FIDO2 и U2F 309
16.1.20. Настройки сервиса OIDC Discovery 310
16.1.21. Изменение адресов вызовов внешних поставщиков идентификации 310
16.1.22. Настройка внешнего SAML поставщика входа 310
16.1.23. Настройка внешнего поставщика входа СУДИС 312
16.1.24. Включение режима регистрации незавершенных попыток входа 314
16.1.25. Настройка передачи событий безопасности в файл или Kafka 316
16.1.26. Изменение системных имен полей ввода логина и пароля 324
16.1.27. Настройка использования базы геоданных 324
16.1.28. Настройки вспомогательных приложений (pipes) 325
16.1.29. Одновременное использование нескольких СУБД 326
16.1.30. Хранение настроек подключенных приложений в отдельных файлах 327
16.1.31. Настройка работы с контрольным вопросом 328
16.1.32. Включение метода автоматической идентификации 329

16.2. Настройки текстов интерфейса 330
16.2.1. Мультиязычность 330
16.2.2. Модификация текстовых сообщений веб-интерфейса 332
16.2.3. Кастомизация текстов для разных методов автоматической идентификации 332
16.2.4. Модификация шаблонов писем и SMS-сообщений 333
16.2.5. Настройка логотипов кнопок входа через внешние поставщики идентификации 338
16.2.6. Модификация имен устройств и браузеров 339
16.2.7. Модификация сообщений для разных приложений 340
16.2.8. Настройка сообщений вспомогательных приложений (pipes) 341

16.3. Файлы настроек консоли управления 342
16.3.1. Настройка входа в консоль управления через SSO 343
16.3.2. Ограничение сессий 345
16.3.3. Настройка ролей и прав доступа в консоль управления 346

17. МОНИТОРИНГ ФУНКЦИОНИРОВАНИЯ ПРИЛОЖЕНИЙ 347

17.1. Стандартный сервис мониторинга 347

17.2. Использование Grafana и Prometheus 349

18. РЕШЕНИЕ ПРОБЛЕМ 351

ПРИЛОЖЕНИЕ 1. ФУНКЦИОНАЛЬНАЯ СПЕЦИФИКАЦИЯ BLITZ IDENTITY PROVIDER 353

ПРИЛОЖЕНИЕ 2. РЕКОМЕНДАЦИИ ПО ОБЕСПЕЧЕНИЮ МЕР ЗАЩИТЫ ИНФОРМАЦИИ СОГЛАСНО ТРЕБОВАНИЯМ ФСТЭК 358

ПРИЛОЖЕНИЕ 3. МЕТОДИКА ПРОВЕРКИ ОСНОВНЫХ ПАРАМЕТРОВ И ХАРАКТЕРИСТИК ПО 360

Введение

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

Основные[1] функции Blitz Identity Provider:

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

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

Blitz Identity Provider используется как интеграционная платформа для подключения приложений компании к LDAP-каталогам и контроллерам домена. Если компания используется домен, то Blitz Identity Provider обеспечит сквозной доступ сотрудников к приложениям компании таким образом, что сотрудник будет проходить аутентификацию однократно, при входе в сетевой домен.

При работе с сертифицированной версией приемка Blitz Identity Provider осуществляется в соответствии правилами, указанными в документе 1147746651733.62.01.000.001.ТУ «Программное обеспечение «Blitz Identity Provider». Технические условия». Комплектность должна соответствовать комплектности, приведенной в документе 1147746651733.62.01.000.001.ТУ «Программное обеспечение «Blitz Identity Provider». Технические условия».

Безопасная установка и настройка Blitz Identity Provider должна осуществляться в соответствии с настоящим Руководством администратора.

[1] Подробная функциональная спецификация Blitz Identity Provider приведена в Приложении 1.

1. Подготовка к установке

При развертывании Blitz Identity Provider нужно установить и настроить:

1. Веб-сервер. Можно использовать существующий веб-сервер компании для балансировки нагрузки и снятия SSL-шифрования с входящего трафика.
2. Приложения Blitz Identity Provider – сервис аутентификации, приложение регистрации, приложение восстановления доступа, шлюз безопасности, консоль управления. Приложения регистрации, восстановления доступа, шлюз безопасности можно не устанавливать, если связанные с ними функции не планируется использовать.
3. СУБД. Можно использовать Couchbase Server, PostgreSQL[2], Postgres Pro, Jatoba.
4. Хранилище учетных записей и паролей. Можно использовать LDAP-сервер, Microsoft Active Directory или любую (потребуется разработать коннектор) существующую систему хранения учетных записей и паролей.
5. Сервер очередей – используется RabbitMQ. Также можно настроить передачу событий безопасности в Kafka.

Развертывание возможно в конфигурации с минимальными ресурсами либо в кластерной конфигурации.

1.1. Минимальные требования к развертыванию

Рекомендуется применять при подготовке сред тестирования и для продуктивных контуров при внедрениях со средними требованиями к обеспечению доступности и производительности.

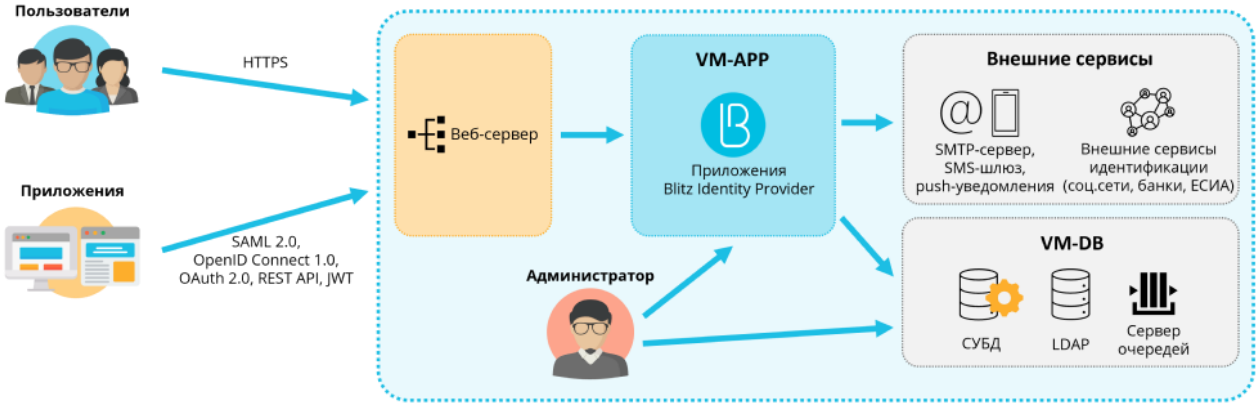

Рисунок 1 – Развертывание в минимальной конфигурации

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

[2] Взаимодействие Blitz Identity Provider с PostgreSQL осуществляется по JDBC. Вместо PostgreSQL можно использовать любую реляционную СУБД с поддержкой JDBC, но это должно быть отдельно согласована с ООО «РЕАК СОФТ» в рамках соответствующих проектов внедрения.

Таблица 1

Минимальные требования к серверам для развертывания

Описание	Поддерживаемые ОС	Технические характеристики	Программное обеспечение
ВМ для приложений (VM-APP)	CentOS 7/8 Rocky Linux 8/9 AlmaLinux 8/9	4 ядра ЦПУ, 8 ГБ ОЗУ, 50 ГБ НЖМД (HDD)	Blitz Identity Provider (blitz-idp, blitz-console, blitz-registration, blitz-recovery, blitz-keeper), JDK, nginx, memcached
ВМ для базы данных (VM-DB)	RHEL 7/8/9 Oracle Linux 8/9 Astra Linux SE 1.6/1.7 РЕД ОС 7.3 Альт Сервер 10 Альт 8 СП Сервер ОСнова 2.5.1	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)

Требуемые версии системного ПО:

- OpenJDK 8, Liberica JDK 8, Axiom JDK 8 Certified или Oracle JDK 8;
- 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):

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

  - к SMS-шлюзу (при его использовании);

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

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

1.2. Рекомендуемые требования к развертыванию в кластере

Схема развертывания в кластерной конфигурации приведена на рисунке 2. Рекомендуется использовать при построении продуктивных контуров систем аутентификации с высокими требованиями к доступности и пиковой производительности.

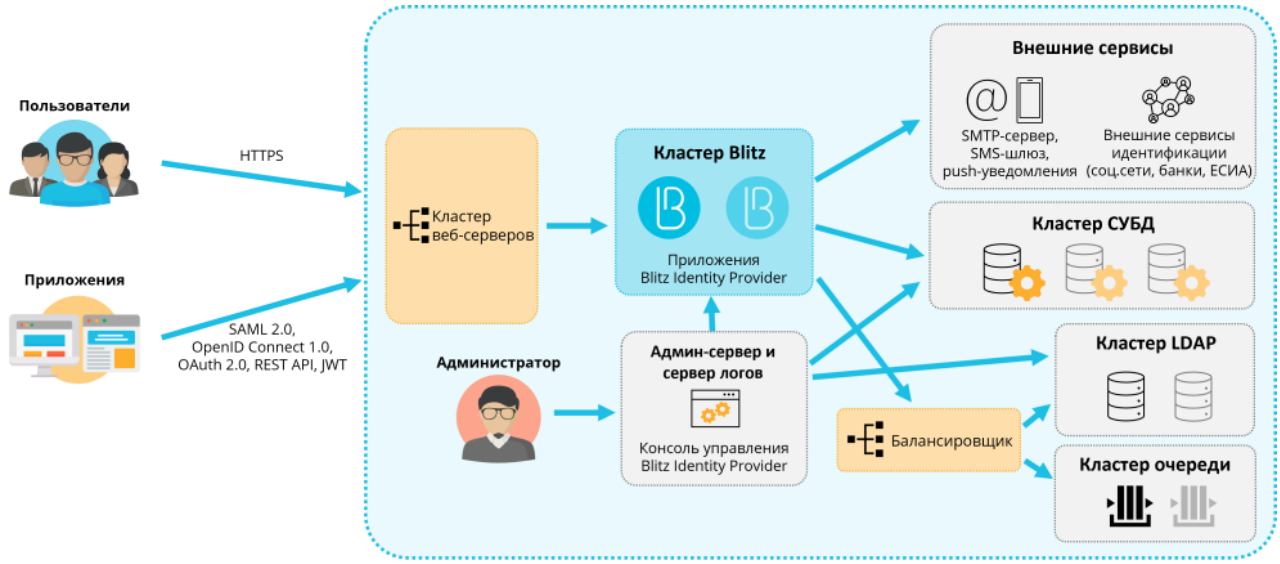

Рисунок 2 – Развертывание в кластерной конфигурации

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

Таблица 2

Рекомендуемые требования к серверам для развертывания в кластере

Описание	Кол-во	Поддерживаемые ОС	Технические характеристики	Программное обеспечение	Комментарий
ВМ веб-серверов (VM-WEB)	1-2	CentOS 7/8 Rocky Linux 8/9 AlmaLinux 8/9 RHEL 7/8/9 Oracle Linux 8/9 Astra Linux SE 1.6/1.7 РЕД ОС 7.3 Альт Сервер 10 Альт 8 СП Сервер ОСнова 2.5.1	4 ядра ЦПУ, 4 ГБ ОЗУ, 50 ГБ НЖМД (HDD)	nginx	Можно использовать существующий веб-сервер для балансировки нагрузки и снятия TLS с входящего трафика
ВМ приложений Blitz Identity Provider (VM-APP)	2		4 ядра ЦПУ, 8 ГБ ОЗУ, 50 ГБ НЖМД (HDD)	Blitz Identity Provider (blitz-idp, blitz-registration, blitz-recovery, blitz-keeper),	При высокой нагрузке рекомендуется развертывать каждое приложение Blitz Identity

Описание	Кол-во	Поддерживаемые ОС	Технические характеристики	Программное обеспечение	Комментарий
				memcached JDK	Provider в своем кластере на отдельных серверах
ВМ для консоли администрирования (VM-ADM)	1		2 ядра ЦПУ, 4 ГБ ОЗУ, 100 ГБ НЖМД (HDD)	Blitz Identity Provider (blitz-console), memcached, JDK	На этот сервер рекомендуется настроить сбор логов с различных серверов кластера Blitz Identity Provider
ВМ для СУБД (VM-DB):	2-3		Для PostgreSQL: 4 ядра ЦПУ, 8 ГБ ОЗУ, 100 ГБ НЖМД (HDD) (данные), 50 ГБ НЖМД (HDD) (система) Для Couchbase Server[3]: 8 ядер ЦПУ, 16 ГБ ОЗУ, 500 ГБ НЖМД (HDD) (данные), 100 ГБ НЖМД (SSD) (индексы), 50 ГБ НЖМД (HDD) (система)	PostgreSQL (9.6 или новее) или Couchbase Server Community Edition (6.0 или новее)	Для PostgreSQL рекомендуется выделить один физический сервер под основной экземпляр и один под резерв (standby). Для Couchbase Server рекомендуется минимум[4] 3 ВМ.
ВМ для LDAP (VM-LDAP)	2		4 ядра ЦПУ, 8 ГБ ОЗУ, 100 ГБ НЖМД (HDD)	389 Directory Server	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)
ВМ для сервера очередей (VM-MQ)	1-2		4 ядра ЦПУ, 8 ГБ ОЗУ, 50 ГБ НЖМД (HDD)	RabbitMQ версии 3.7.9	Использование сервера очередей опционально
ВМ для балансировщика (VM-NLB)	1-2		2 ядра ЦПУ, 4 ГБ ОЗУ, 50 ГБ НЖМД (HDD)	HAProxy, keepalived	Внутренний балансировщик нужен в случае кластеризации LDAP и сервера очередей

Требуемые версии системного ПО:

- OpenJDK 8, Liberica JDK 8, Axiom JDK 8 Certified или Oracle JDK 8;
- 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) и VM-ADM по 9001 (blitz-console);
- с VM-APP должен быть доступ:

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

- 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):

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

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

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

5432 (порт PostgreSQL);

- с VM-LDAP должен быть доступ до других VM-LDAP по 389, 636 (порты LDAP);
- с VM-MQ должен быть доступ до других VM-MQ по 4369, 35197, 5672.

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

2. Установка

Для установки Blitz Identity Provider необходимо:

1. Установить JDK.
2. Установить менеджер памяти memcached.
3. Установить и настроить СУБД.
4. Установить и настроить сервер очередей RabbitMQ (опционально).
5. Установить консоль управления Blitz Console.
6. Установить приложения Blitz Identity Provider.
7. Установить шлюз безопасности Blitz Keeper (опционально).
8. Настроить синхронизацию конфигурационных файлов.
9. Настроить веб-сервер.
10. Настроить внешнее хранилище учетных записей (опционально).

В зависимости от используемой операционной системы есть своя специфика по установке необходимого окружения. Далее приводятся:

1. общая инструкция по установке на операционных системах CentOS 7, RHEL 7 и Astra Linux Special Edition 1.6;
2. специализированные экспресс-инструкции по установке для следующих операционных систем с использованием СУБД PostgreSQL:
   - Astra Linux Special Edition 1.7;
   - Альт 8 СП Сервер;
   - Альт Сервер 10;
   - ОСнова 2.5.1;
   - Red OS 7.3;
   - Rocky Linux 8, AlmaLinux 8, Oracle Linux 8;
   - Rocky Linux 9, AlmaLinux 9, Oracle Linux 9.

2.1. Общая инструкция по установке

2.1.1. Установка JDK

На серверах, предназначенных для установки ПО сервера Blitz Identity Provider и административной консоли Blitz Identity Provider, необходимо установить и настроить JDK 8.

В качестве JDK рекомендуется использовать один из следующих:

- OpenJDK 8;
- Liberica JDK 8;
- Axiom JDK 8 Certified;

- Oracle JDK 8.

При использовании сертифицированной версии Blitz Identity Provider рекомендуется использовать один из следующих JDK:

- OpenJDK 8;
- Axiom JDK 8 Certified.

Инструкция по установке OpenJDK 8 в CentOS и RHEL:

- Выполнить команду:

```
sudo yum install java-1.8.0-openjdk-devel
```

Инструкция по установке Liberica JDK 8 в Astra Linux Special Edition 1.6:

- загрузить дистрибутив Liberica JDK 8 с сайта производителя;
- выполнить команду:

```
pkg -i bellsoft-jdk8u252+9-linux-amd64.deb
```

- открыть на редактирование файл java.security в директории /usr/lib/jvm/bellsoft-java8-amd64/jre/lib/security;
- раскомментировать (или добавить) строку:

```
crypto.policy=unlimited
```

Инструкция по установке Axiom JDK 8 Certified в Astra Linux Special Edition 1.7:

- загрузить дистрибутив Axiom JDK 8 Certified с сайта производителя;
- выполнить команду:

```
dpkg -i bellsoft-jdk-certified8u322+7-linux-amd64-astra.deb
```

- открыть на редактирование файл java.security в директории /usr/lib/jvm/bellsoft-java8-amd64/jre/lib/security;
- раскомментировать (или добавить) строку:

```
crypto.policy=unlimited
```

Инструкция по установке и настройке Oracle JDK 8:

- загрузить дистрибутив Oracle JDK 8 в виде архива tar[5];
- скопировать загруженный дистрибутив на сервера (например, в директорию /tmp);
- создать директорию под установку Oracle JDK 8:

```
mkdir -p /opt/oracle/jdk/
```

- распаковать в созданную директорию дистрибутив Oracle JDK 1.8:

```
tar xf /tmp/jdk-8uXXX-linux-x64.tar.gz -C /opt/oracle/jdk/
```

Если версия Oracle JDK 1.8.0_151 и выше:

- открыть на редактирование файл java.security в директории /opt/oracle/jdk/jdk1.8.0_XXX/jre/lib/security;
- раскомментировать (или добавить) строку:

```
crypto.policy=unlimited
```

[5] См.: https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html

Если версия Oracle JDK 1.8.0_144 и ниже:

- загрузить дистрибутив Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 8[6];
- скопировать загруженный дистрибутив на сервера (например, в директорию /tmp);
- распаковать архив и скопировать содержимое в директорию с установленным Oracle JDK 8:

```
cd /tmp
unzip jce_policy-8.zip
cp UnlimitedJCEPolicyJDK8/*.jar /opt/oracle/jdk/jdk1.8.0_XXX/jre/lib/security/
```

2.1.2. Установка memcached

Версия memcached должна быть 1.4.15 или выше. Сервис memcached должен быть установлен на серверах, предназначенных для установки приложений Blitz Identity Provider: blitz-console, blitz-idp, blitz-registration, blitz-recovery. Для приложения blitz-keeper сервис memcached не нужен.

Для установки memcached в CentOS и RHEL:

- выполнить команду:

```
yum install memcached
```

- после завершения установки добавить сервис memcached в автозапуск и запустить сервис:

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

Для установки memcached в Astra Linux Special Edition 1.6:

- выполнить команду:

```
apt-get install memcached
```

- после завершения установки добавить сервис memcached в автозапуск и запустить сервис:

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

Сервис memcached запускается на порту 11211. Нужно убедиться, что этот порт открыт на межсетевых экранах и может быть использован для соединения между серверами с приложениями Blitz Identity Provider.

2.1.3. Установка и настройка СУБД

Сервер аутентификации Blitz Identity Provider поддерживает для работы использование следующих СУБД:

- СУБД 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

[6] См.: https://www.oracle.com/java/technologies/javase-jce8-downloads.html

отечественных операционных систем.

- СУБД 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.

2.1.3.1. Установка и настройка PostgreSQL

Версия PostgreSQL должна быть 9.6 или новее.

В случае CentOS и RHEL необходимо установить PostgreSQL согласно инструкции: https://www.postgresql.org/download/linux/redhat/.

В случае Astra Linux Special Edition 1.6 для установки PostgreSQL необходимо:

- выполнить команду:

```
apt-get install postgresql
```

- 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;
- далее перейти в директорию и по очереди выполнить команды:

```
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, для релиза 5.21 это скрипты:

```
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
```

- настроить резервное копирование БД, используя инструкцию[7].

2.1.3.2. Установка и настройка 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.

Необходимо установить Couchbase Server на каждый из выделенных под установку СУБД серверов согласно инструкции:

https://docs.couchbase.com/server/current/install/install-linux.html

Дистрибутив Couchbase Server можно загрузить здесь:

https://www.couchbase.com/downloads

Примечание: 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 и Couchbase Server.

После завершения установки добавить сервис Couchbase Server в автозапуск и запустить сервис:

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

Проверить работоспособность сервиса, выполнив команду:

```
systemctl status couchbase-server
```

Далее необходимо:

- инициализировать на каждом сервере кластер Couchbase Server согласно инструкции[8] (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;
- на одном любом сервере кластера Couchbase Server выполнить скрипт по подготовке Couchbase Server к использованию Blitz Identity Provider:
  - скрипт находятся в директории couchbase в архиве resources.zip в составе

[7] См.: https://postgrespro.ru/docs/postgresql/9.6/backup-dump#backup-dump-all
[8] См.: https://docs.couchbase.com/server/current/manage/manage-nodes/initialize-node.html

дистрибутива Blitz Identity Provider;

- скрипт нужно скопировать на любой сервер кластера Couchbase Server;
- далее перейти в директорию и выполнить скрипт создания buckets для хранения информации Blitz Identity Provider и индексов для выполнения поисковых запросов Blitz Identity Provider в БД:

```
./cb_init.sh
```

- 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 приложений Blitz Identity Provider (рекомендуется задать имя blitz);
  - пароль учетной записи Couchbase Server для подключения приложений Blitz Identity Provider.
- после выполнения скрипта произвести следующие настройки:
  - 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 3 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).
  - настроить резервное копирование БД, используя инструкцию[9].

[9] См.: https://docs.couchbase.com/server/current/manage/manage-backup-and-restore/manage-backup-and-restore.html

2.1.4. Установка и настройка сервера очередей RabbitMQ

Установка сервера очередей RabbitMQ опциональна и требуется если сервер очередей будет использоваться для передачи событий в смежные системы (см. п. 16.1.8) или в качестве брокера сообщений (см. п. 16.1.9).

В случае CentOS и RHEL необходимо установить RabbitMQ согласно инструкции: https://www.rabbitmq.com/install-rpm.html.

В случае Astra Linux Special Edition 1.6 для установки RabbitMQ необходимо:

- выполнить команду:

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

- после завершения установки запустить сервис:

```
systemctl start rabbitmq-server
```

2.1.5. Установка приложений Blitz Identity Provider

Blitz Identity Provider состоит из следующих приложений:

- blitz-console – консоль управления;
- blitz-idp – сервис аутентификации и веб-приложение «личный кабинет»;
- blitz-registration – сервис регистрации;
- blitz-recovery – сервис восстановления пароля;
- blitz-keeper – шлюз безопасности.

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

Для установки приложения blitz-keeper используется свой установщик blitz-keeper-5.X.X.bin.

При установке сертифицированной версии Blitz Identity Provider дополнительно используются файлы blitz-idp-thirdparty-5.X.X.tar.gz и blitz-keeper-thirdparty-5.X.X.tar.gz, содержащие архивы с используемыми Blitz Identity Provider сторонними библиотеками.

Установку консоли управления можно провести 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 (см. п. 2.1.1) и memcached (см. п. 2.1.2).

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

- на предназначенные для установки сервера скопировать (например, в директорию /tmp) из дистрибутива Blitz Identity Provider файл blitz-5.X.X.bin и blitz-idp-thirdparty-5.X.X.tar.gz (только в случае установки сертифицированной версии);
- 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 – задать директорию, в которую на сервере установлен JDK (например, /usr/lib/jvm/bellsoft-java8-amd64 для Liberica JDK и Axiom JDK Certified, /usr/lib/jvm/java-1.8.0-openjdk для OpenJDK 8, /opt/oracle/jdk для Oracle JDK 8);

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

Пример запуска установщика:

```
cd /tmp
chmod +x blitz-5.X.X.bin
./blitz-5.X.X.bin -- -j /usr/lib/jvm/bellsoft-java8.x86_64 -i "idp console recovery registration"
```

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
************************************************************
```

- создать файл blitz_param.txt, к котором задать первичные настройки Blitz Identity Provider:
  - DOMAIN – внешнее имя домена, на котором будет функционировать Blitz Identity Provider;
  - ROOT_CONTEXT – URL-путь, на котором будет функционировать Blitz Identity Provider (если параметр не указывать, то по умолчанию будет задан /blitz);
  - ADMIN_USERNAME – имя учетной записи администратора в Blitz Identity Provider (если параметр не указывать, то по умолчанию будет задан admin);
  - ADMIN_PASSWORD – пароль от учетной записи администратора в Blitz Identity Provider (если параметр не указывать, то пароль будет автоматически сгенерирован и выведен в результатах работы скрипта конфигурации);
  - KEYSTORE_PASSWORD – пароль от создаваемого в процессе установки ключевого контейнера (если параметр не указывать, то пароль будет автоматически сгенерирован и выведен в результатах работы скрипта конфигурации);
  - MEMCACHED_SERVERS – адреса серверов с memcached;
  - DB_MODE – используемая СУБД: PG для PostgreSQL (Jatoba), CB для Couchbase Server;
  - PG_HOSTNAME – адрес СУБД PostgreSQL;
  - PG_DB_NAME – имя БД в СУБД PostgreSQL (рекомендуется использовать blitzdb);

- PG_USERNAME – имя учетной записи в СУБД PostgreSQL (рекомендуется использовать blitz);
- PG_PASSWORD – пароль от учетной записи в СУБД PostgreSQL;
- 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 Blitz Identity Provider (по умолчанию 127.0.0.1/32).

Пример конфигурационного файла для работы с PostgreSQL:

```
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
```

Пример конфигурационного файла для работы с Couchbase Server:

```
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
```

- 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 Blitz Identity Provider configured on domain: test.loc
Your Blitz Identity Provider Console available on addresses:
  http://test.loc: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
```

```
************************************************************
```

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-*
```

- если планируется использовать функцию защиты REST-сервисов с помощью Blitz Identity Provider, то скопировать на предназначенные для установки шлюза безопасности сервера (например, в директорию /tmp) из дистрибутива Blitz Identity Provider файлы blitz-keeper-5.X.X.bin и blitz-keeper-thirdparty-5.X.X.tar.gz (только в случае установки сертифицированной версии);
- запустить установщик blitz-keeper-5.X.X.bin:

```
cd /tmp
chmod +x blitz-keeper-5.X.X.bin
./blitz-keeper-5.X.X.bin
```

- в ответ на запросы установщика задать:
  - значение JAVA_HOME – задать директорию, в которую на сервере установлен JDK (например, /usr/lib/jvm/bellsoft-java8-amd64 для Liberica JDK и Axiom JDK Certified, /usr/lib/jvm/java-1.8.0-openjdk для OpenJDK 8, /opt/oracle/jdk для Oracle JDK 8);
  - путь к файлу blitz-keeper-thirdparty-5.X.X.tar.gz (только в случае установки сертифицированной версии).
- дождаться окончания установки приложения. Установка будет произведена в директорию /usr/share/identityblitz.
- добавить приложения 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
systemctl enable blitz-keeper
systemctl start blitz-keeper
```

2.1.6. 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:

- 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 (см. п. 16.1.14);
- blitzDispatchedQueues – задает имя очереди, из которой приложение Blitz Identity Provider обрабатывает задачи на отправку писем, регистрацию пользователей и восстановление паролей. По умолчанию используется очередь с именем default;
- blitz.stores.united.u-cache.ttlInSec – срок действия кэша данных учетной записи, предоставляемых через REST API. По умолчанию 1 секунда;
- blitz.csrf.cookie.ttlInSec – задает время действия cookie, препятствующей CSRF. По умолчанию соответствует 6 часам (значение задается в секундах). Это максимальное время с момента открытия пользователем страницы и до выполнения заполненной страницы пользователем на сервер;
- blitz.jdbc.cols.types.strings – задает тип колонки, используемой для сохранения строковых атрибутов в реляционной СУБД (PostgreSQL). По умолчанию используется тип text;
- blitz.jdbc.pool.stat-period – задает периодичность, с которой статистика использования JDBC записывается в лог. По умолчанию 300 секунд;
- saml.numThreads – задает количество потоков, которые в Blitz Identity Provider обрабатывают запросы на вход через SAML. По умолчанию 32 потока;
- blitz.oauth.exchange.rules.fs.cache.capacity – задает размер кэша, используемый Blitz Identity Provider для проверки правил доступа к микросервисам. По умолчанию размер кэша в 10000 проверок;
- blitz.oauth.dyn.reg.clientSecretLength – задает размер client_secret, генерируемого при

динамической регистрации пары client_id и client_secret. По умолчанию генерируется client_secret размером в 15 символов.

- blitz.oauth.dyn.reg.clientAttachingTllInSec – задает время, в течение которого сгенерированная при динамической регистрации пара client_id и client_secret должна быть ассоциирована с пользователем (если в течение этого времени пара не будет ассоциирована с пользователем, то она будет аннулирована). По умолчанию соответствует 1 часу (значение задается в секундах).
- blitz.webauthn.residentKey.preferred – если опция задана, то ключи безопасности регистрируются с параметром residentKey=preferred. При этом, в случае если опция задана как true, то requireResidentKey=true, а если опция false, то requireResidentKey=false.
- blitz.ldap.store.extension.class – при передаче в опцию значения com.identityblitz.idp.store.ldap.custom.PasswordMigrationExt включается режим миграции пароля.
- blitz.ldap.store.extension.PasswordMigrationExt.passwordHashAttr – задает имя LDAP-атрибута, в котором храниться хеш-пароля для опции миграции пароля. Хэш должен содержать префикс {bcrypt} для миграции паролей из хэшей с алгоритмом bcrypt.
- extensionsDir – адрес директории с модулями расширений для Blitz Identity Provider (см. п. 16.1.3).
- 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[10]:
  - 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;

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

  - 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.

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

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"
```

2.1.7. Настройка синхронизации файлов конфигурации

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

1. На сервере с консолью управления Blitz Console:
   - установить rsync и incron:

```
sudo yum install rsync incron
```

     или (для Astra Linux Special Edition 1.6)

```
sudo apt install rsync incron
```

   - переключиться на пользователя blitz

```
sudo su - blitz
```

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

```
ssh-keygen
```

- прочитать и сохранить для дальнейшего использования публичный ssh ключ:

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

- 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 $# $%
/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 $# $%
```

- 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
```

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"
```

- сделать файл /usr/share/identityblitz/scripts/config_sync.sh исполняемым:

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

- запустить incrontab, выполнив под пользователем root команду:

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

2. На остальных серверах приложений Blitz Identity Provider:

- установить rsync:

```
sudo yum install rsync
```

или (для Astra Linux Special Edition 1.6)

```
sudo apt install rsync
```

- переключиться в пользователя blitz:

```
sudo su - blitz
```

- выполнить следующий скрипт:

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

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

2.1.8. Настройка веб-сервера

В качестве веб-сервера рекомендуется использовать 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-keeper {
      server [BLITZ-KPR-NODE-01]:9012 max_fails=3 fail_timeout=120;
      server [BLITZ-KPR-NODE-02]:9012 max_fails=3 fail_timeout=120;
}
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 приложениями Blitz Identity Provider (blitz-idp, blitz-registration, blitz-recovery, blitz-keeper);

– [BLITZ-CONSOLE-NODE-01] – имя (hostname) сервера с Blitz Console.

2. Скорректировать блок настроек снятия 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 $remote_addr;
```

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.

2.1.9. Установка и настройка LDAP-каталога

В качестве хранилища учетных записей 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.

Для установки 389 Directory Server в CentOS и RHEL:

- выполнить команды установки:

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

- 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
```

- инициализировать LDAP-каталог. Ответить на вопросы установщика.

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

- после завершения установки добавить LDAP-каталог в автозапуск и запустить сервис:

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

Для установки в Astra Linux Special Edition 1.6:

- выполнить команду установки и скрипт инициализации каталога:

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

- 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:

- Скопировать на LDAP-сервер конфигурационные скрипты LDAP из состава дистрибутива Blitz Identity Provider (это папка ldap в архиве resources.zip).
- 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
```

- 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
```

- После выполнения скрипта перезапустить LDAP-каталог:

```
systemctl restart dirsrv.target
```

- 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
```

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

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

  - 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;
  - перезапустить LDAP-каталог, чтобы применить изменения схемы каталога:

```
systemctl restart dirsrv.target
```

2.1.10. 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 (см. п. 2.1.5).

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 3:

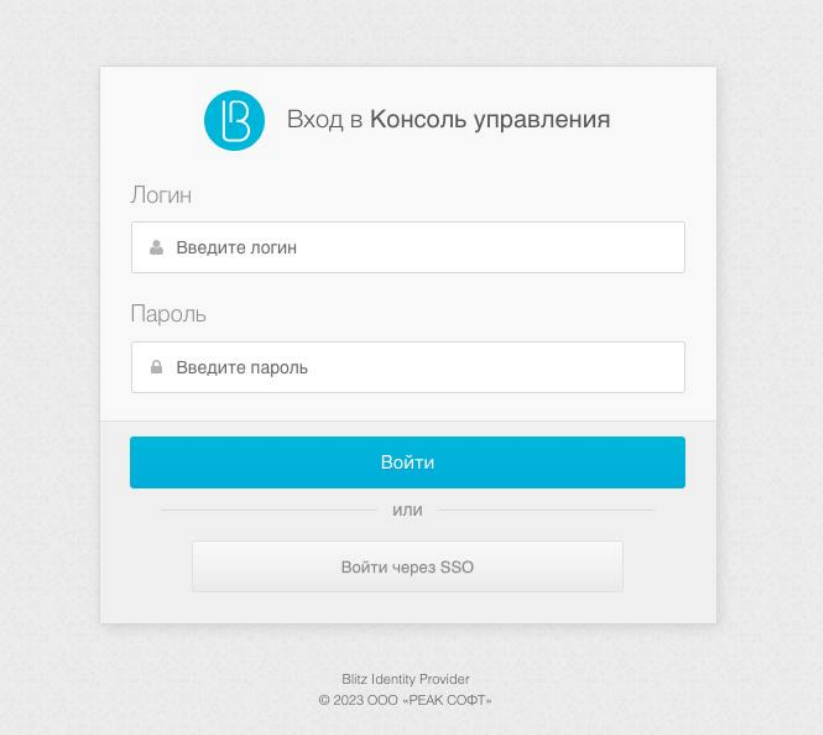

Рисунок 3 – Стандартный вид экрана входа в консоль управления

После успешного входа откроется главная страница консоли управления, вид которой приведен на рисунке 4. Навигация между различными настройками Blitz Identity Provider осуществляется с помощью меню, расположенного в левой части экрана.

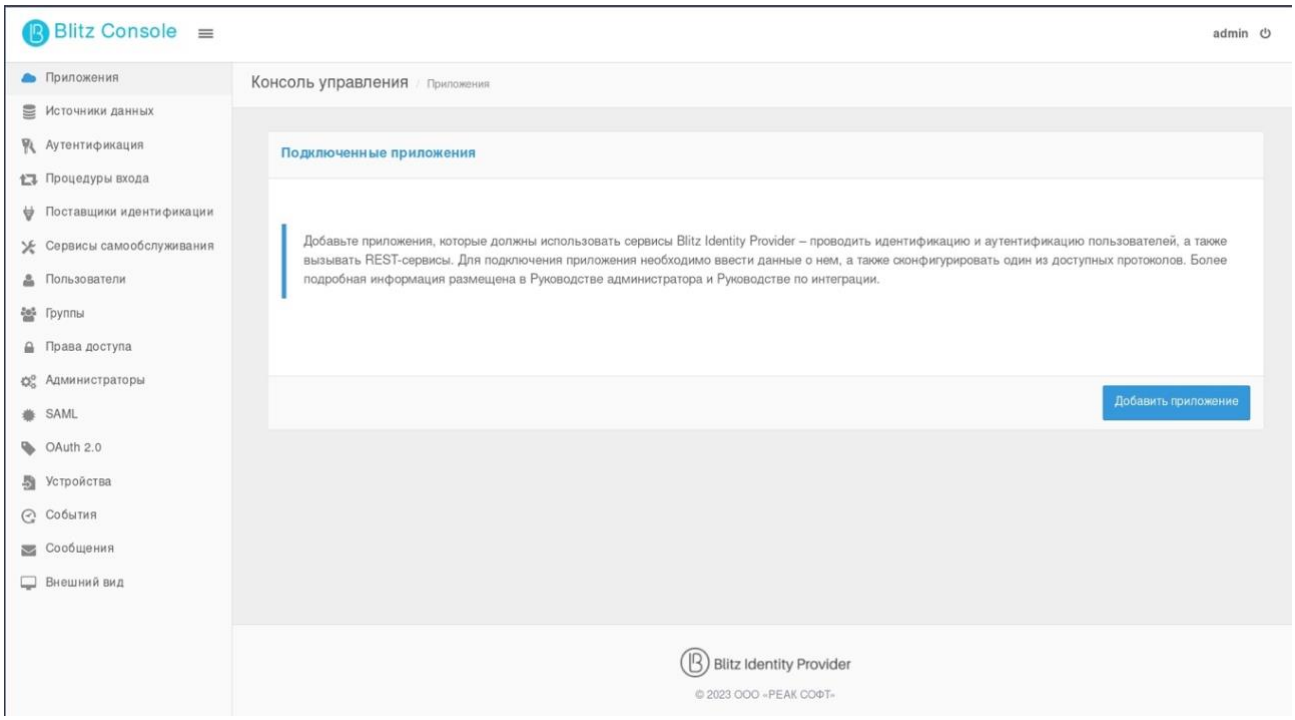

Рисунок 4 – Вид главного экрана консоли управления

2.1.11. Установка лицензионного ключа

Если нажать на ссылке «Вы используете Blitz Identity Provider …, версия …» в футере любой страницы консоли управления Blitz Identity Provider, то будет отображен экран, приведенный на рисунке 5.

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.

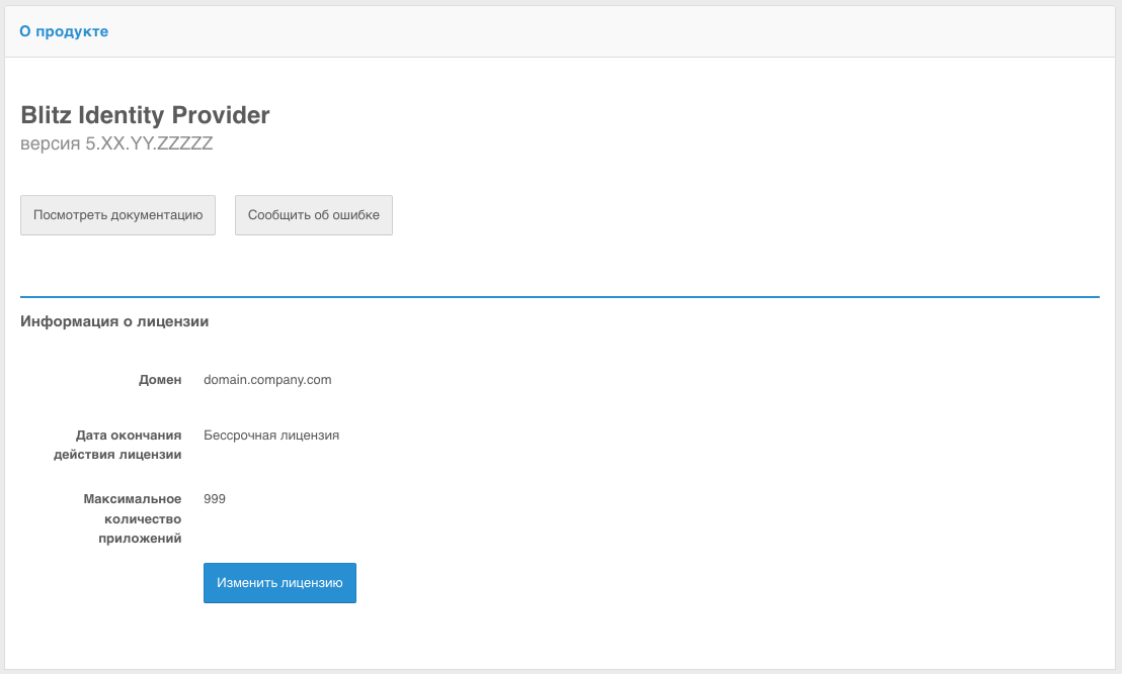

Рисунок 5 – Просмотр информации о лицензии

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

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

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

2.1.12. 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» (Рисунок 6).

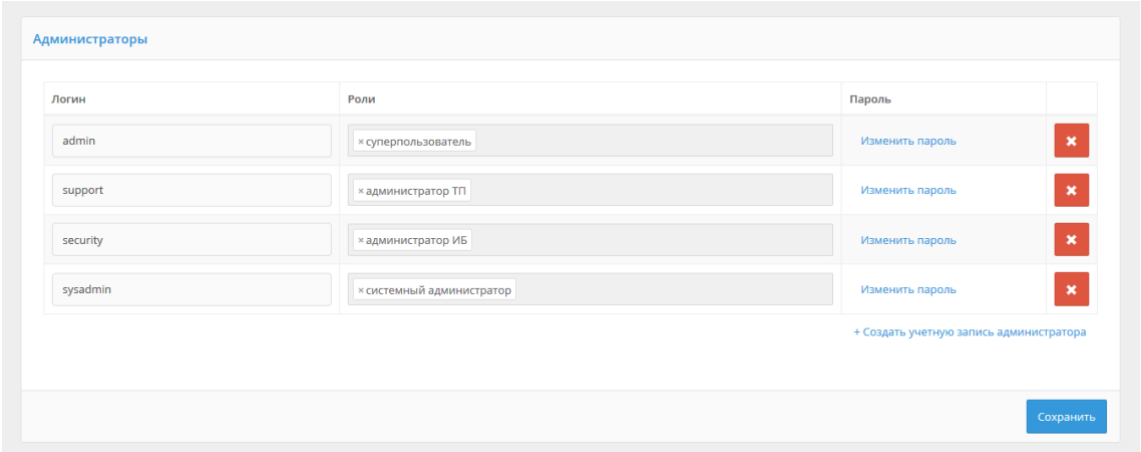

Рисунок 6 – Управление администраторами

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 3. Можно перенастроить существующие роли или создать новые через настройки конфигурационного файла credentials (см. п. 16.3.3).

Таблица 3

Стандартные роли администраторов в Blitz Identity Provider

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

Дополнительно к стандартной идентификации и аутентификации администраторов по логину и паролю при входе в консоль управления можно настроить использование идентификации и аутентификации пользователей в консоль управления с использованием сервера аутентификации Blitz Identity Provider. Настройки выполняются через конфигурационный файл console.conf (см. п. 16.3.1).

2.1.13. Перезапуск приложений Blitz Identity Provider

Для перезапуска приложений 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
```

2.1.14. Рекомендуемые действия после первого запуска Blitz Identity Provider

При первом запуске Blitz Identity Provider зашифровывает созданные при установке пароли администратора и пароли подключения к СУБД. При этом первоначальный конфигурационный файл копируется в каталог /usr/share/identityblitz/blitz-config/.snapshot.

Рекомендуется удалить использованные при установке файлы blitz_param.txt и первые созданные копии конфигурационного файла blitz.conf. Для этого можно выполнить команду:

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

2.2. Экспресс-инструкции по установке

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

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 выполнением работ необходимо обновить операционную систему до актуальных патчей.

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.reaxoft.ru/nextcloud/index.php/s/3W48EBrNXf3R3WC/download?path=%2F'$BLITZ_REL'&files=bli
tz-'$BLITZ_REL.bin -O ~/tmp/blitz/blitz-$BLITZ_REL.bin
wget -q
'https://nc.reaxoft.ru/nextcloud/index.php/s/3W48EBrNXf3R3WC/download?path=%2F'$BLITZ REL'&files=res
ources.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
```

2.2.1. Astra Linux Special Edition 1.7

2.2.1.1. Установка JDK

Установить дистрибутив Liberica JDK:

```
apt install ./bellsoft-jdk8u292+10-linux-amd64.deb
```

2.2.1.2. Установка memcached

Установить дистрибутив:

```
apt install memcached
```

Запустить службу:

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

2.2.1.3. Установка PostgreSQL

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
```

2.2.1.4. Установка RabbitMQ

Установка сервера очередей RabbitMQ опциональна и требуется если сервер очередей будет использоваться для передачи событий в смежные системы (см. п. 16.1.8) или в качестве брокера сообщений (см. п. 16.1.9).

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"
```

2.2.1.5. Установка 389 Directory Server

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
```

2.2.1.6. Установка nginx

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
```

2.2.1.7. Установка Blitz Identity Provider

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

```
/tmp/blitz/blitz-5.X.X.bin -- -j /usr/lib/jvm/bellsoft-java8.x86_64 -i "idp console recovery
registration"
```

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

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

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

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

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

```
************************************************************
Your Blitz Identity Provider configured on domain: test.loc
Your Blitz Identity Provider 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.

2.2.2. Альт 8 СП Сервер

2.2.2.1. Установка JDK

Установить дистрибутив JDK:

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

2.2.2.2. Установка memcached

Установить дистрибутив:

```
apt-get install memcached
```

Запустить службу:

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

2.2.2.3. Установка PostgreSQL

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
```

2.2.2.4. Установка RabbitMQ

Установка сервера очередей RabbitMQ опциональна и требуется если сервер очередей будет использоваться для передачи событий в смежные системы (см. п. 16.1.8) или в качестве брокера сообщений (см. п. 16.1.9).

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"
```

2.2.2.5. Установка 389 Directory Server

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
```

2.2.2.6. Установка nginx

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
```

2.2.2.7. Установка Blitz Identity Provider

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

```
/tmp/blitz/blitz-5.X.X.bin -- -j /usr/lib/jvm/java-1.8.0-openjdk -i "idp console recovery
registration"
```

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

```
DOMAIN=testinstallation.local
MEMCACHED SERVERS="127.0.0.1"
DB MODE=PG
PG HOSTNAME=127.0.0.1
PG_DB_NAME=blitzdb
PG_USER_NAME=blitz
PG_USER_PASSWORD=12ABcd45
```

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

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

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

```
****************************************************************
Your Blitz Identity Provider configured on domain: test.loc
Your Blitz Identity Provider 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, запустить nginx:

```
systemctl start nginx
```

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

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

Запустить службы:

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

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

2.2.3. Альт Сервер 10

2.2.3.1. Установка JDK

Установить дистрибутив JDK:

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

2.2.3.2. Установка memcached

Установить дистрибутив:

```
apt-get install memcached
```

Запустить службу:

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

2.2.3.3. Установка PostgreSQL

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
```

2.2.3.4. Установка RabbitMQ

Установка сервера очередей RabbitMQ опциональна и требуется если сервер очередей будет использоваться для передачи событий в смежные системы (см. п. 16.1.8) или в качестве брокера сообщений (см. п. 16.1.9).

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"
```

2.2.3.5. Установка 389 Directory Server

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
```

2.2.3.6. Установка nginx

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
```

2.2.3.7. Установка Blitz Identity Provider

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

```
/tmp/blitz/blitz-5.X.X.bin -- -j /usr/lib/jvm/java-1.8.0-openjdk -i "idp console recovery
registration"
```

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

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

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

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

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

```
************************************************************
Your Blitz Identity Provider configured on domain: test.loc
Your Blitz Identity Provider 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.

2.2.4. ОСнова 2.5.1

2.2.4.1. Установка JDK

Установить дистрибутив Liberica JDK:

```
apt install ./bellsoft-jdk8u292+10-linux-amd64.deb
```

2.2.4.2. Установка memcached

Установить дистрибутив:

```
apt install memcached
```

Запустить службу:

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

2.2.4.3. Установка PostgreSQL

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
```

2.2.4.4. Установка RabbitMQ

Установка сервера очередей RabbitMQ опциональна и требуется если сервер очередей будет использоваться для передачи событий в смежные системы (см. п. 16.1.8) или в качестве брокера сообщений (см. п. 16.1.9).

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"
```

2.2.4.5. Установка 389 Directory Server

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
```

2.2.4.6. Установка nginx

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
```

2.2.4.7. Установка Blitz Identity Provider

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

```
/tmp/blitz/blitz-5.X.X.bin -- -j /usr/lib/jvm/bellsoft-java8.x86_64 -i "idp console recovery
registration"
```

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

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

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

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

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

```
****************************************************************
Your Blitz Identity Provider configured on domain: test.loc
Your Blitz Identity Provider 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 start nginx
```

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

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

Запустить службы:

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

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

2.2.5. Red OS 7.3

2.2.5.1. Установка JDK

Установить дистрибутив:

```
dnf install java-1.8.0-openjdk-devel
```

2.2.5.2. Установка memcached

Установить дистрибутив:

```
dnf install memcached
```

Запустить службу:

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

2.2.5.3. Установка PostgreSQL

Установить дистрибутив:

```
dnf install postgresql14-server
```

Инициализировать СУБД командой:

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

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

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

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

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

Запустить службу:

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

Подключиться к СУБД и провести первичную настройку

```
su - postgres
psql

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

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

```
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/000-service-tasks.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/001-init-database.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/002-new_pp_columns.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/003-usd_id_table.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/004-usr_auth_table.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/005-usr_agt_table.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/006-usr htp hmc alg.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/007-usr atr cfm.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/008-wak.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/009-fix_pp_column.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/010-add_usr_prp.sql
```

```
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/011-pp_audit.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/012-geo to audit.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/013-tasks.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/014-sec_ch_ua.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/015-5.12.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/016-5.13.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/017-5.15.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/018-5.17.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/019-5.18.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/020-5.20.0.sql
psql -U blitz -h 127.0.0.1 blitzdb -f ~/tmp/blitz/postgres/ddl/021-5.21.0.sql
```

2.2.5.4. Установка RabbitMQ

Установка сервера очередей RabbitMQ опциональна и требуется если сервер очередей будет использоваться для передачи событий в смежные системы (см. п. 16.1.8) или в качестве брокера сообщений (см. п. 16.1.9).

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"
```

2.2.5.5. Установка 389 Directory Server

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
```

2.2.5.6. Установка nginx

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
```

2.2.5.7. Установка Blitz Identity Provider

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

```
/tmp/blitz/blitz-5.X.X.bin -- -j /usr/lib/jvm/java-1.8.0-openjdk -i "idp console recovery
registration"
```

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

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

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

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

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

```
************************************************************
Your Blitz Identity Provider configured on domain: test.loc
Your Blitz Identity Provider 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 start nginx
```

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

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

Запустить службы:

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

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

2.2.6. Rocky Linux 8, AlmaLinux 8, Oracle Linux 8, RHEL 8

2.2.6.1. Установка JDK

Установить дистрибутив:

```
dnf install java-1.8.0-openjdk-devel
```

2.2.6.2. Установка memcached

Установить дистрибутив:

```
dnf install memcached
```

Запустить службу:

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

2.2.6.3. Установка PostgreSQL

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
```

2.2.6.4. Установка RabbitMQ

Установка сервера очередей RabbitMQ опциональна и требуется если сервер очередей будет использоваться для передачи событий в смежные системы (см. п. 16.1.8) или в качестве брокера сообщений (см. п. 16.1.9).

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/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
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"
```

2.2.6.5. Установка 389 Directory Server

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
```

2.2.6.6. Установка nginx

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
```

2.2.6.7. Установка Blitz Identity Provider

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

```
/tmp/blitz/blitz-5.X.X.bin -- -j /usr/lib/jvm/java-1.8.0-openjdk -i "idp console recovery
registration"
```

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

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

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

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

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

```
****************************************************************
Your Blitz Identity Provider configured on domain: test.loc
Your Blitz Identity Provider 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 start nginx
```

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

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

Запустить службы:

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

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

2.2.7. Rocky Linux 9, AlmaLinux 9, Oracle Linux 9, RHEL 9

2.2.7.1. Установка JDK

Установить дистрибутив:

```
dnf install java-1.8.0-openjdk-devel
```

2.2.7.2. Установка memcached

Установить дистрибутив:

```
dnf install memcached
```

Запустить службу:

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

2.2.7.3. Установка PostgreSQL

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
```

2.2.7.4. Установка RabbitMQ

Установка сервера очередей RabbitMQ опциональна и требуется если сервер очередей будет использоваться для передачи событий в смежные системы (см. п. 16.1.8) или в качестве брокера сообщений (см. п. 16.1.9).

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"
```

2.2.7.5. Установка 389 Directory Server

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
```

2.2.7.6. Установка nginx

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
```

2.2.7.7. Установка Blitz Identity Provider

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

```
/tmp/blitz/blitz-5.X.X.bin -- -j /usr/lib/jvm/java-1.8.0-openjdk -i "idp console recovery
registration"
```

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

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

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

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

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

```
************************************************************
Your Blitz Identity Provider configured on domain: test.loc
Your Blitz Identity Provider 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 start nginx
```

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

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

Запустить службы:

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

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

3. Настройка атрибутов учетных записей

3.1. Конфигурирование доступных атрибутов

Учетная запись пользователя описывается набором атрибутов. Значения атрибутов формируются следующими способами:

- считываются из подключенных хранилищ атрибутов (см. подробнее в п. 3.2.2–3.2.4);
- считываются из базы данных Blitz Identity Provider – 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. Например, 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.

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

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

3.1.1. Настройка хранимых атрибутов

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 этого хранилища (см. 3.2.2);
- указать тип значения атрибута – формат данных (String, Number, Boolean, Bytes, Array of Strings);
- определить параметры атрибута:
  - возможно ли производить по нему поиск (колонка «Поиск»)[11];
  - является ли атрибут обязательным (колонка «Обяз.»);
  - должно ли значение атрибута быть уникальным в системе (колонка «Уник.»).

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

В разделе «Пользователи» в карточке пользователя (см. п. 9.3) 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 в messages (см. п. 16.2.2) строки с описанием названий атрибутов для используемых языков. Строки должны иметь вид custom.user.attr.name.<имя атрибута>.

Пример строчек:

```
custom.user.attr.name.sub=Идентификатор
custom.user.attr.name.family_name=Фамилия
custom.user.attr.name.given_name=Имя
custom.user.attr.name.middle_name=Отчество
custom.user.attr.name.email=E-mail
custom.user.attr.name.phone_number=Телефон
```

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.

Если в качестве СУБД используется PostgreSQL, то необходимо создать колонку в таблице USR_ATR, а также в таблице USR (только в случае, если используется внутреннее

[11] Если это атрибут из подключенного хранилища, то в целях производительности рекомендуется создать по нему поисковый индекс.

хранилище, см. п. 3.2.1). Имя колонки должно соответствовать имени добавляемого атрибута с нормализацией из lowerCamelCase в UPPERCASE_SEPARATED_BY_UNDERSCORE, например, middleName -> MIDDLE_NAME. Тип колонки должен быть выбран в зависимости от типа значения атрибута:

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

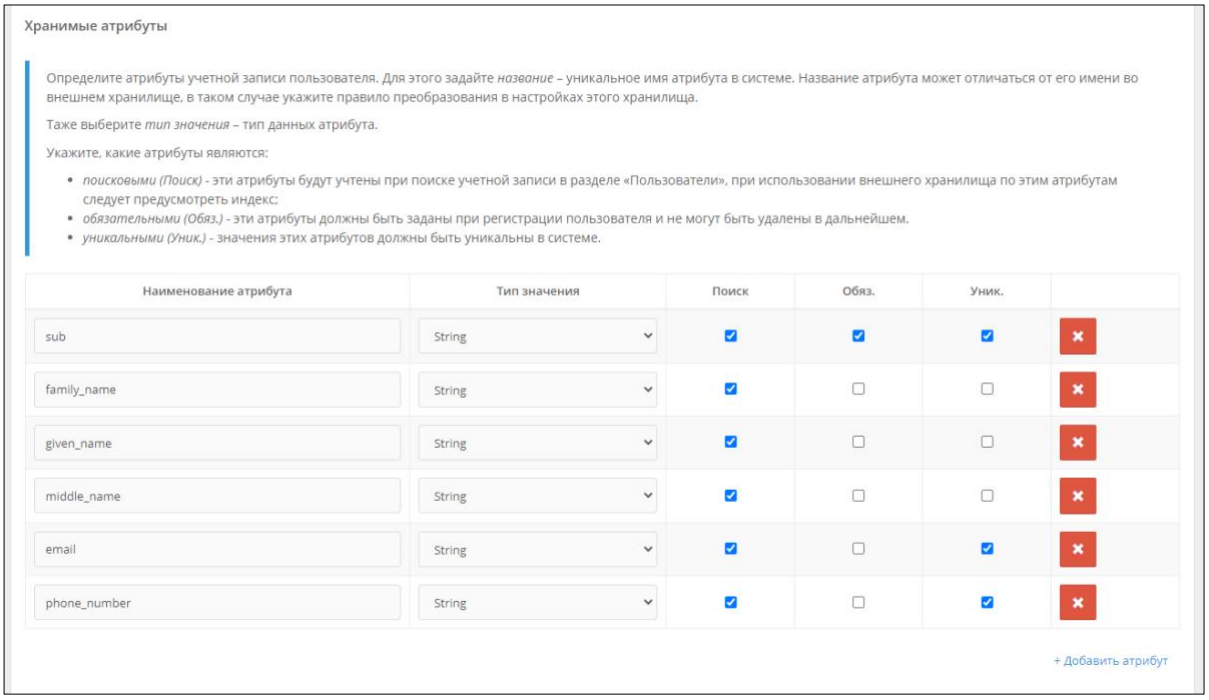

Рисунок 7 – Пример настройки хранимых атрибутов

Предусмотрена возможность назначить для атрибута LDAP-каталога транслятор, осуществляющий преобразование атрибута из хранимого в LDAP формата в требуемый формат в Blitz Identity Provider. Например, это может быть полезно при необходимости обрабатывать в Blitz Identity Provider атрибут objectGUID из LDAP-каталога Active Directory, чтобы этот атрибут представлялся не в байтовом виде, а в форме строки GUID. Настройка выполняется через конфигурационный файл, см. п. 16.1.4.

3.1.2. Настройка вычисляемых атрибутов

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##*@}[12].

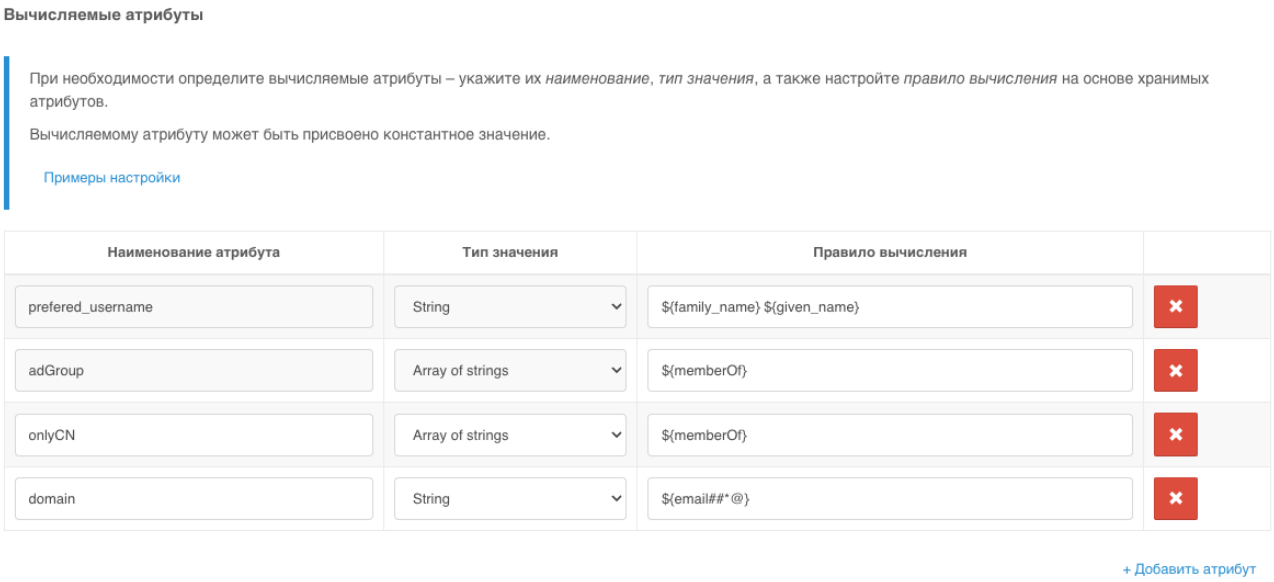

Рисунок 8 – Пример настройки вычисляемых атрибутов

3.1.3. Настройка правил 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).

[12] Справку по поддерживаемым параметрам строк подстановки можно посмотреть здесь: http://tldp.org/LDP/abs/html/parameter-substitution.html

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-}.

Рисунок 9 – Пример настройки правил преобразования входных значений

3.1.4. Настройка правил 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 соответствующие вычисляемые атрибуты, см. п. 3.1.2).

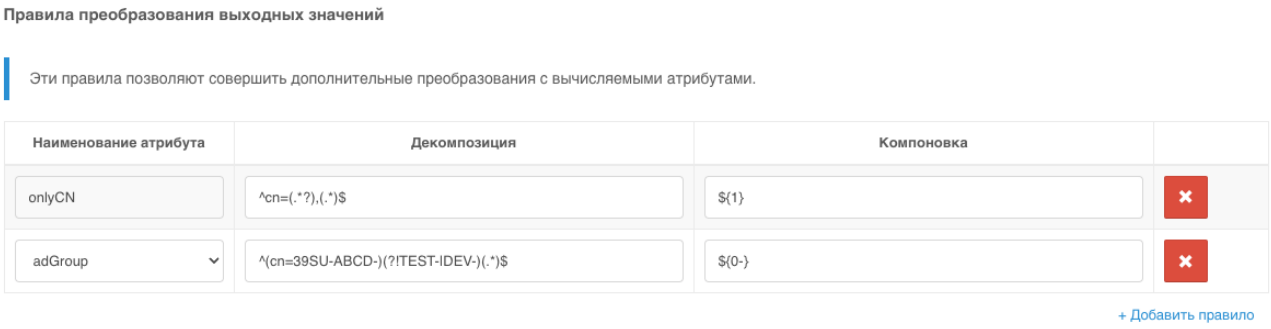

Рисунок 10 – Пример настройки правил преобразования выходных значений

3.1.5. Настройка назначения атрибутов

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

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 нужно 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 LDAP-каталога (см. п. 3.2.2).
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 номеров мобильных телефонов.

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

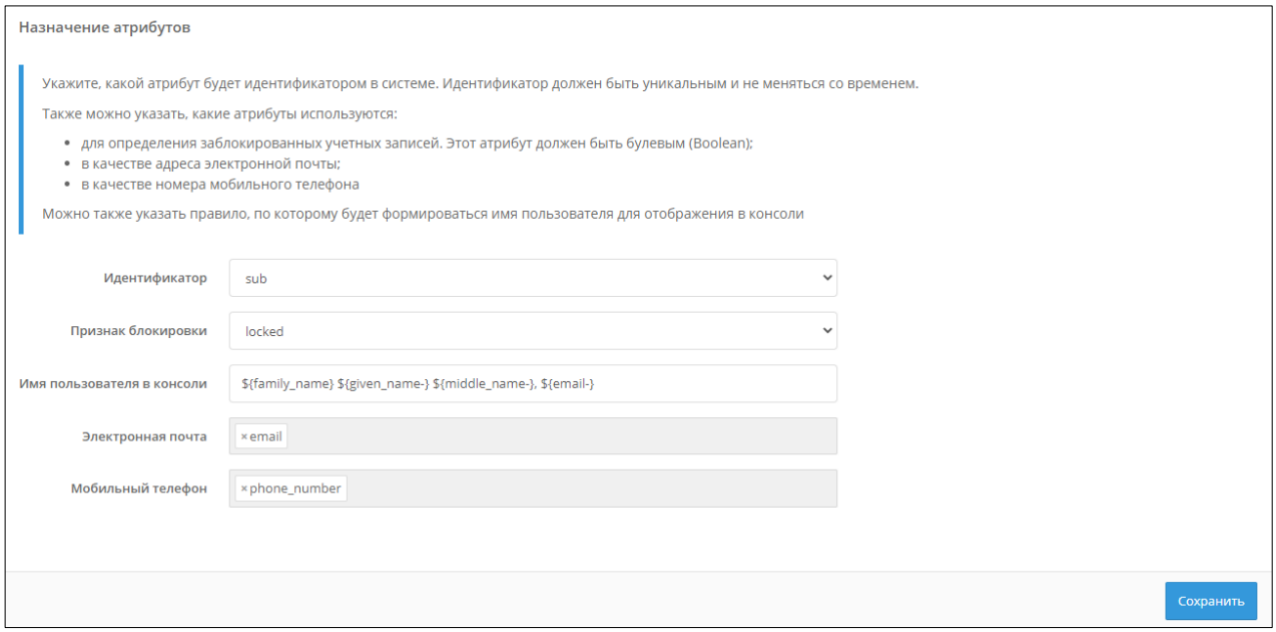

Рисунок 11 – Конфигурирование назначения атрибутов

3.2. Подключение хранилищ атрибутов

3.2.1. Типы хранилищ

В качестве хранилищ атрибутов пользователей Blitz Identity Provider позволяет использовать:

1. Внешнее (подключенное) хранилище. В качестве такового может выступать:
   - LDAP-хранилище – это может быть любой сервер, поддерживающий протокол LDAP (389 Directory Server, OpenLDAP, FreeIPA и другие), а также Microsoft Active Directory или Samba4;
   - иное хранилище, для подключения которого к Blitz Identity Provider необходимо разработать специальные REST-сервисы (см. п. 3.2.3).
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 (см. п. 3.1). По умолчанию настроено внутреннее хранилище и добавлен ряд атрибутов.

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».

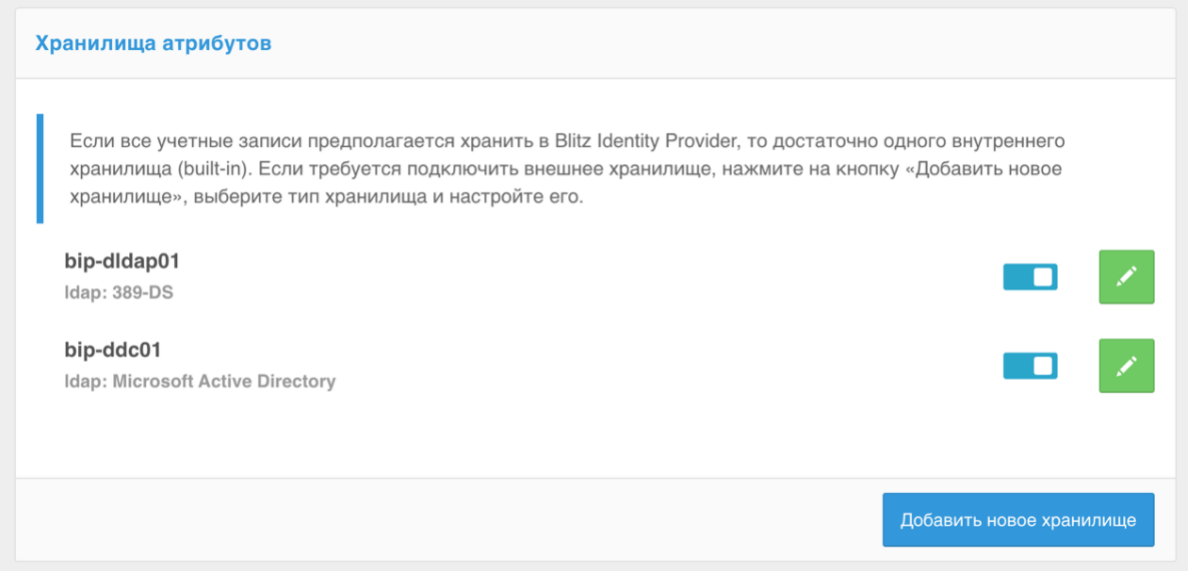

Рисунок 12 – Хранилища атрибутов

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.

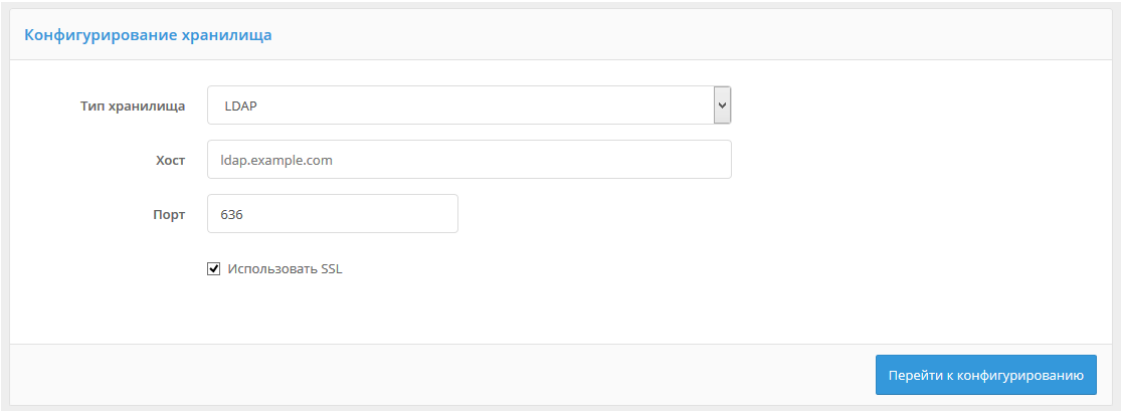

Рисунок 13 – Экран добавления хранилища учетных записей

3.2.2. Подключение хранилища по протоколу 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 галочку «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[13] 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»;
  - 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[14]), а также базовый DN – раздел каталога с учетными записями пользователей;
- указать настройки поиска – глубину поиска и максимальное число возвращаемых учетных записей (это влияет на число пользователей, отображаемых в разделе «Пользователи» консоли управления).

[13] 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 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».

[14] Допустимо указать пользователя только с правами на чтение, если хранилище используется только для чтения.

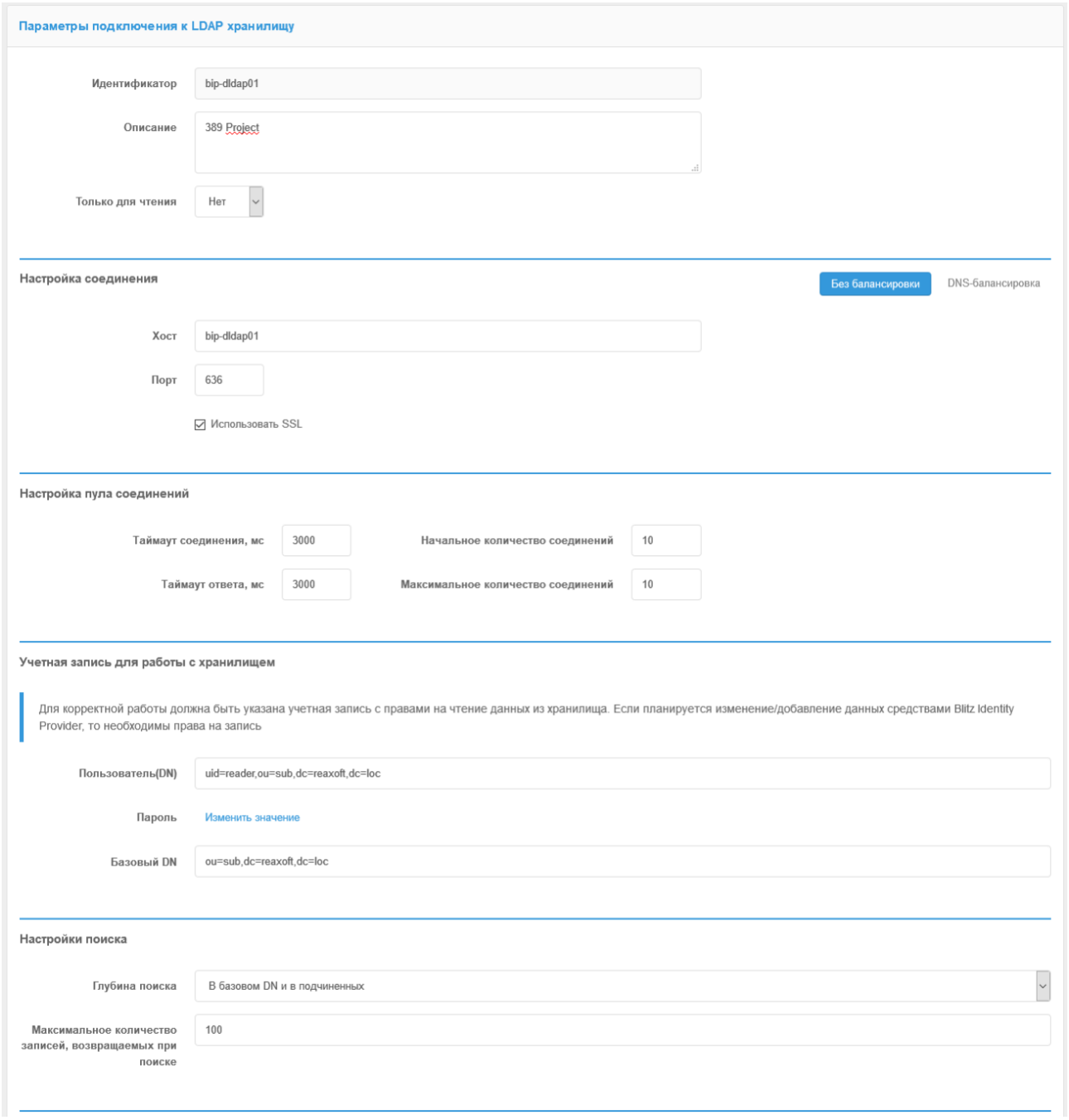

Рисунок 14 – Настройка подключения к LDAP-хранилищу данных (фрагмент)

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-каталога. 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-}.

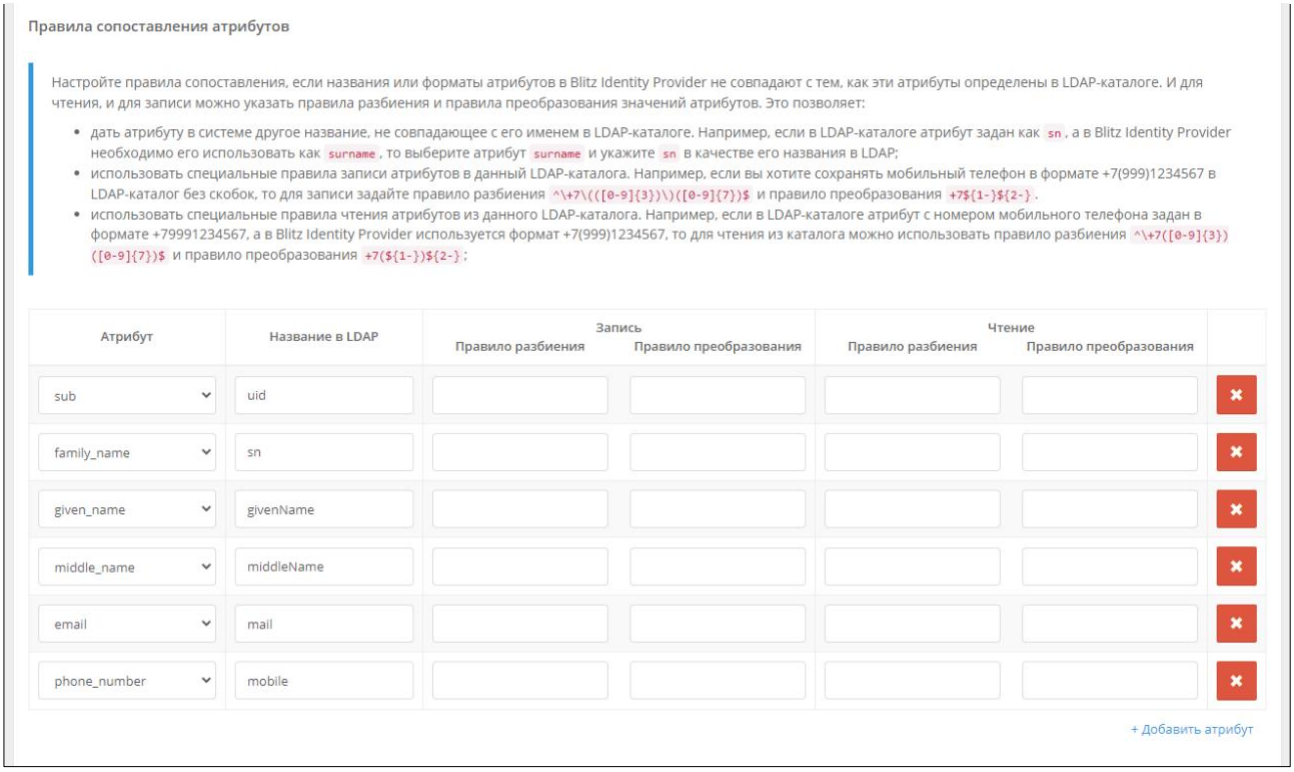

Рисунок 15 – Настройка правил сопоставления атрибутов (фрагмент)

Если хранение созданного ранее (см. п. 3.1) 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[15].

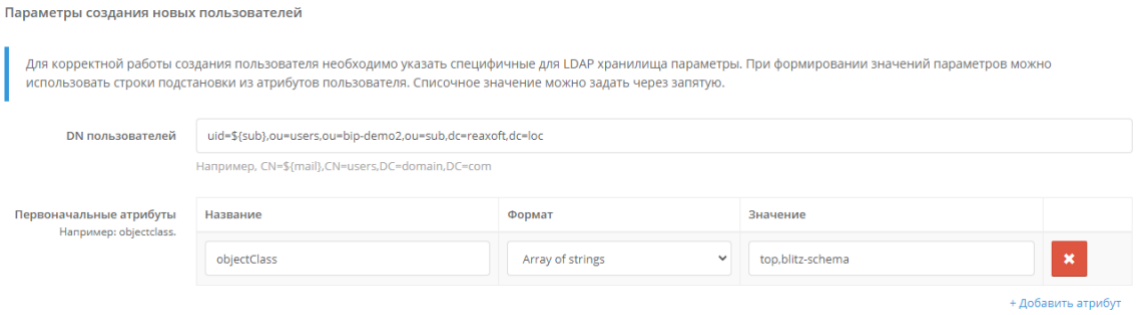

Рисунок 16 – Настройка параметров создания новых пользователей

3.2.3. Подключение к хранилищу по REST

Если в качестве источника учетных записей пользователей используется внешняя база данных (не LDAP-хранилище), то для подключения к ней нужно 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 коннектора необходимо выполнить следующие шаги:

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

Скриншот страницы с настройками подключения к хранилищу с использованием REST-сервисов представлен на рис. 17.

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

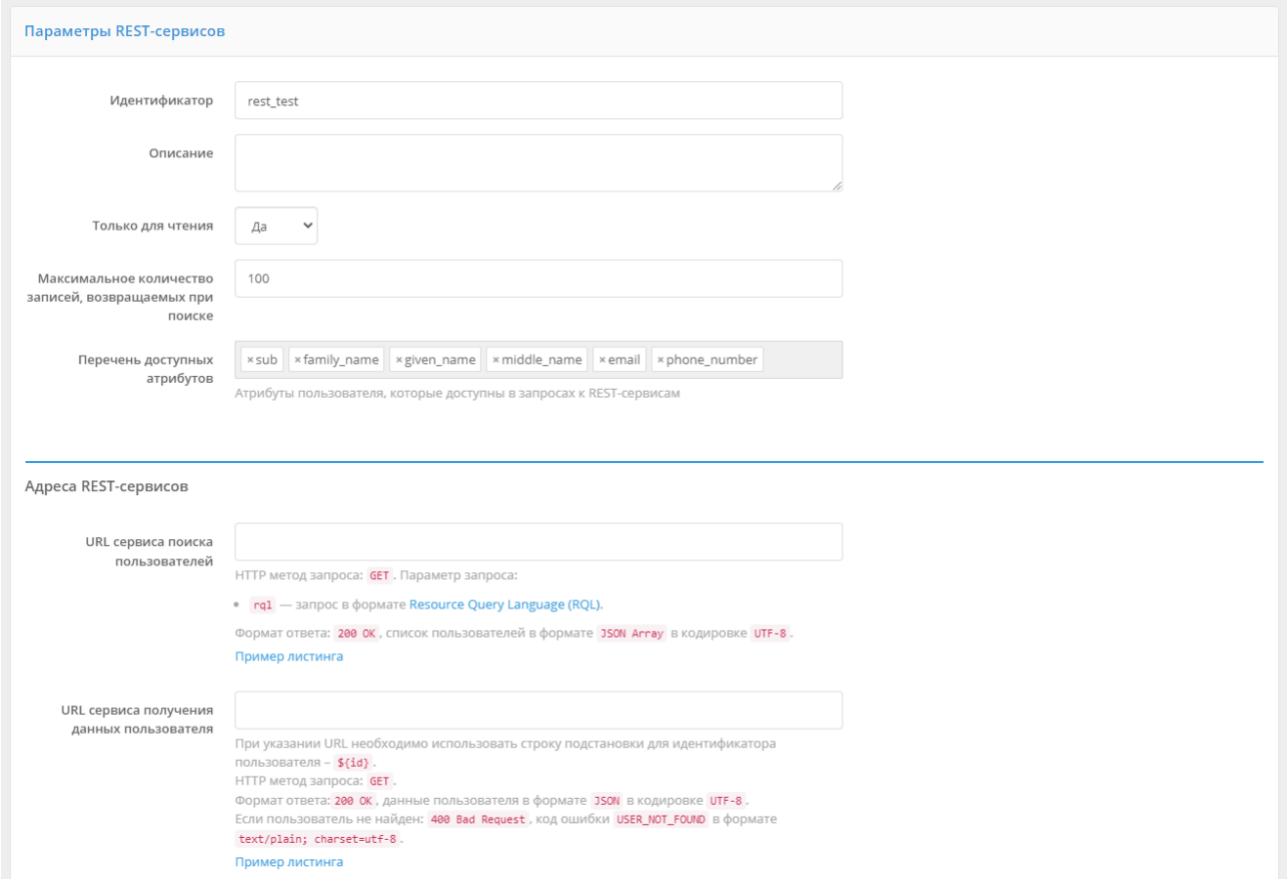

Рисунок 17 – Настройка подключения к хранилищу с использованием REST (фрагмент)

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

3.2.3.1. 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)[16] и должен 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 звезды (*)).

[16] См.: https://github.com/kriszyp/rql

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 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))
```

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))
```

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",
    "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"
    }
  }
]
```

3.2.3.2. 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 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
```

3.2.3.3. 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
```

3.2.3.4. 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
```

3.2.3.5. 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
```

3.2.3.6. 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 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"
    ]
}
```

3.2.3.7. 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
```

3.2.4. 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.

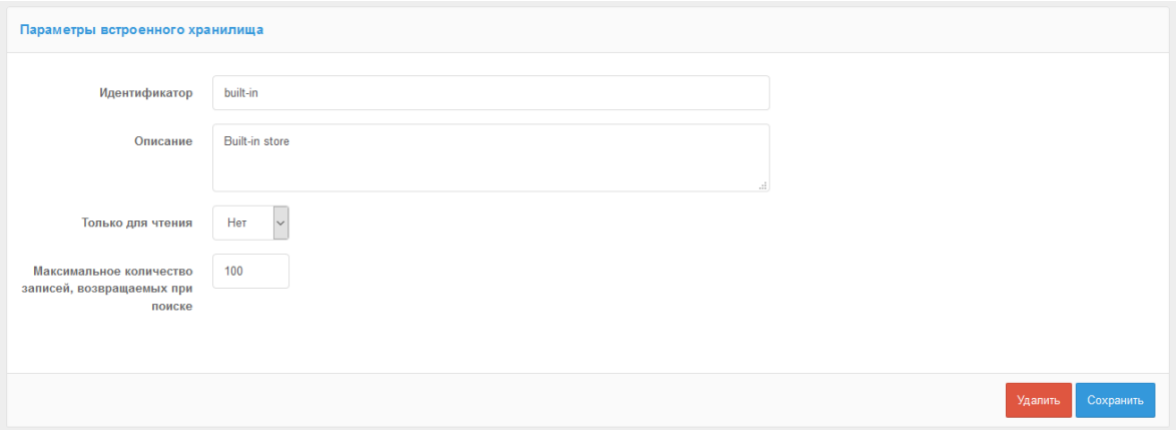

Рисунок 18 – Настройка внутреннего хранилища

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.

4. Настройки аутентификации

Настройки аутентификации задаются в разделе «Аутентификация» консоли управления (см. Рисунок 19).

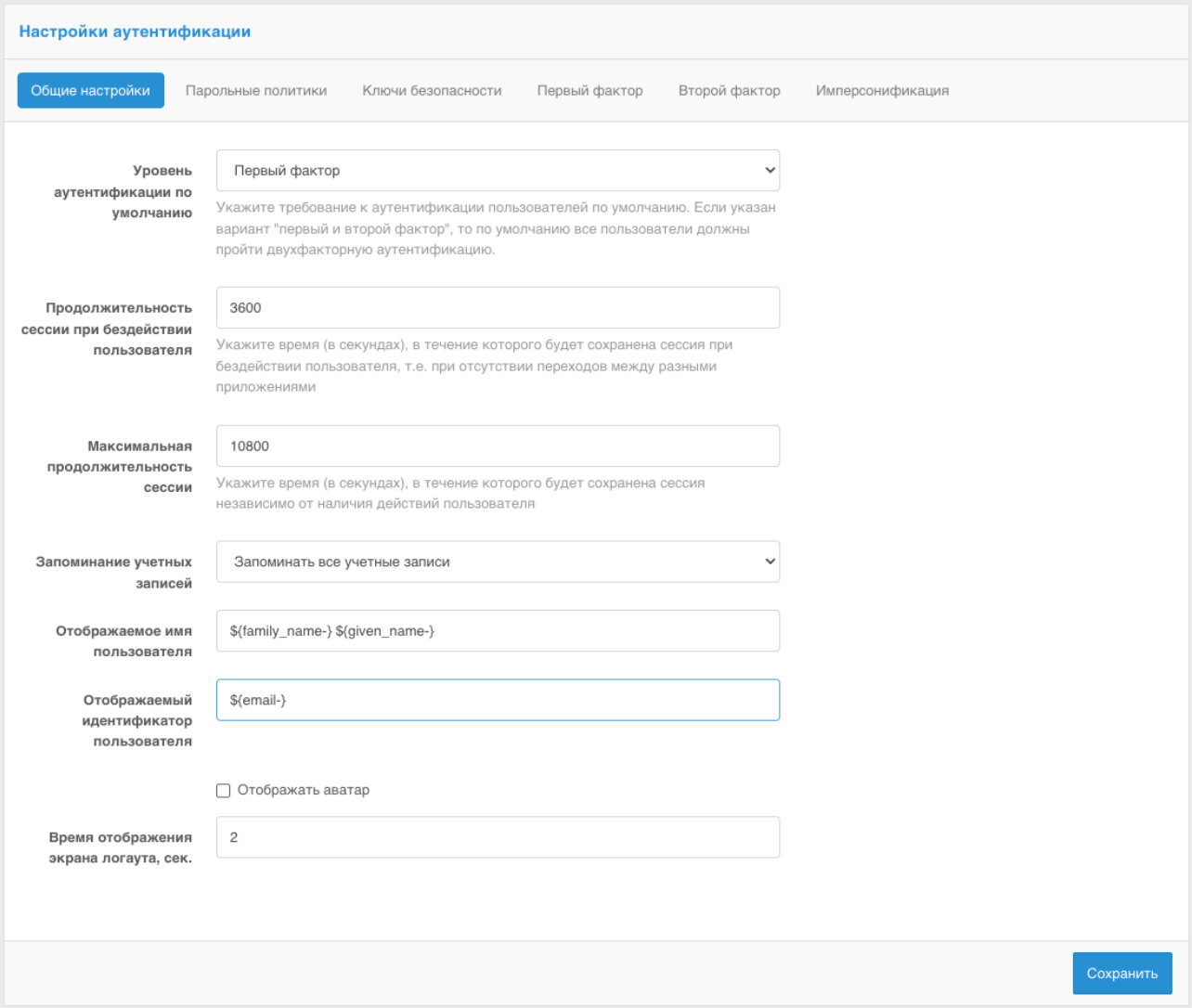

Рисунок 19 – Настройки аутентификации – вкладка «Общие настройки»

Настройки аутентификации разделены по вкладкам:

- Общие настройки – задаются общие настройки, см. описание далее в этой главе;
- Парольные политики – задаются настройки парольной политики (см. п. 4.1);
- Ключи безопасности – задаются настройки ключей безопасности (см. п. 4.2);
- Первый фактор – можно перейти к настройкам методов аутентификации, применяемых при первичной идентификации и аутентификации (см. п. 4.3);
- Второй фактор – можно перейти к настройкам методов аутентификации, применяемых для подтверждения входа (см. п. 4.3);
- Третий фактор – опциональная вкладка, отображается, только если сконфигурировано наличие метода аутентификации, применяемого дополнительно после прохождения

проверок первого и второго фактора.

На вкладке «Общие настройки» 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;
  - максимальная продолжительность сессии.
- 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 *;
- признак необходимости отображать аватар на странице входа;
- время отображения экрана логаута (в секундах) – сколько времени пользователю будет показан экран выхода до автоматического перенаправления пользователя на страницу перехода в приложение после логаута.

4.1. Настройка парольной политики

Парольные политики настраиваются на вкладке «Парольные политики» раздела «Аутентификация» консоли управления (см. Рисунок 20).

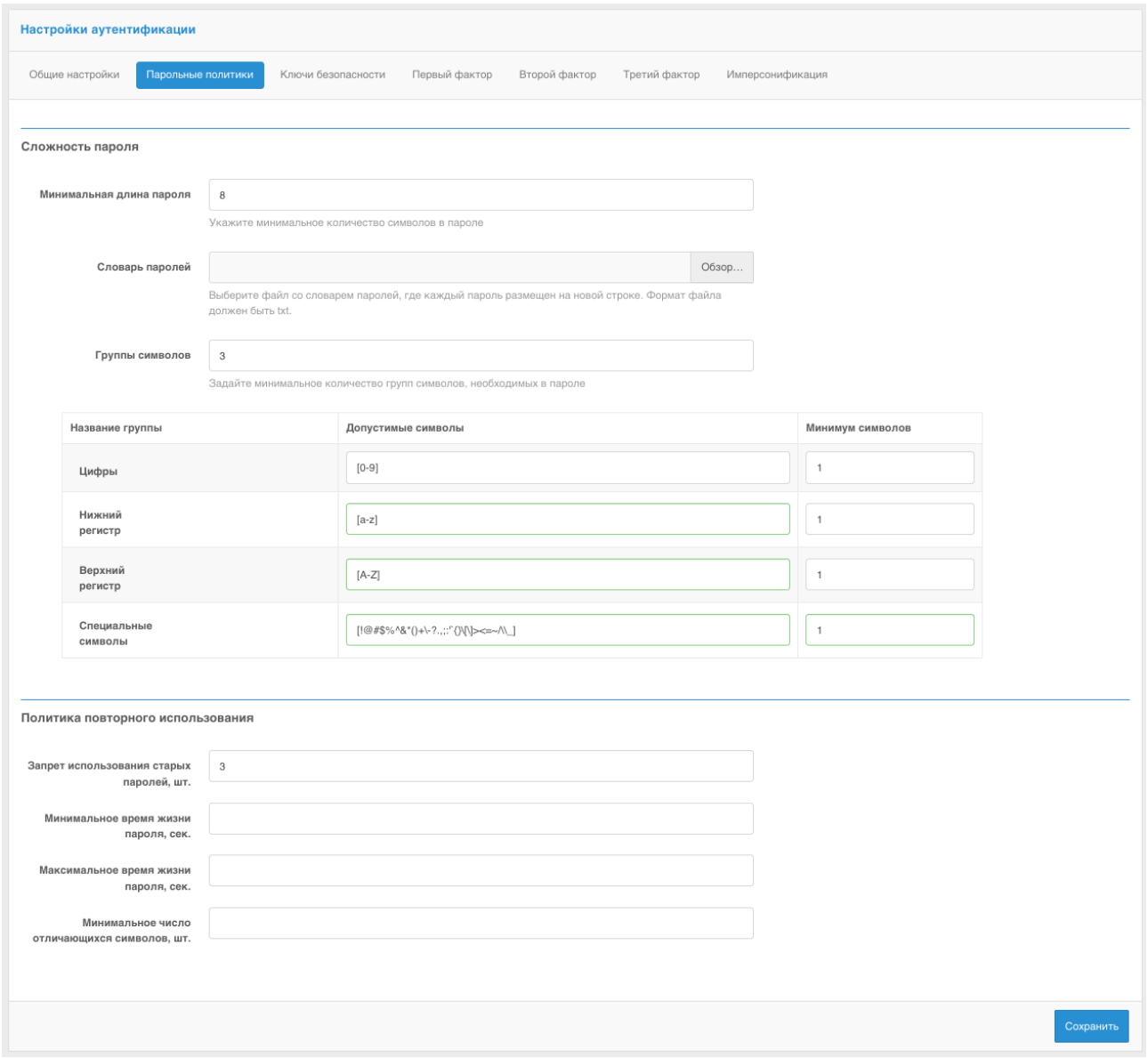

Рисунок 20 – Настройка парольных политик

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 – [0-9۹-۰], можно расширить допустимые наборы символов букв – [a-za-я] и [A-ZA-Я], добавить или убрать допустимые спецсимволы – [!@#$%^&*()+\-?.,;:'`"{}\[\]><=~/_].
- Минимум символов – сколько минимум символов из группы должно использоваться в пароле, что считалось, что группа задействована в пароле.

- Запрет использования старых паролей – настройка указывает, какое количество старых паролей должно запоминаться, чтобы при задании нового пароля не допускать ввод пароля из истории использованных паролей.
- Минимальное время жизни пароля – минимальное время жизни пароля, в секундах; пока это 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.

4.2. Настройка ключей безопасности

Blitz Identity Provider позволяет использовать для идентификации и аутентификации ключи безопасности (WebAuthn, Passkey, FIDO2, U2F).

Ключи безопасности настраиваются на вкладке «Ключи безопасности» раздела «Аутентификация» консоли управления (см. Рисунок 21).

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 скорректировать разрешенные аттестационные сертификаты необходимо выполнить настройки согласно п. 16.1.19.

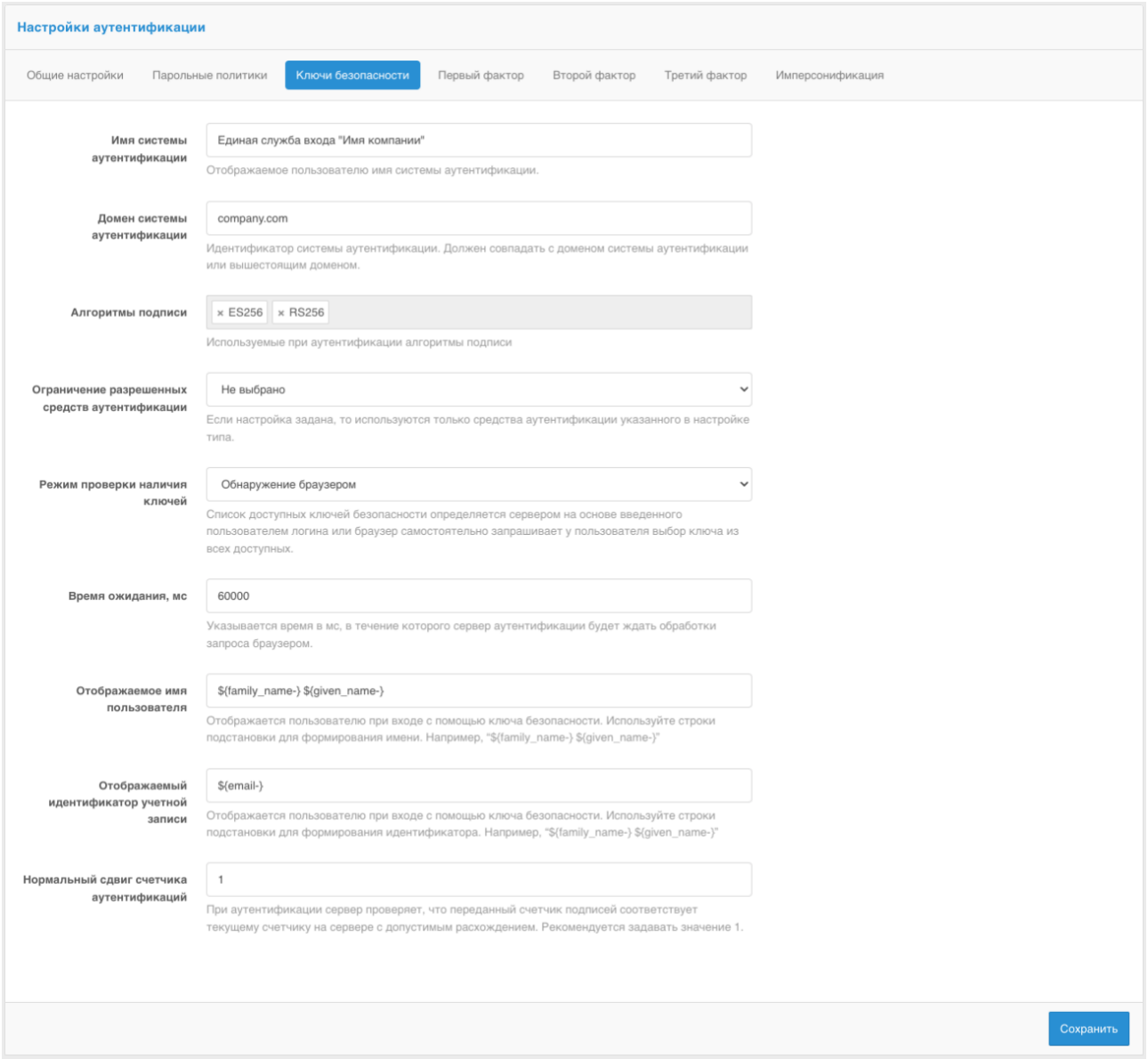

Рисунок 21 – Настройка ключей безопасности

После настройки ключей безопасности необходимо сконфигурировать методы аутентификации с использованием ключей безопасности (см. п. 4.13 и п. 4.22).

4.3. Настройка доступных методов аутентификации

Методы аутентификации сгруппированы к первому, либо ко второму фактору (второй фактор используется для «усиления» первого фактора, например, пользователю в дополнение к паролю требуется ввести специальный код, сгенерированный мобильным приложением). Чтобы включить метод аутентификации, его нужно сначала настроить.

Доступные для настройки методы первого фактора и второго фактора приведены на вкладках «первый фактор» (Рисунок 22) и «второй фактор» (Рисунок 23). Набор методов

может отличаться в зависимости от типа используемой лицензии. Для перехода к настройкам метода нужно нажать кнопку «Перейти к конфигурации метода» (при первичной настройке метода) либо ссылку «Перейти к настройкам» (для корректировки текущих заданных настроек). Руководства по настройкам каждого метода приведены в последующих разделах. Для включения или отключения метода аутентификации необходимо установить переключатель в требуемое положение.

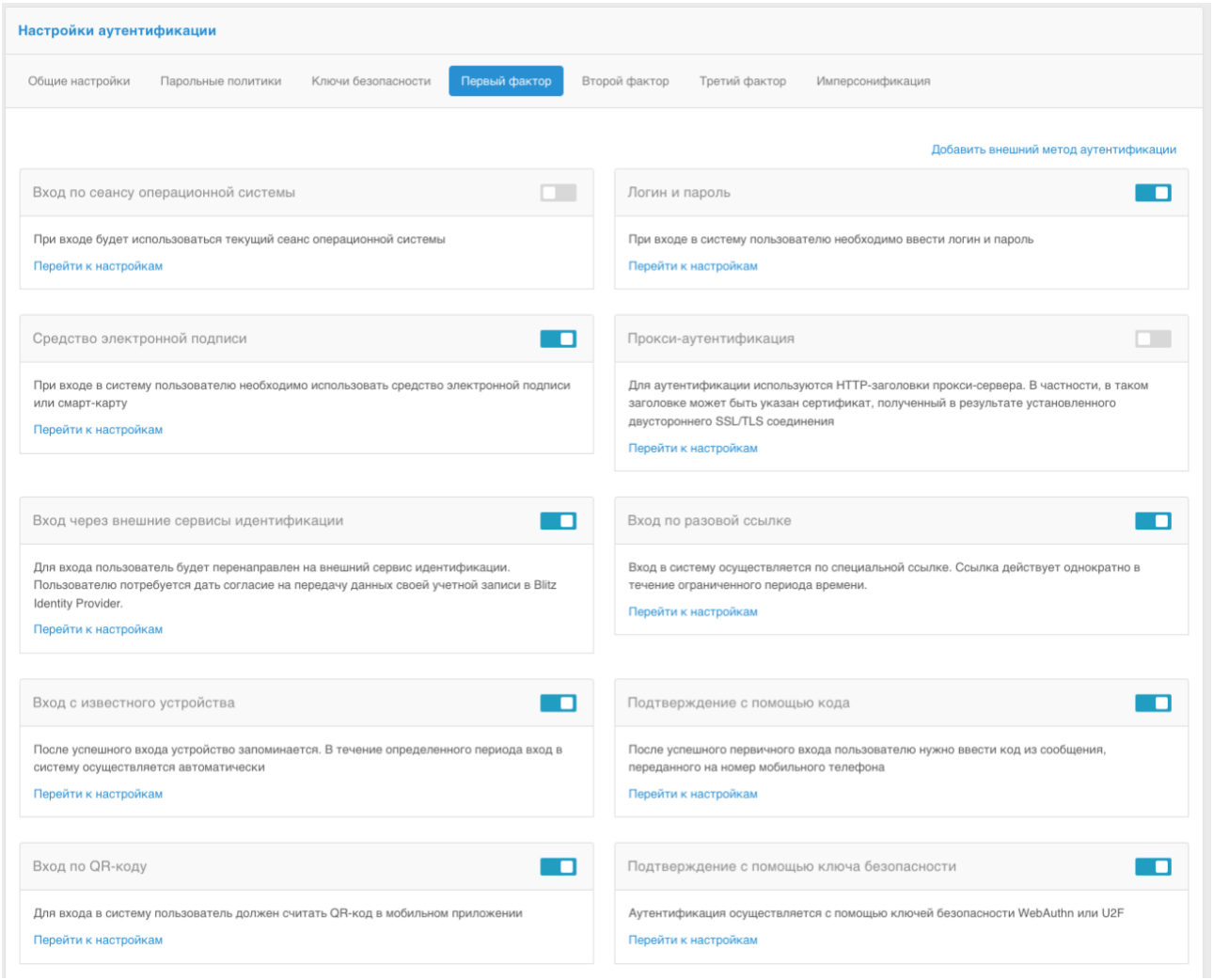

Рисунок 22 – Настройка аутентификации – вкладка «Первый фактор»

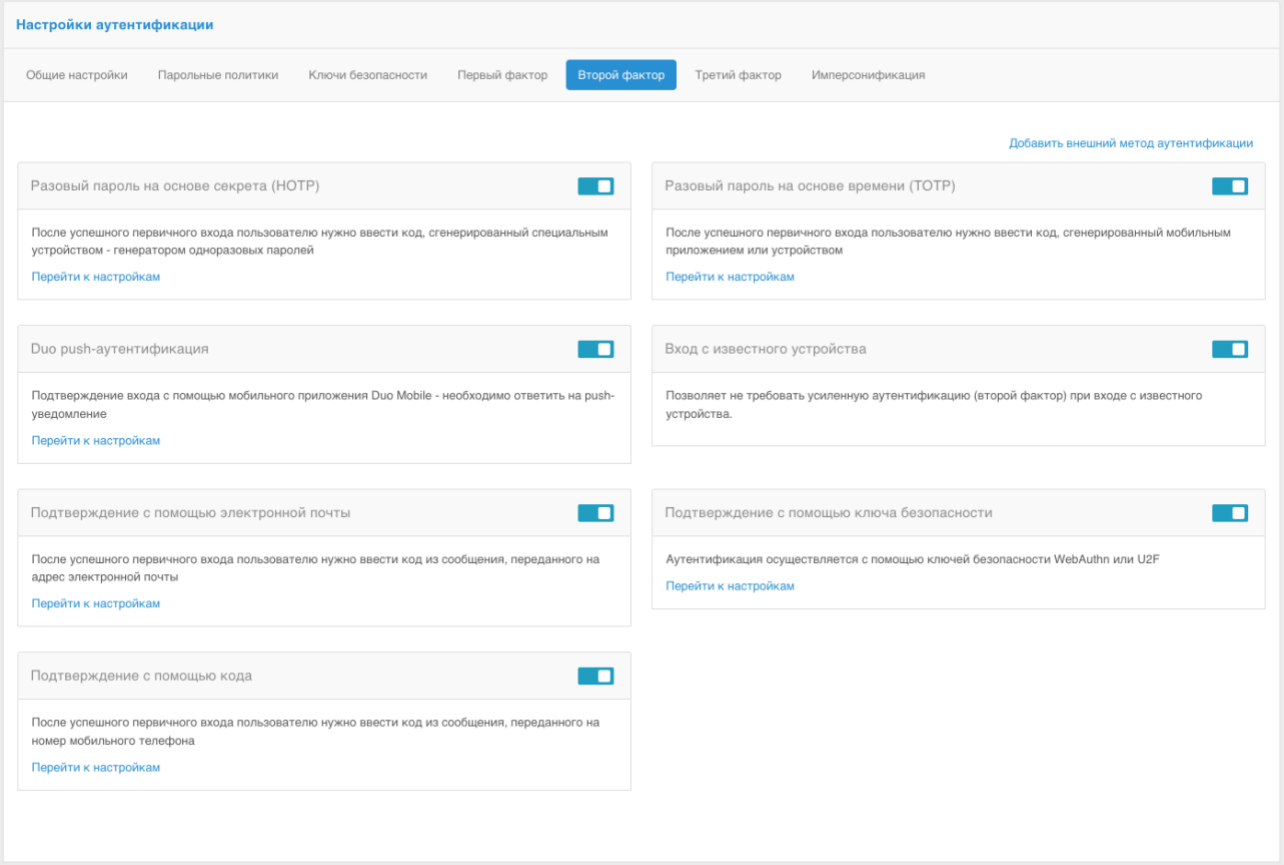

Рисунок 23 – Настройка аутентификации – вкладка «Второй фактор»

4.4. Настройка входа 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 (пример настройки см. Рисунок 24);

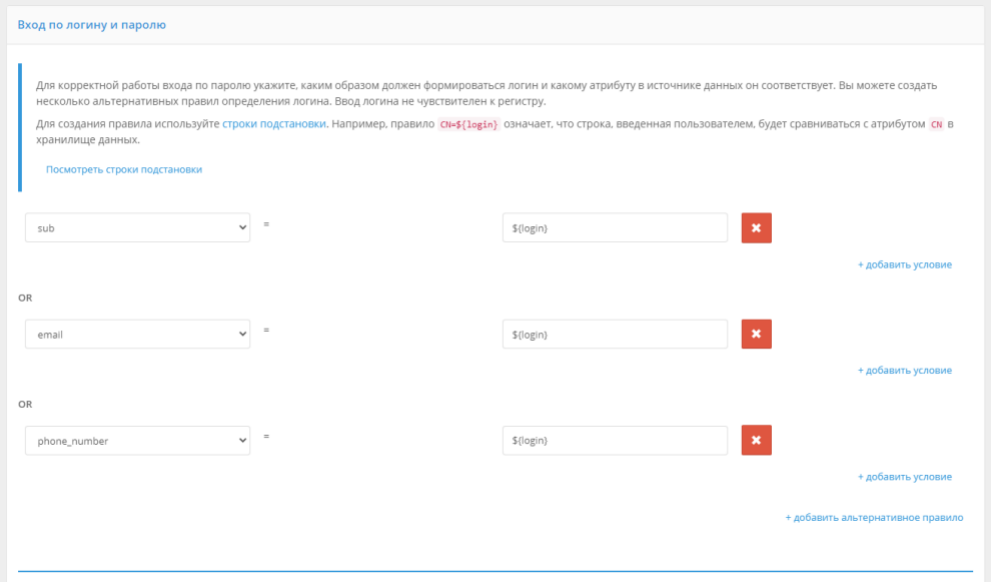

Рисунок 24 – Настройка входа по логину и паролю

В настройках входа по логину и пароля можно включить проверку на соответствие пароля парольной политике (см. п. 4.1). 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 (см. Рисунок 25):

- 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).

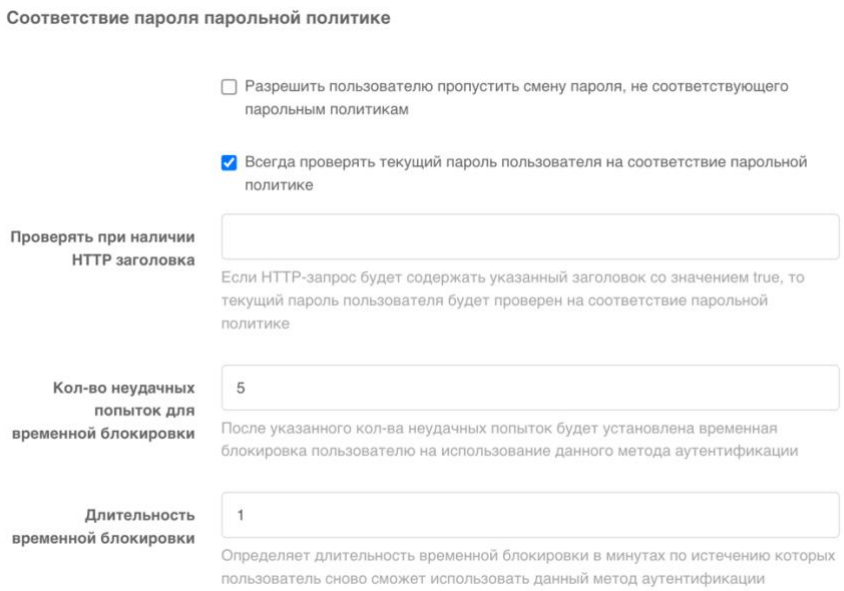

Рисунок 25 – Настройка проверки на соответствие пароля парольной политике при входе

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

Администратор в настройке «Защита» может выбрать следующие режимы защиты:

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

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

Пример настройки защиты от подбора пароля представлен ниже (Рисунок 26).

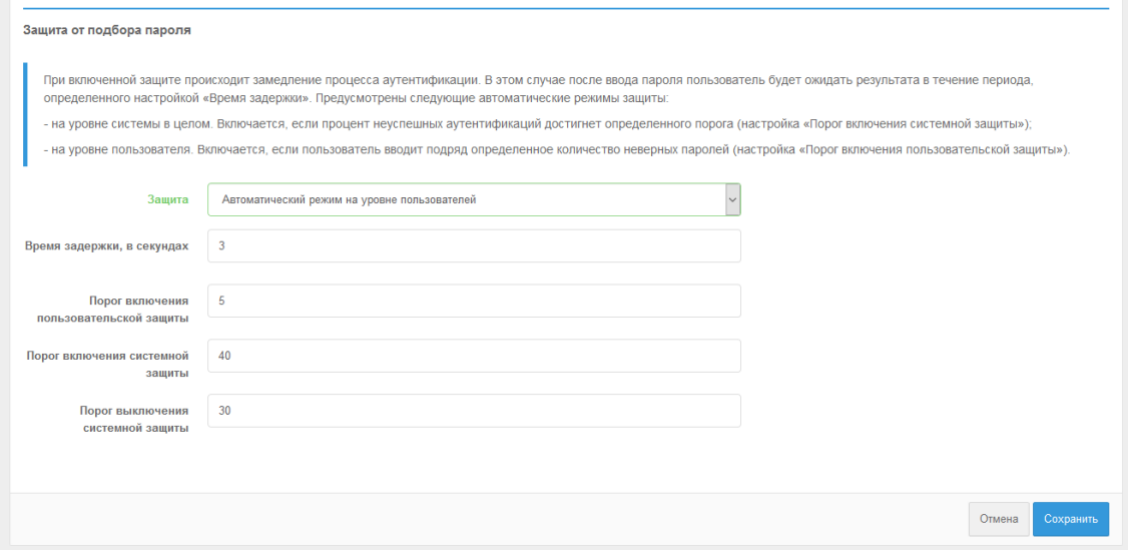

Рисунок 26 – Настройка защиты от подбора пароля

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.

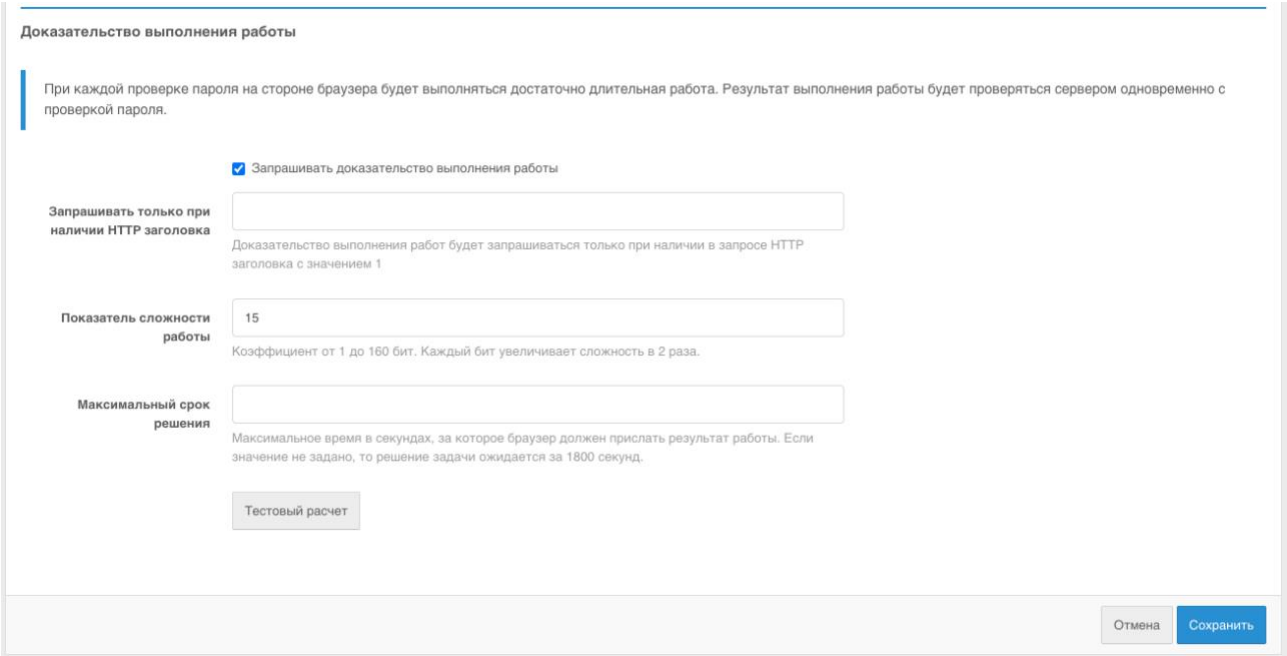

Рисунок 27 – Настройка доказательства выполнения работы браузером

В блоке настроек «Доказательство выполнения работы» можно настроить следующее:

- включить настройку «Запрашивать доказательство выполнения работы»;
- при необходимости задать настройку «Запрашивать только при наличии HTTP заголовка» – это полезно, если нужно оставить возможность автотестам выполнять

вход по паролю без необходимости прохождения проверки. В этом случае на веб-сервере нужно для пользовательских запросов настроить установку заголовка из этой настройки, а для запросов, приходящих от автотестов, заголовок не устанавливать.

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

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

В блоке «Правила выбора хранилища атрибутов» (см. Рисунок 28) 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.

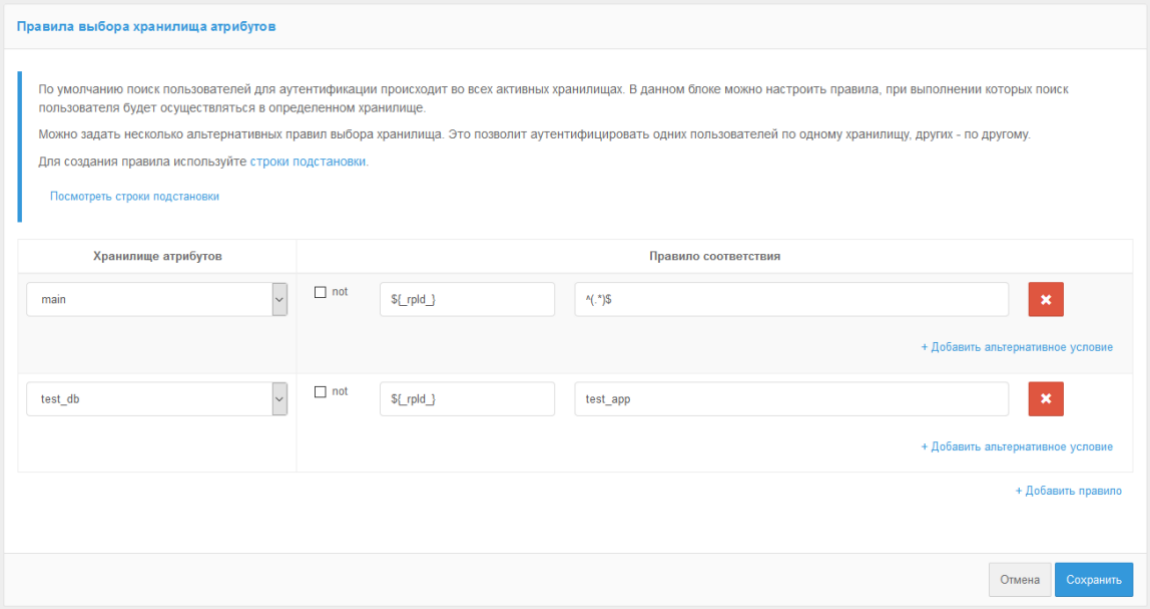

Рисунок 28 – Настройка правил выбора хранилища атрибутов

4.5. Настройка входа с помощью средства электронной подписи

4.5.1. Настройка метода аутентификации в консоли управления

При использовании для аутентификации средств электронной подписи необходимо:

- в блоке настроек «Сертификаты» загрузить сертификаты удостоверяющих центров, подтверждающих подлинность сертификатов ключей электронной подписи или настроить взаимодействие с внешним сервисом проверки электронной подписи (см. п. 16.1.5);
- 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 см. Рисунок 29).
- 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;

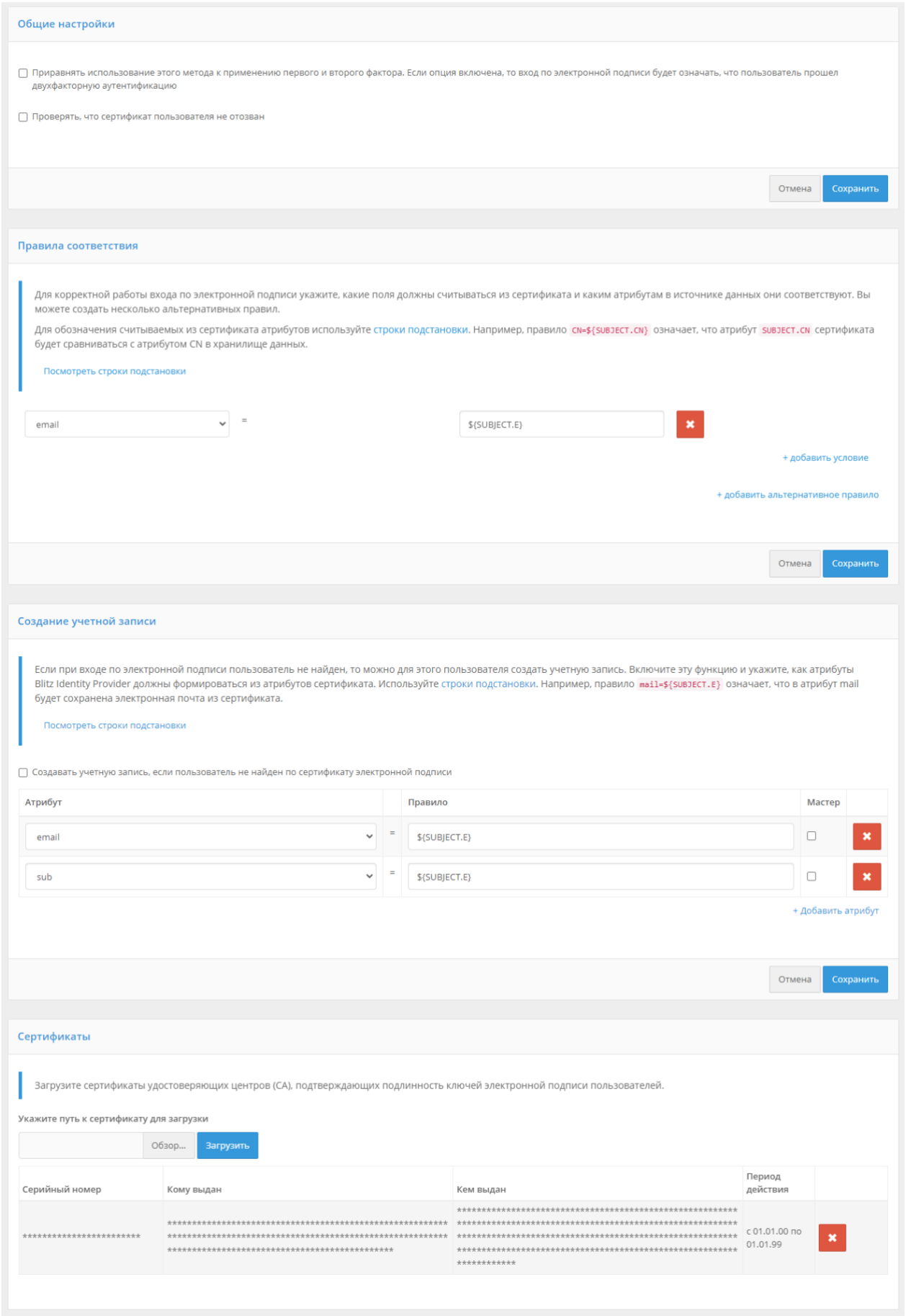

Рисунок 29 – Настройка входа по электронной подписи

4.5.2. 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 настройки использования плагина электронной подписи (см. п. 16.1.6).

4.6. Настройка входа через внешние сервисы идентификации

Возможен вход с использованием следующих внешних сервисов идентификации:

- поставщика идентификации Apple ID;
- поставщика идентификации социальной сети Facebook;
- поставщика идентификации социальной сети ВКонтакте;
- поставщика идентификации Яндекс;
- поставщика идентификации Google;
- поставщика идентификации социальной сети Одноклассники;
- поставщика идентификации Mail.ru (Mail ID);
- поставщика идентификации VK ID;
- поставщика идентификации ЕСИА (gosuslugi.ru);
- поставщика идентификации ЕСИА в режиме «Цифровой профиль» (gosuslugi.ru);
- поставщика идентификации Сбер ID;
- поставщика идентификации Tinkoff ID;
- поставщика идентификации ВТБ ID;
- поставщика идентификации СберБизнес ID;

- поставщика идентификации Альфа ID;
- поставщика идентификации Mos ID (СУДИР);
- поставщика идентификации, работающего по OpenID Connect;
- поставщика идентификации, работающего по SAML.

Подключения к внешним сервисам идентификации должны быть предварительно сконфигурированы в консоли управления в разделе «Поставщики идентификации» (см. п. 8 документа).

В разделе настроек «Вход через внешние сервисы идентификации» необходимо выбрать, какие из настроенных поставщиков идентификации должны использоваться при входе (см. Рисунок 30).

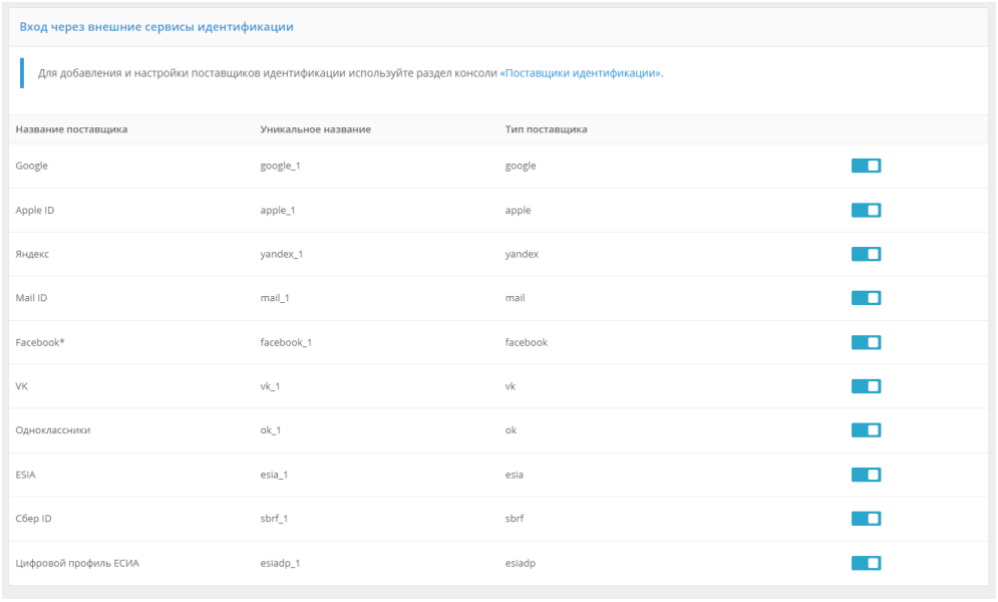

Рисунок 30 – Включение необходимых внешних сервисов идентификации

4.7. Настройка входа с помощью прокси-аутентификации

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

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 в запросе 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 на рисунке 31.

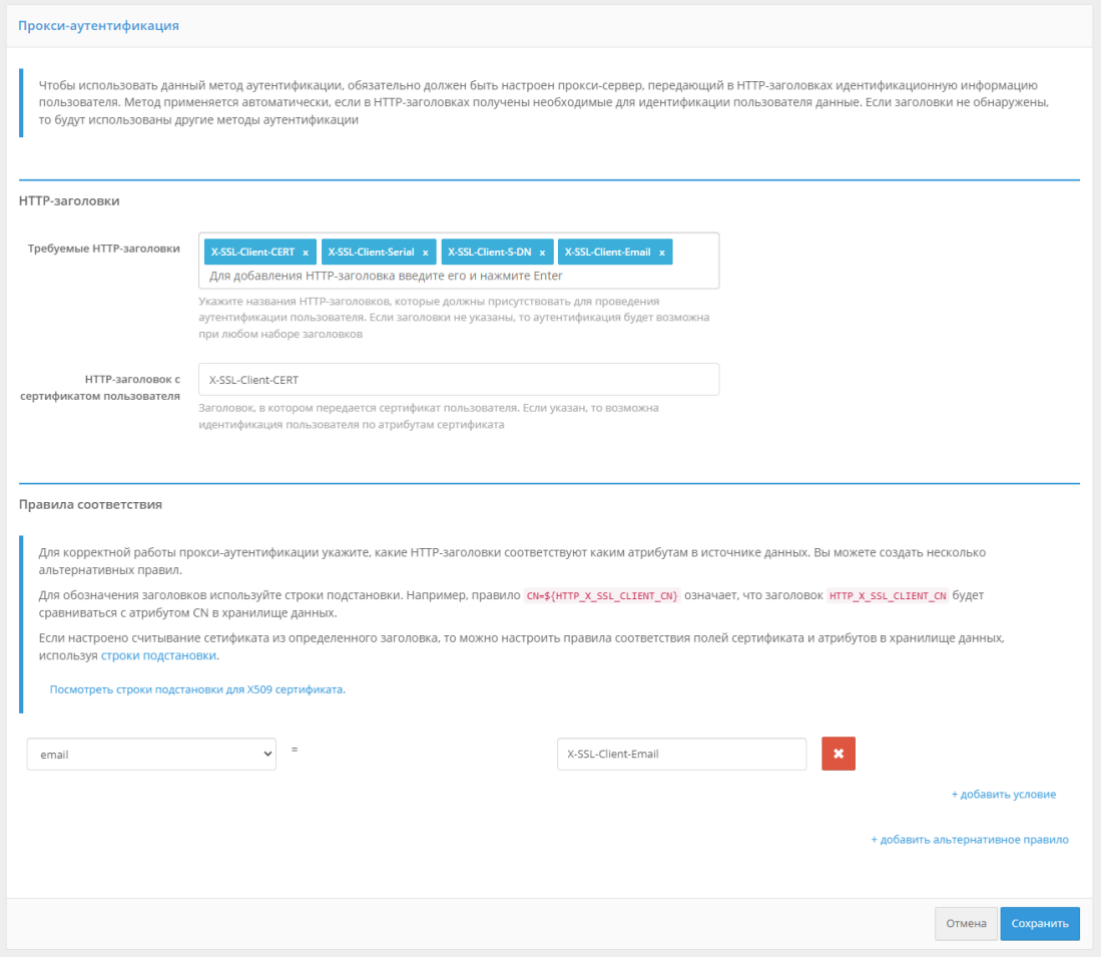

Рисунок 31 – Настройка входа с помощью прокси-аутентификации

4.8. Настройка входа с помощью сеанса операционной системы

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

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

выполнены следующие настройки (см. описания далее в подразделах):

1. Настройки контролера домена (Kerberos-сервера).
2. Настройки в консоли управления Blitz Identity Provider.
3. Настройки браузеров пользователей.
4. Настройки запуска приложений Blitz Identity Provider.
5. Настройки веб-сервера.

4.8.1. Настройки контроллера домена (Kerberos-сервера)

На 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 отметить опции «This account supports Kerberos AES 256 bit encryption» и «Do not require Kerberos preauthentication» (см. Рисунок 32).

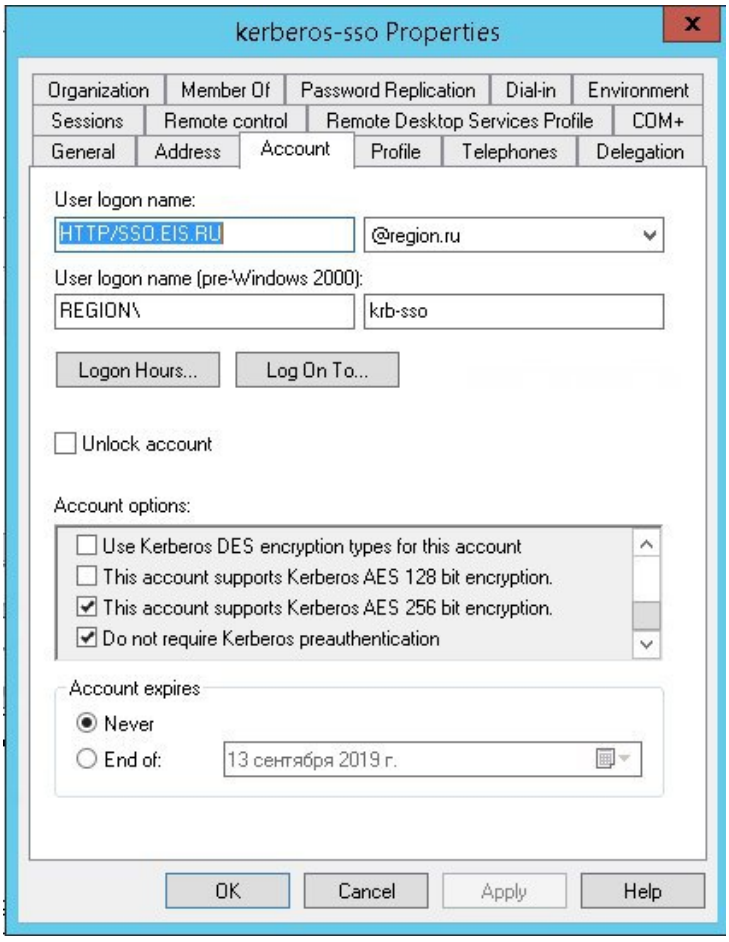

Рисунок 32 – Свойства Kerberos

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:

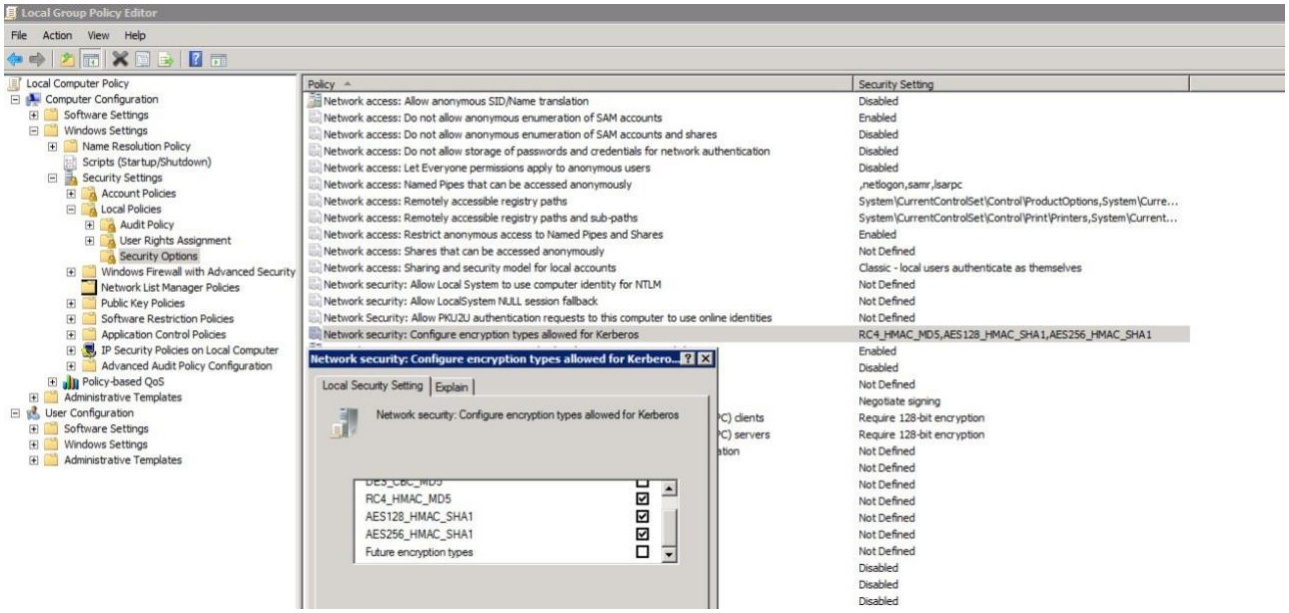

Рисунок 33 – Настройка политик шифрования

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.

4.8.2. 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 Kerberos-службе (см. Рисунок 34).

При необходимости можно:

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

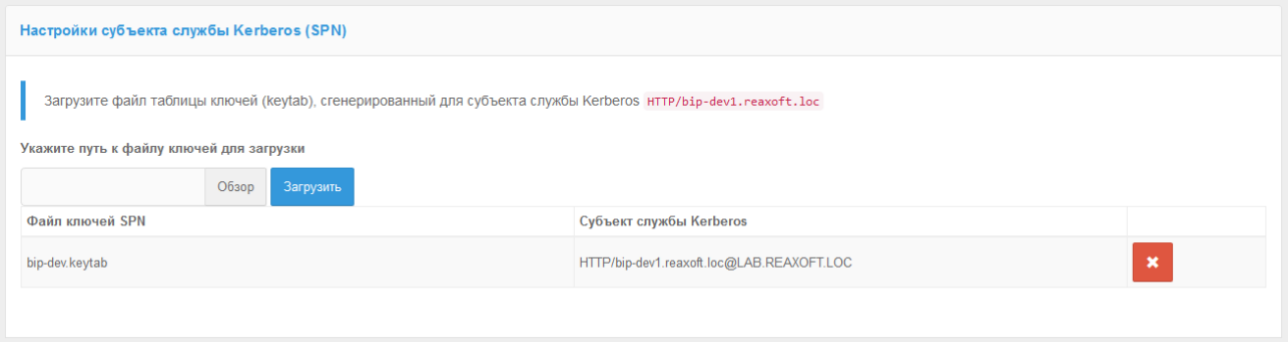

Рисунок 34 – Keytab-файл успешно загружен

Далее необходимо определить параметры соответствия Kerberos-токена (TGS) и учетной записи в Blitz Identity Provider (см. Рисунок 35).

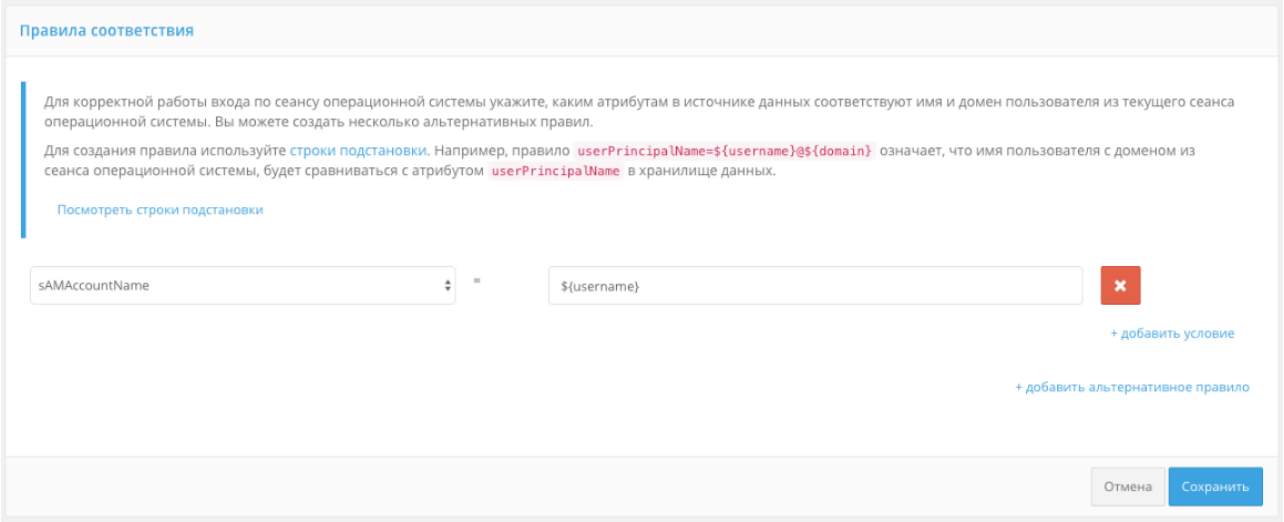

Рисунок 35 – Настройка соответствия Kerberos-идентификатора пользователя и его учетной записи в хранилище

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

Далее необходимо установить параметры задержек при использовании метода входа с использованием сеанса операционной системы (см. Рисунок 36).

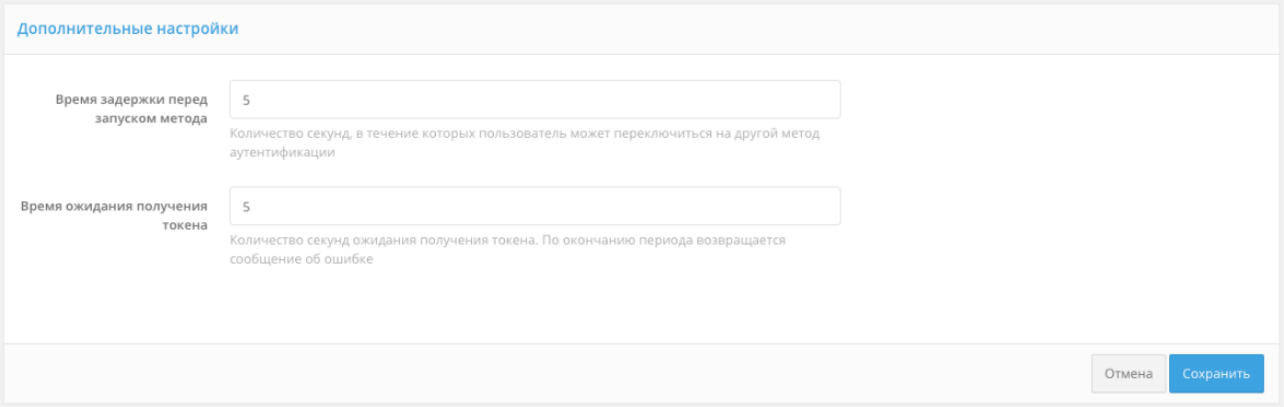

Рисунок 36 – Дополнительные настройки

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 (подробнее см. п. 4.4).

4.8.3. 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» (см. Рисунок 37);
- 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» (см. Рисунок 38).

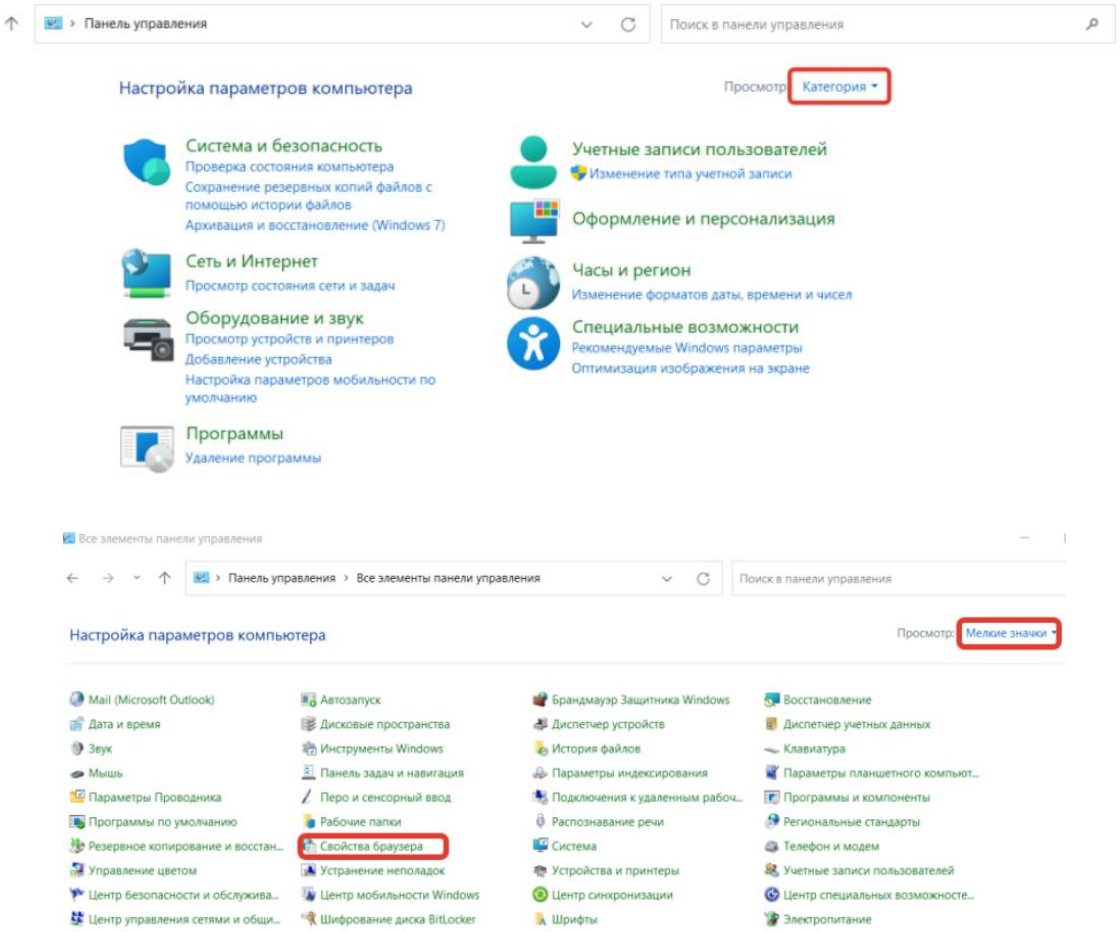

Рисунок 37 – Настройки Internet Explorer для Kerberos – включение Blitz Identity Provider в ресурсы Локальной вычислительной сети

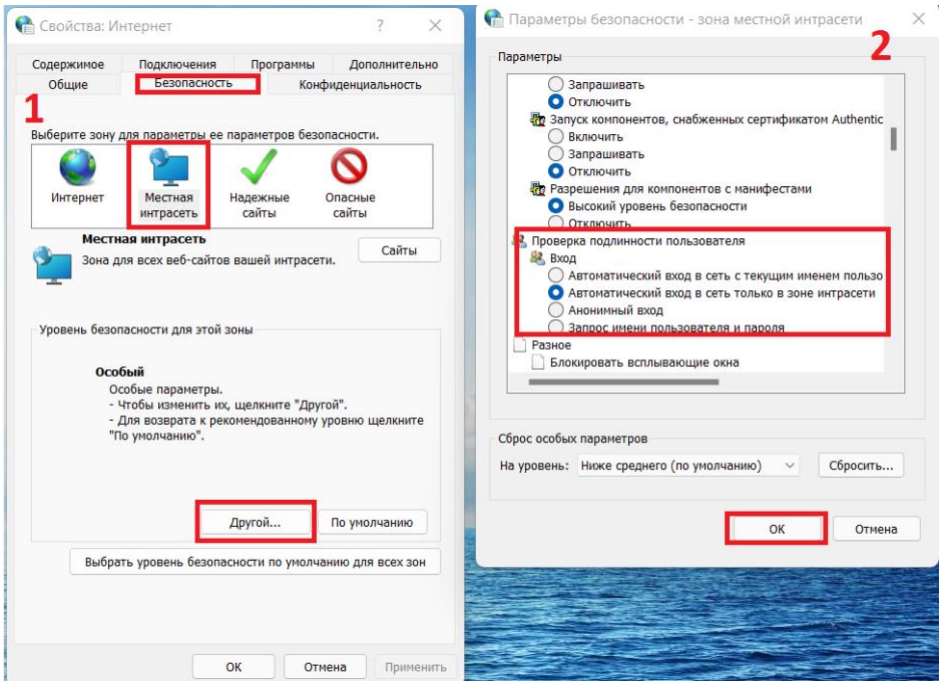

Рисунок 38 – Настройки Internet Explorer для Kerberos – включение встроенной идентификации

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 звезду (*) и указать несколько URL через запятую, например: https://*.idp.domain.ru,http://*.idp.domain.ru. Закрыть всплывающее окно кнопкой «ОК»;
- 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 звезду (*) и указать несколько URL через запятую, например: https://*.idp.domain.ru,http://*.idp.domain.ru. Закрыть всплывающее окно кнопкой «ОК»;

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

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

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

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

Для Apple Safari в macOS отдельная настройка не требуется.

4.8.4. 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
```

4.8.5. 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;
```

4.8.6. 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.

4.9. Настройка входа с помощью кодов подтверждения

Можно использовать отправляемые в мобильное приложение push-уведомления или SMS-сообщения для проверки:

- первого фактора аутентификации;
- второго фактора аутентификации (см. п. 4.18).

Для использования кодов подтверждения необходимо:

- настроить и включить метод аутентификации «Подтверждение с помощью кода» (см. Рисунок 39). Необходимо настроить:
  - способ идентификации учетной записи – задать регулярное выражение. Например, правило phone_number=${login} означает, что введенный пользователем логин в форме входа будет сопоставлен 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);
  - сконфигурировать 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};
- настроить подключение Blitz Identity Provider к SMS-шлюзу и сервису отправки push-уведомления (см. п. 13.1).

Если у пользователя не задан номер мобильного телефона, то он не сможет использовать

способ подтверждения входа с помощью кода подтверждения, отправляемого по SMS.

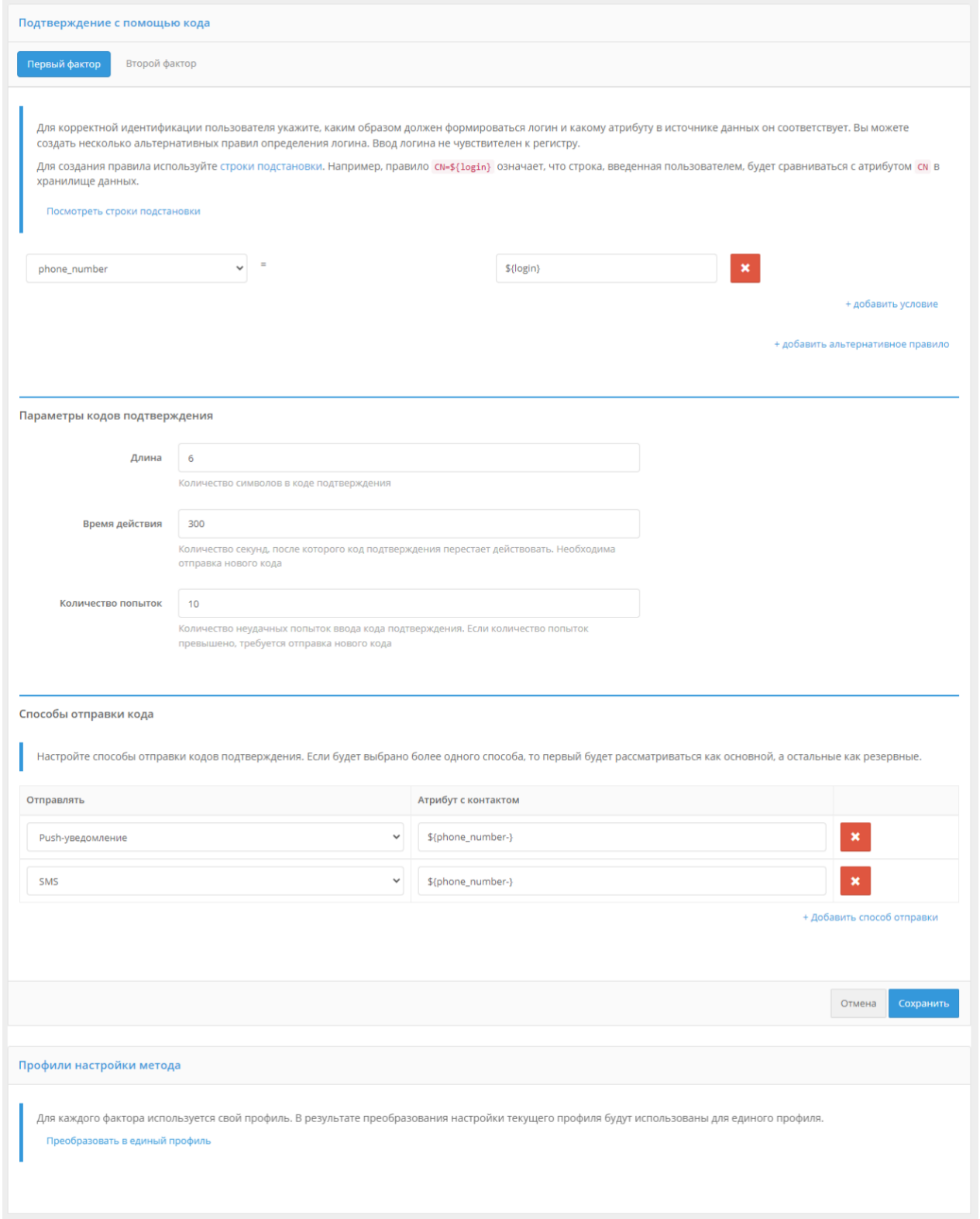

Рисунок 39 – Настройки входа с помощью кода подтверждения

По умолчанию используются единые настройки для кодов подтверждения, отправляемых для проверки первого и второго фактора. Для разделения настроек необходимо перейти по ссылке «Сконфигурировать профиль для каждого фактора» в блоке «Профили

настройки метода». Тогда настройки будут разведены и можно будет переключаться между первым и вторым фактором.

При необходимости перейти к единым настройкам следует перейти по ссылке «Преобразовать в единый профиль» в блоке «Профили настройки метода».

4.10. Настройка входа с известного устройства

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

Настройка метода включает в себя указание длительности запоминания устройства. Также можно установить, что при входе с запомненного устройства не будет требоваться двухфакторная аутентификация (опция «Приравнять использование этого метода к применению первого и второго фактора»). Если эта опция включена, то вход с известного устройства будет означать, что пользователь прошел двухфакторную аутентификацию (см. Рисунок 40).

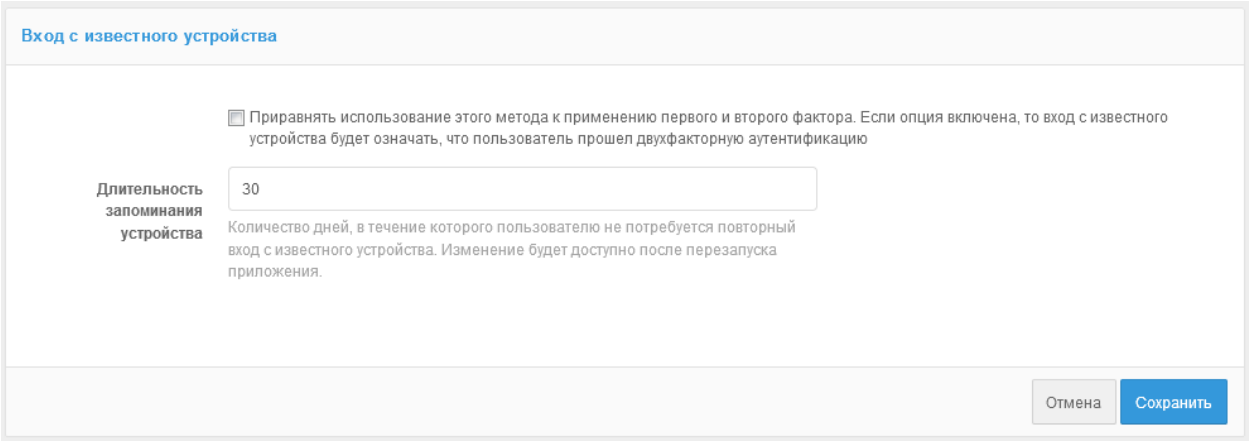

Рисунок 40 – Настройка входа с известного устройства

4.11. 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[17].

Настройка метода включает в себя указание времени действия ссылки (см. Рисунок 41), используемой для автоматического входа. Чтобы сработал автоматический вход, с момента

[17] Подробно этот сценарий описан в «Руководстве по интеграции» в главе «Открытие веб-ресурсов из мобильного приложения в режиме сквозной аутентификации»

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.

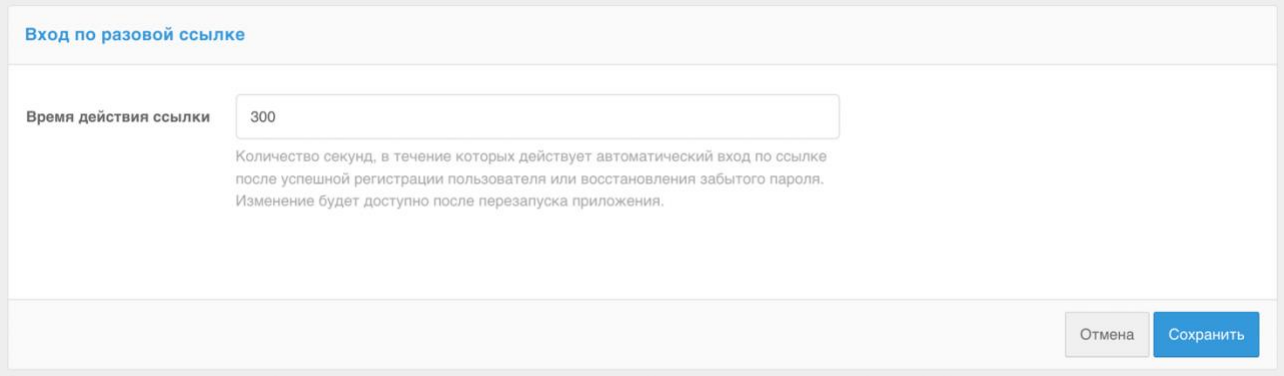

Рисунок 41 – Настройка времени действия ссылки

4.12. 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[18] 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

[18] Описание интеграции мобильного приложения с Blitz Identity Provider описано в «Руководство по интеграции» в главе «Добавление в мобильное приложение функции входа по QR-коду».

пользователя в приложение или запрос входа отклоняется.

Настройка метода включает в себя указание следующих параметров (см. Рисунок 42):

- 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.

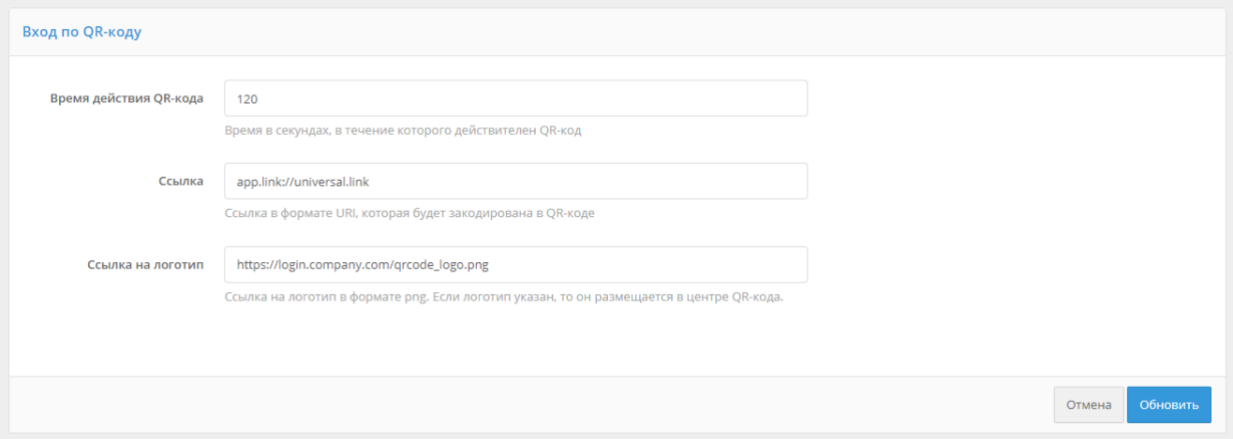

Рисунок 42 – Настройка входа по QR-коду

4.13. Вход с помощью ключей безопасности (WebAuthn, Passkey, FIDO2)

Можно использовать ключи безопасности (WebAuthn, Passkey, FIDO2[19]) для входа в Blitz Identity Provider. Для взаимодействия с ключами безопасности используется спецификация WebAuthn[20]. 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:

[19] См.: https://fidoalliance.org/fido2/
[20] См.: https://www.w3.org/TR/webauthn-2/

- 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.

Для использования ключей безопасности необходимо задать настройки ключей безопасности (см. п. 4.2), а также сконфигурировать метод аутентификации в консоли управления (см. Рисунок 43). В консоли управления необходимо задать следующие настройки:

- 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, при выполнении которых поиск пользователя будет осуществляться в

определенном хранилище (подробнее см. п. 4.4).

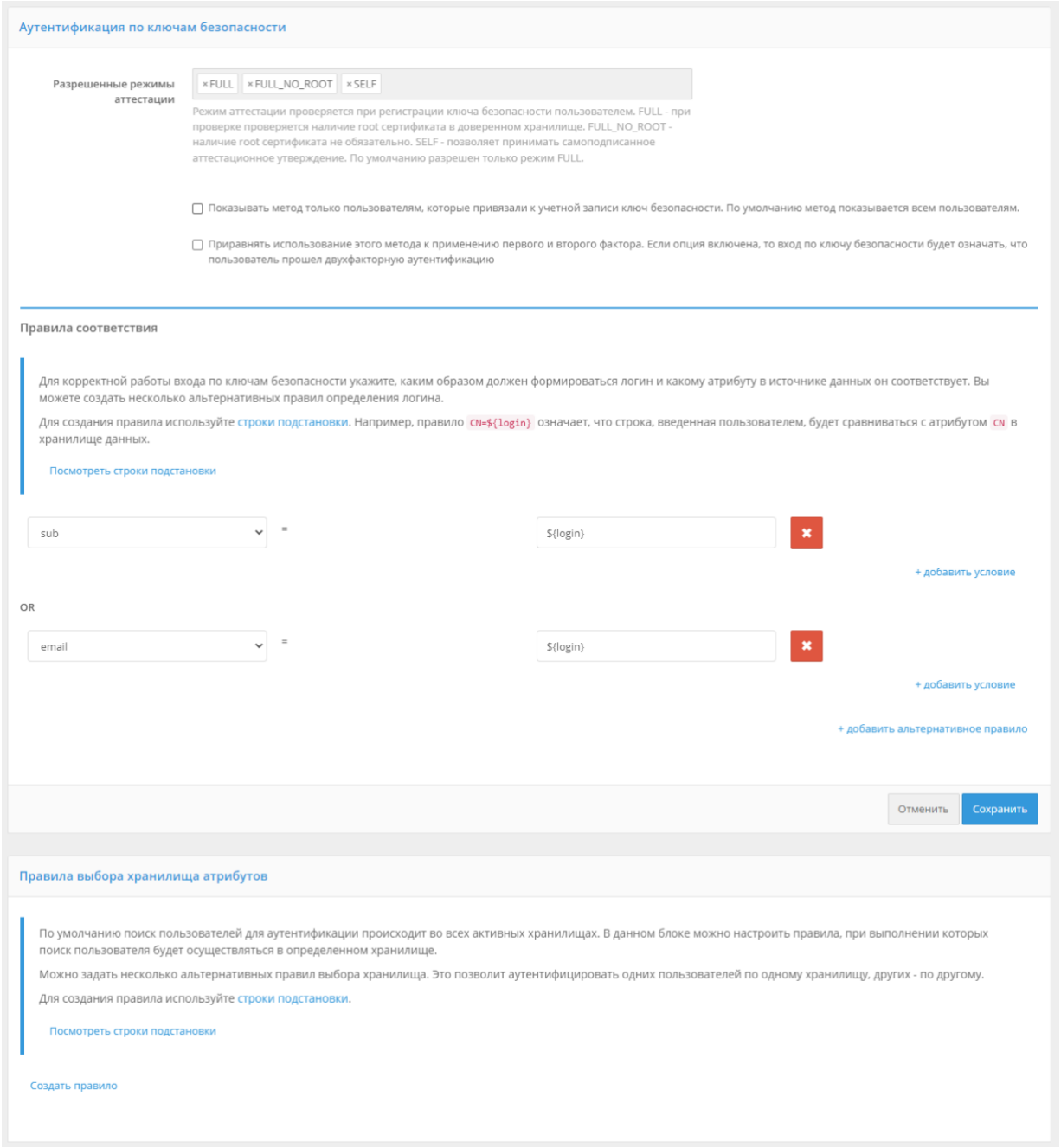

Рисунок 43 – Настройка входа с помощью ключей безопасности

4.14. 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.

Частным случаем использования метода является вход пользователя по номеру мобильного

телефона, автоматически определенного по его IP-адресу Заказчиком-оператором сотовой связи.

Автоматическая идентификация возможна только для первого фактора.

Для использования данного метода аутентификации необходимо выполнить следующие действия:

1. Создайте процедуру входа (см. п. 6), выполняемую до прохождения первого фактора аутентификации, которая запрашивает свойства сессии от сервиса Заказчика.

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

   Например, в частном случае при входе по автоматически определенному номеру телефона процедура выполняет следующие действия:

   – 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.
   – После получения номера телефона процедура запрашивает у Blitz Identity Provider вход методом автоматической идентификации.
2. Внесите изменения в файл /usr/share/identityblitz/blitz-config/blitz.conf согласно п. 16.1.32.
3. Сконфигурируйте метод в консоли управления.
4. Если вы используете несколько методов автоматической идентификации, модифицируйте тексты интерфейсов для каждого из них согласно п. 16.2.3.

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 (см. Рисунок 44).

   Вы можете добавить несколько условий для поиска среди атрибутов, которые

должны выполниться одновременно, чтобы пользователь был идентифицирован, а также ввести альтернативное правило.

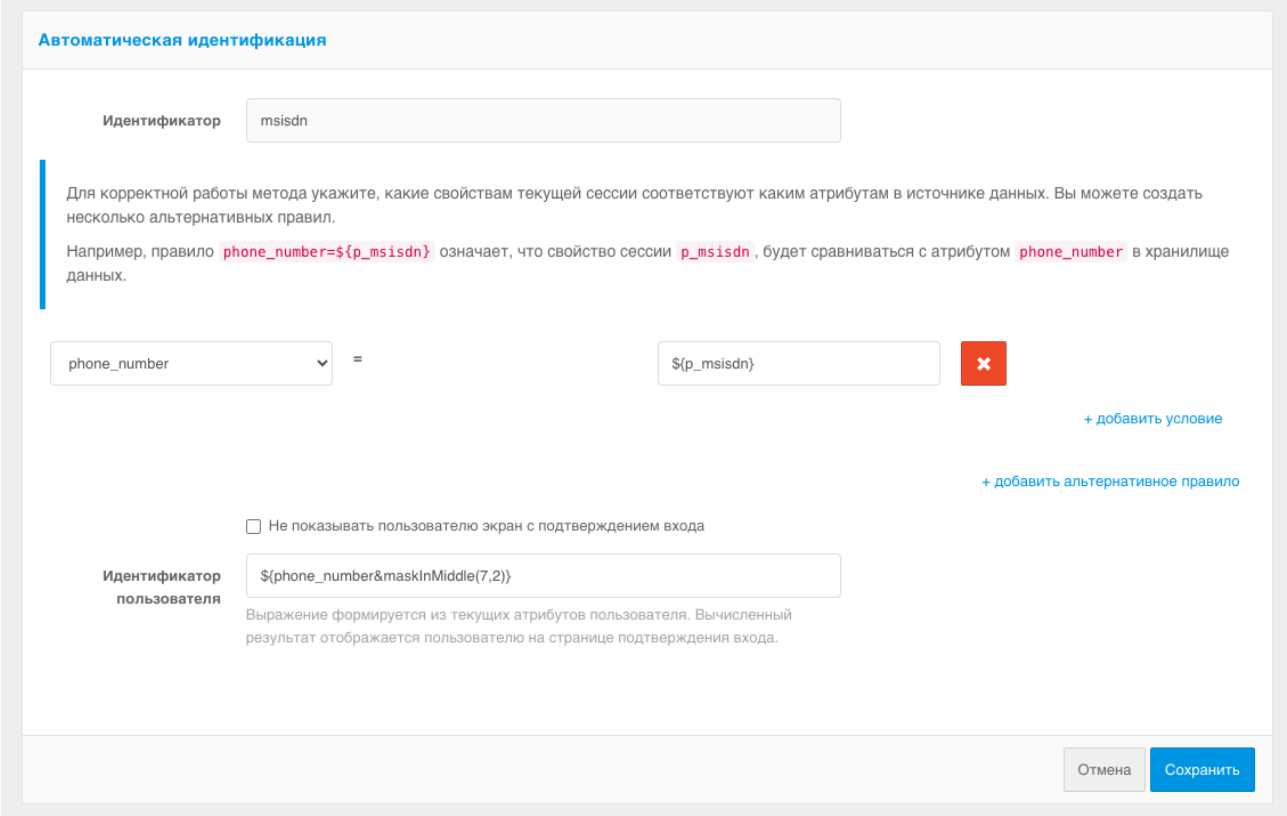

Рисунок 44 – Конфигурирование метода автоматической идентификации

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

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

4. Нажмите **Сохранить**.

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

   Для создания правила используйте следующие компоненты:

   – флажок **not**: признак инвертирования условия;

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

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

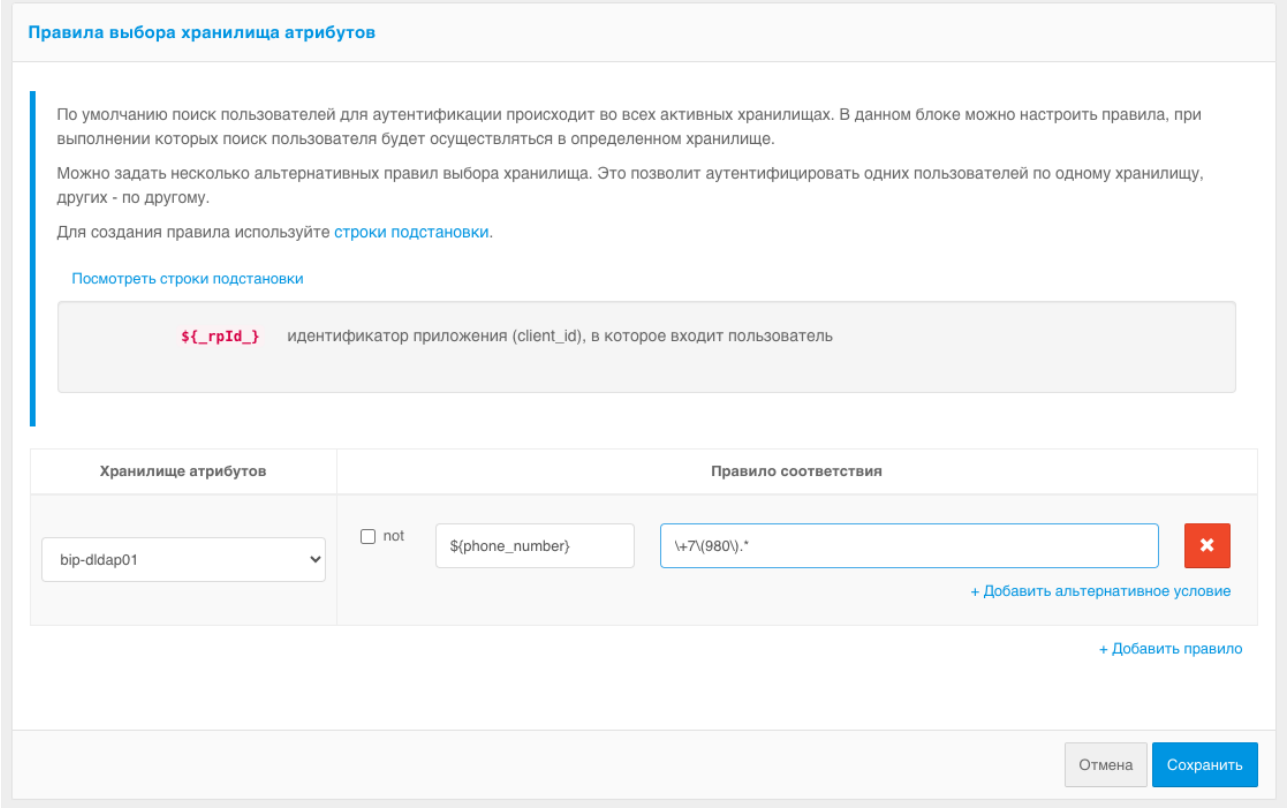

Рисунок 45 - Правило выбора хранилища атрибутов

6. Нажмите **Сохранить**.

4.15. 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 RFC 4226 «HOTP: An HMAC-Based One-Time Password Algorithm»[21].

Для использования HOTP необходимо:

– настроить и включить метод аутентификации (см. Рисунок 46);

[21] См.: https://tools.ietf.org/html/rfc4226

- 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» (см. п. 7.3).

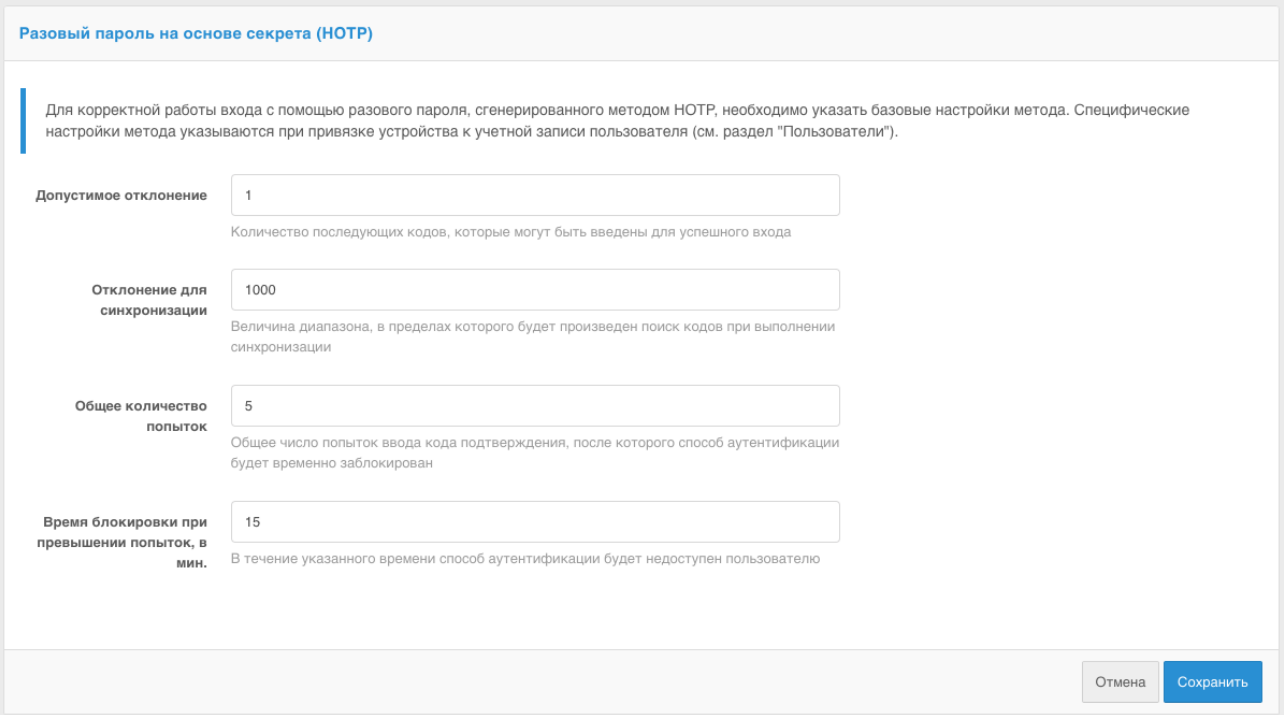

Рисунок 46 – Настройки HOTP-аутентификации

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

- максимальное допустимое отклонение при проверке кода – количество последующих кодов (например, если пользователь случайно нажал кнопку генерирования нового пароля и не использовал его в процессе аутентификации), при котором аутентификация пройдет успешно. При этом при вводе пользователем правильного кода Blitz Identity Provider автоматически восстановит синхронизацию с устройством;
- отклонение для синхронизации – если пользователь многократно будет нажимать на устройстве кнопку выработки кода и не будет использовать код для подтверждения входа, то устройство перестанет быть синхронизированным с сервером. В этом случае

при очередном входе пользователя в Blitz Identity Provider ему на странице входа будет предложено пройти процедуру сверки устройства. Для этого пользователь введет три последовательно выработанных устройством кода подтверждения. Далее в соответствии с заданной настройкой «Отклонение для синхронизации» Blitz Identity Provider проверит, встречается ли введенная пользователем последовательность кодов, и восстановит синхронизацию с устройством в случае успеха;

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

4.16. Подтверждение входа разовым паролем 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 RFC 6238 «TOTP: Time-Based One-Time Password Algorithm»[22]. В качестве таковых могут быть:

- аппаратные брелоки (генераторы разовых паролей) на основе времени;
- мобильные приложения[23].

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

1. Допустимое отклонение при проверке кода (количество предыдущих / последующих кодов). По умолчанию оба значения равны 1: пользователь при входе может ввести как текущий код подтверждения, так и следующий или предыдущий (сгенерированный в соседних временных интервалах). Такая необходимость может возникнуть, например, для компенсации возможной незначительной рассинхронизации серверного времени и времени на TOTP-устройствах пользователей.
2. Общее количество попыток – число попыток ввода кода подтверждения, после которого данный способ подтверждения будет заблокирован.
3. Время блокировки при превышении попыток (в минутах).
4. Настройка отображения генераторов разовых паролей, которая включает в себя «Атрибут с именем пользователя» и «Название единой системы входа». Эти параметры

[22] См.: https://tools.ietf.org/html/rfc6238
[23] Наиболее известные приложения для выработки TOTP-кодов: Google Authenticator, Twilio Authy, FreeOTP Authenticator, Microsoft Authenticator, Яндекс.Ключ.

будут отображаться в мобильном приложении после привязки учетной записи пользователя.

5. Ссылки на приложения-генераторы разовых паролей. Следует указать ссылки на приложения, которые рекомендуется использовать пользователям. Эти ссылки будут предложены пользователю в веб-приложении «Личный кабинет».

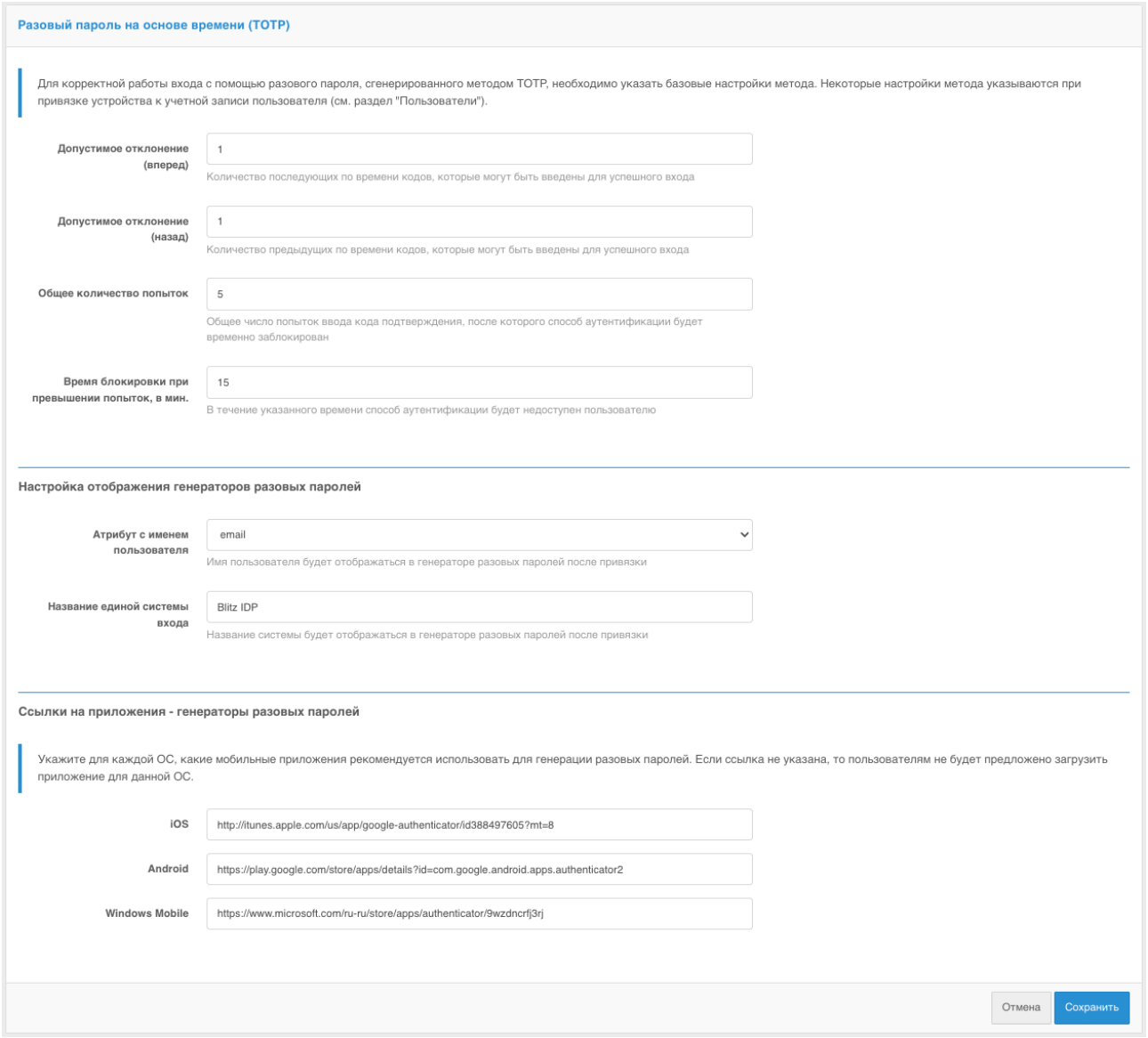

Рисунок 47 – Настройки TOTP-аутентификации

4.17. 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.

4.17.1. 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» (см. Рисунок 48) 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.

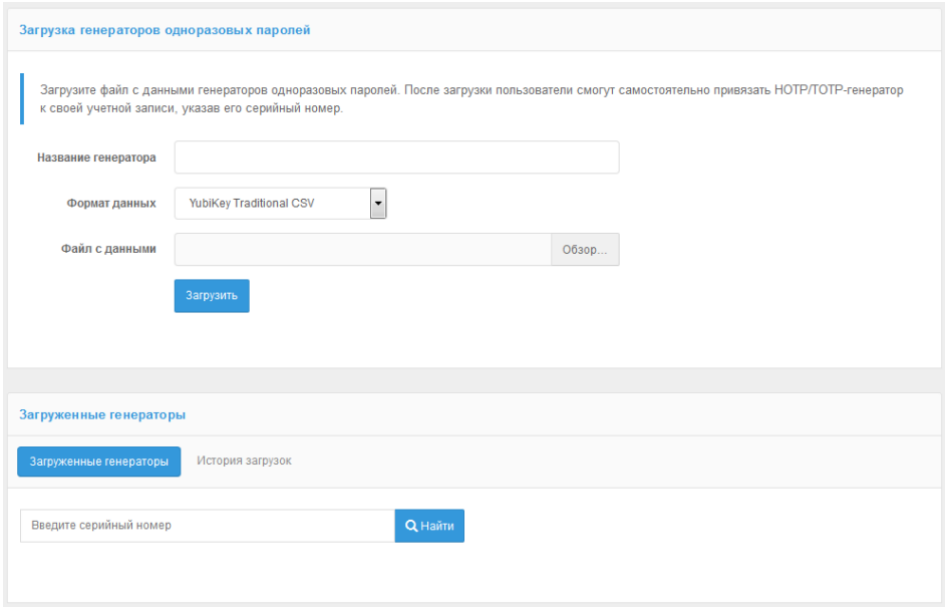

Рисунок 48 – Загрузка файлов с описаниями устройств генерации кодов

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

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

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

Пояснения по значениям параметров в файле:

- serial – серийный номер устройства.
- Seed – ключ устройства в шестнадцатеричном (hex) формате[24].

[24] 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 (меню «Пользователи», см. п. 9.3.5 документа);
- 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 код подтверждения.

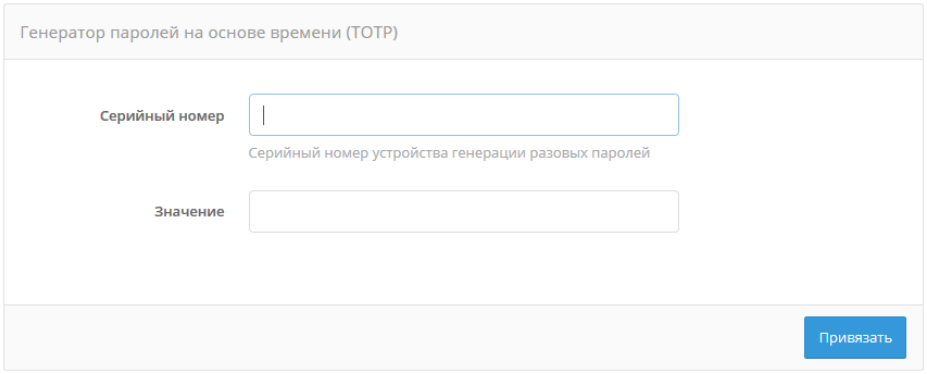

Рисунок 49 – Привязка аппаратного TOTP-генератора

4.17.2. Привязка мобильного приложения

Для привязки мобильного приложения следует:

- перейти к учетной записи пользователя, которому необходимо привязать мобильное приложение (меню «Пользователи», см. п. 9.3.5 документа);
- 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».

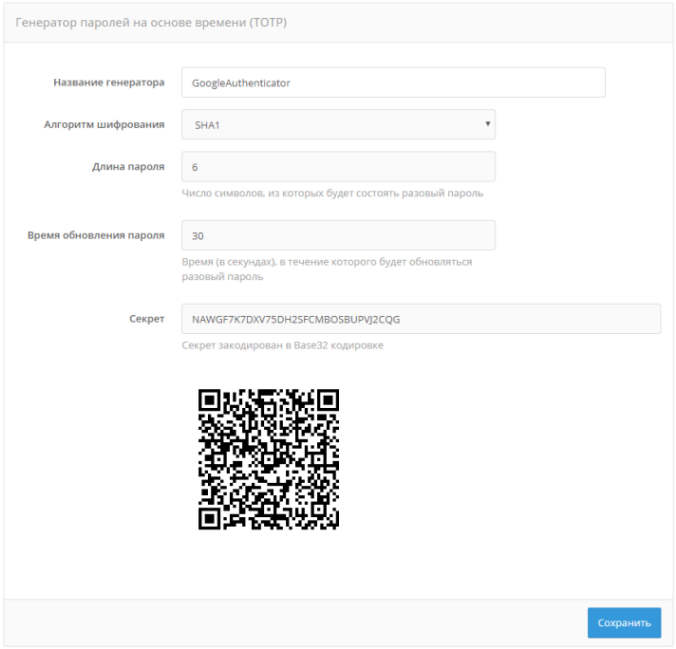

Рисунок 50 – Привязка мобильного приложения, генерирующего TOTP-коды

4.18. Коды подтверждения, отправляемые в SMS и push-уведомлениях

Можно использовать отправляемые в мобильное приложение push-уведомления или SMS-сообщения для подтверждения входа (второго фактора аутентификации).

Для этого необходимо:

- настроить и включить метод аутентификации «Подтверждение с помощью кода». Необходимо задать:
  - 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 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};
- настроить подключение Blitz Identity Provider к SMS-шлюзу и сервису отправки push-уведомления (см. п. 13.1).

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

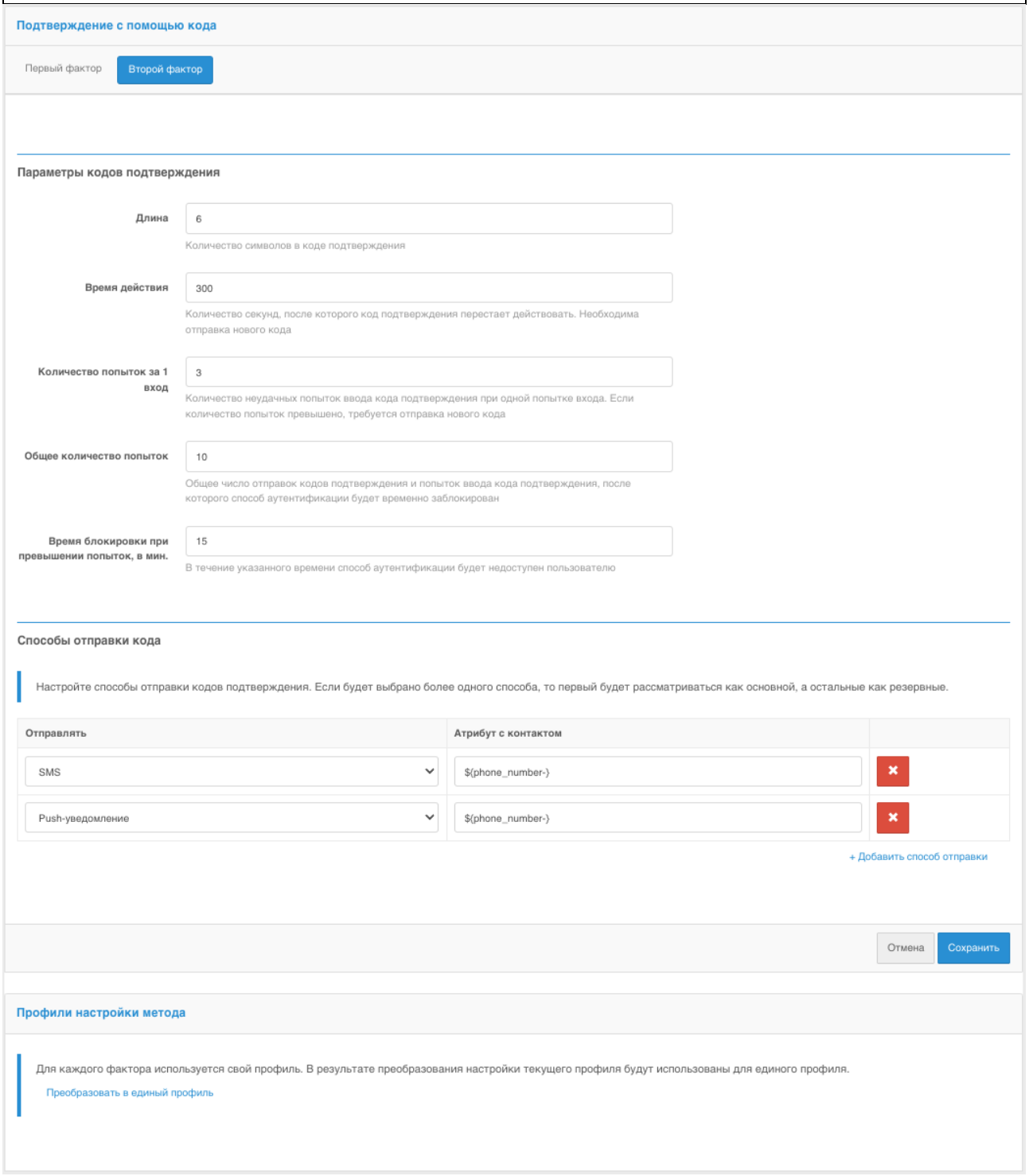

Рисунок 51 – Настройки кодов подтверждения для двухфакторной аутентификации

По умолчанию используются единые настройки для кодов подтверждения, отправляемых для проверки первого и второго фактора (см. п. 4.8.6). Для разделения настроек необходимо перейти по ссылке «Сконфигурировать профиль для каждого фактора» в блоке «Профили настройки метода». Тогда настройки будут разведены и можно будет переключаться между первым и вторым фактором.

При необходимости перейти к единым настройкам следует перейти по ссылке «Преобразовать в единый профиль» в блоке «Профили настройки метода».

4.19. Коды подтверждения, отправляемые по электронной почте

Можно использовать отправляемые по электронной почте коды подтверждения для подтверждения входа (второго фактора аутентификации).

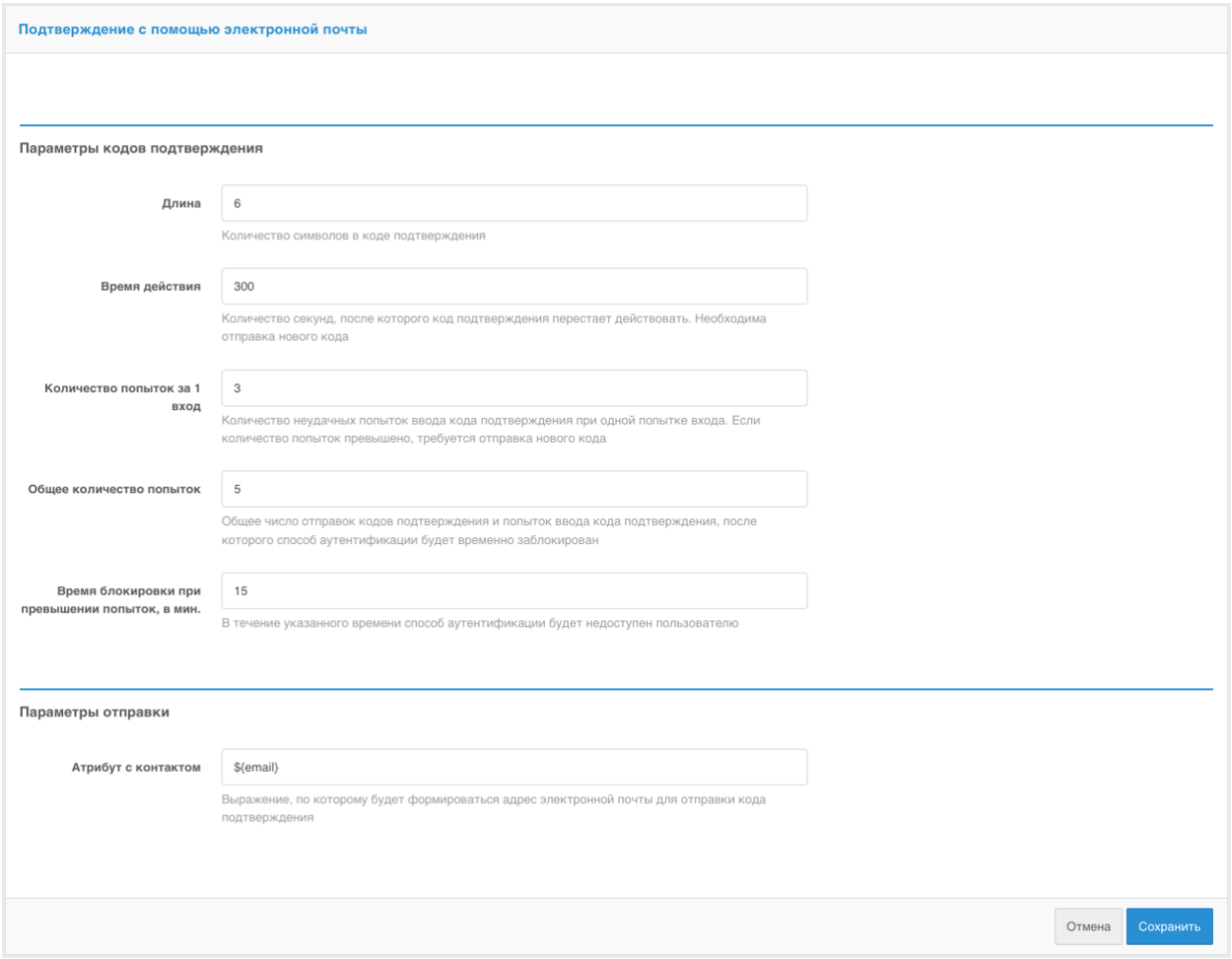

Рисунок 52 – Настройки кодов подтверждения, отправляемых по электронной почте

Для этого необходим:

- настроить и включить этот метод аутентификации. 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 способ отправки: указать атрибут, в которых сохранен адрес электронной почты пользователя, например, ${email};
- настроить подключение Blitz Identity Provider к SMTP-сервису (см. п. 13.3).

Следует помнить, что если у пользователя не задан адрес электронной почты, то он не сможет использовать метод подтверждения входа с помощью кодов, отправляемых на электронную почту.

4.20. Подтверждение входа с помощью Duo Mobile

Можно использовать мобильное приложение Duo Mobile[25] (компания Cisco) для подтверждения входа (второго фактора аутентификации).

Для этого необходимо выполнить настойки на стороне сервиса Duo Security:

- зарегистрировать учетную запись на сайте Duo[26];
- войти в панель администратора[27] 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 этого нужно провести настройки в консоли управления Blitz Identity Provider:
- сконфигурировать метод аутентификации «Duo push-аутентификация» (см. Рисунок 53). 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».

[25] См.: https://duo.com/product/multi-factor-authentication-mfa/duo-mobile-app
[26] См.: https://signup.duo.com/
[27] См.: https://admin.duosecurity.com/

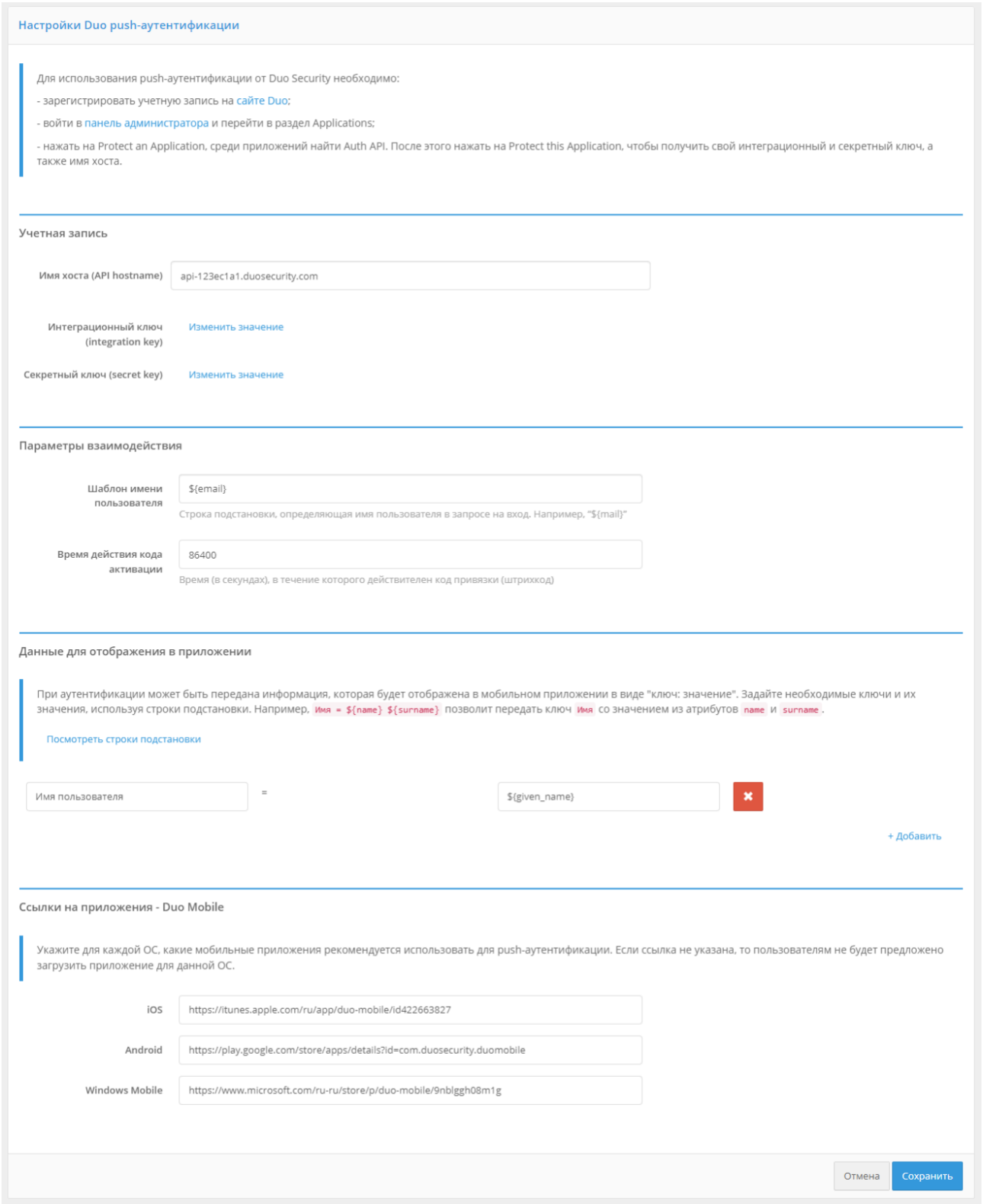

Рисунок 53 – Настройки Duo push-аутентификации

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.

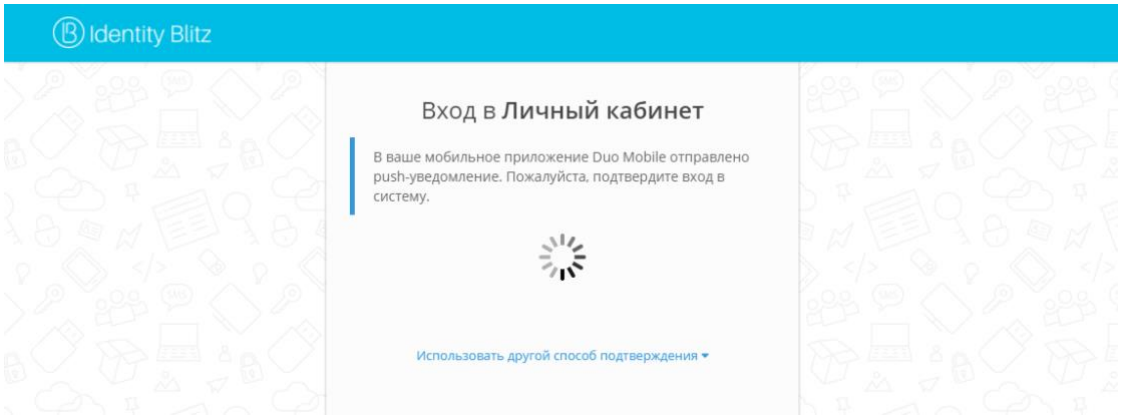

Рисунок 54 – Инициирование push-аутентификации

Рисунок 55 – Запрос push-аутентификации на смартфоне (приложение Duo Mobile)

4.21. 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 (см. Рисунок 56).

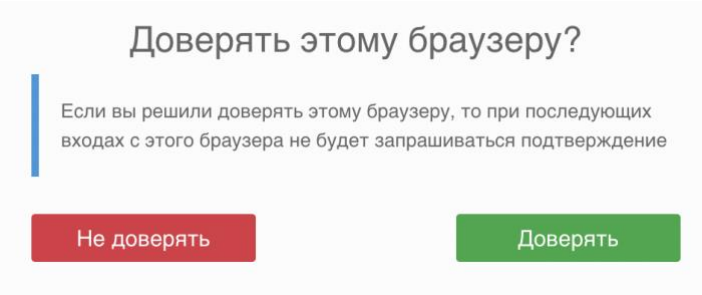

Рисунок 56 – Запрос, доверяет ли пользователь браузеру

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».

4.22. Подтверждение с помощью ключей безопасности WebAuthn, Passkey, FIDO2, U2F

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

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

- Touch ID или Face ID на мобильном телефоне iOS или проверки отпечатка пальца или распознавания лица в Android.

Для использования ключей безопасности необходимо задать настройки ключей безопасности (см. п. 4.2), а также сконфигурировать метод аутентификации в консоли управления (см. Рисунок 57).

В консоли управления необходимо задать настройку «Разрешенные режимы аттестации».

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

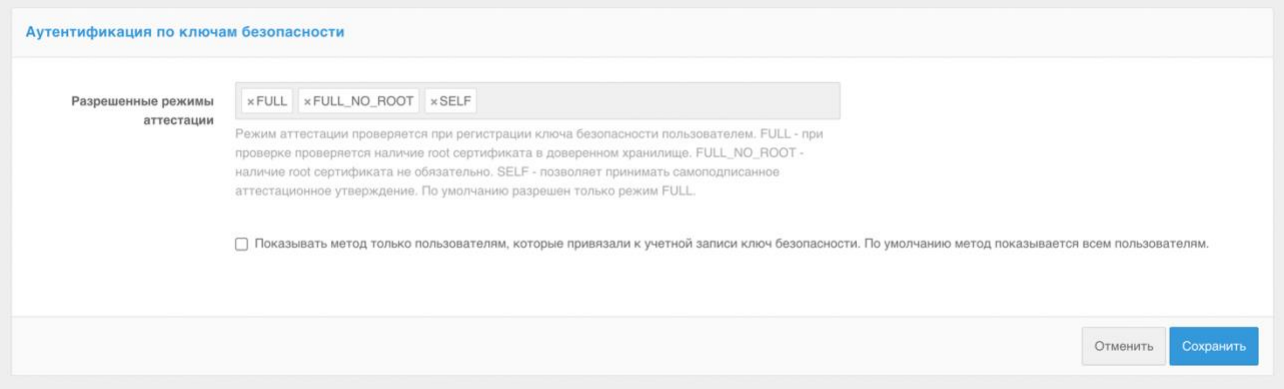

Рисунок 57 – Настройка использования ключей безопасности для подтверждения входа

4.23. 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 способа аутентификации нужно выполнить настройки в конфигурационном файле (см. п. 16.1.31), а также сконфигурировать метод аутентификации в консоли управления (см. Рисунок 58).

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 контрольных вопросов. Создание и редактирование вопросов выполняется через конфигурационные файлы.

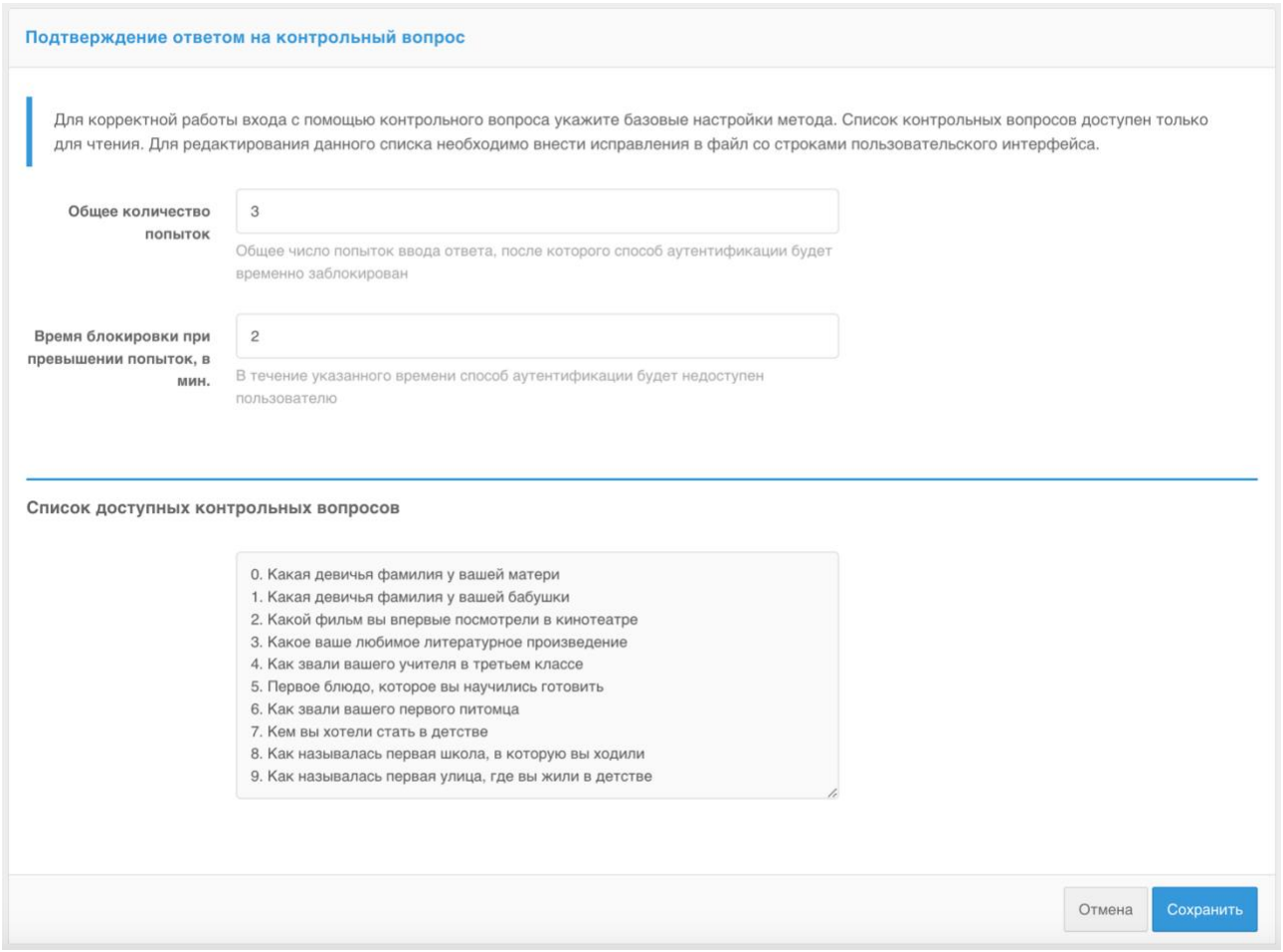

Рисунок 58 – Пример настройки метода подтверждения ответом на контрольный вопрос

4.24. 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 следующие действия:

1) Добавьте метод в файле blitz.conf.
2) Сконфигурируйте метод в консоли управления.

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,
                "method" : "flashCall"
            },
            …
        ]
    ],
    …
}
```

3) Перезапустите сервисы Blitz Identity Provider.

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

Настройка метода в консоли

В консоли управления выполните следующие действия:

1) На вкладке **Подтверждение по телефонному звонку** задайте следующие настройки:

– *Длина кода*: количество последних цифр номера входящего звонка, которые будут использоваться в качестве кода на втором факторе аутентификации.

– *Время действия*: количество секунд, после которого код подтверждения перестает действовать и необходим повторный звонок.

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

– *Общее количество попыток*: общее число отправок кодов подтверждения и попыток ввода кода подтверждения, после которого данный способ аутентификации будет временно заблокирован.

– *Время блокировки при превышении общего количества попыток, в минутах*: в течение указанного времени данный способ аутентификации будет недоступен пользователю.

– *Название атрибута с мобильном номером пользователя*: выберите из списка атрибут, в котором хранится номер телефона пользователя для совершения звонка.

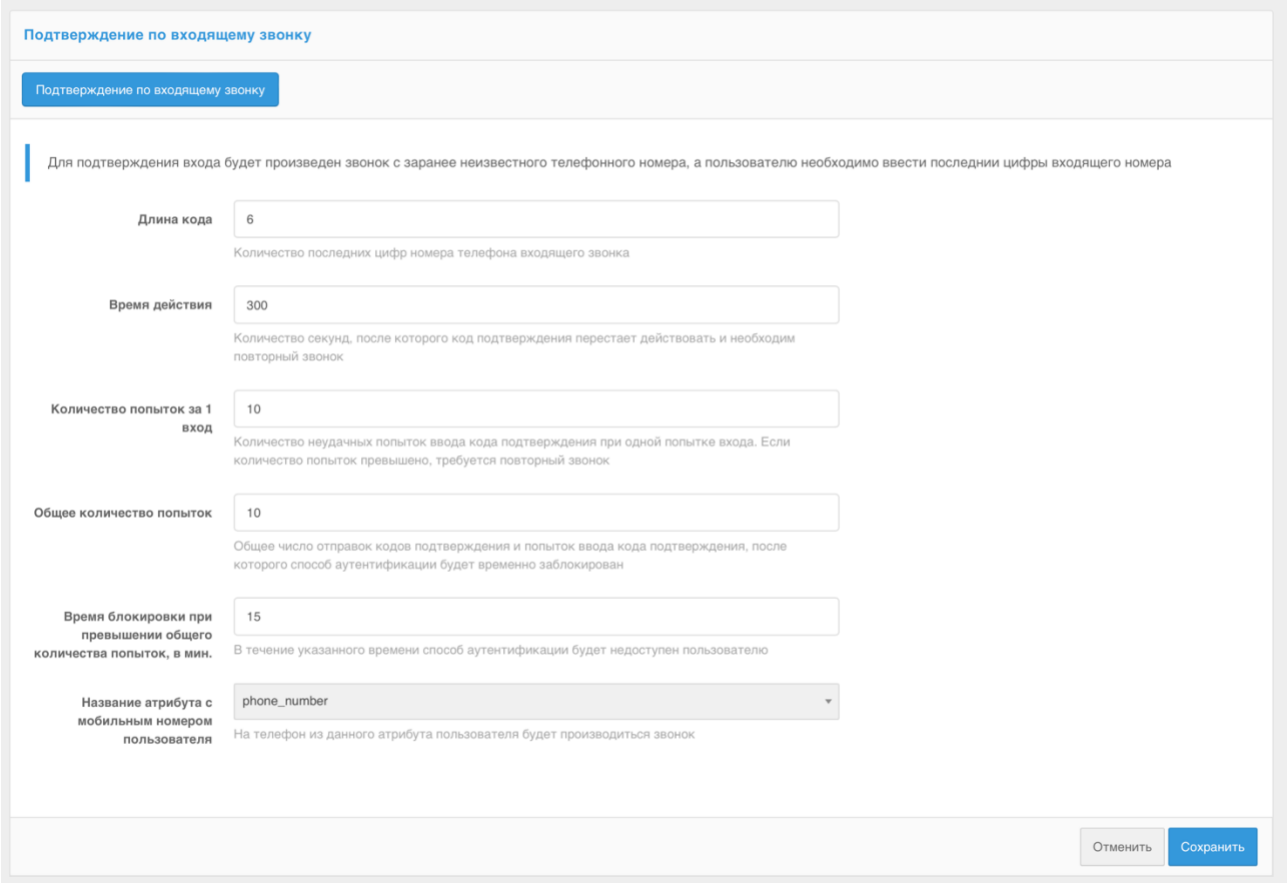

Рисунок 59 – Пример настройки метода подтверждения по входящему звонку

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

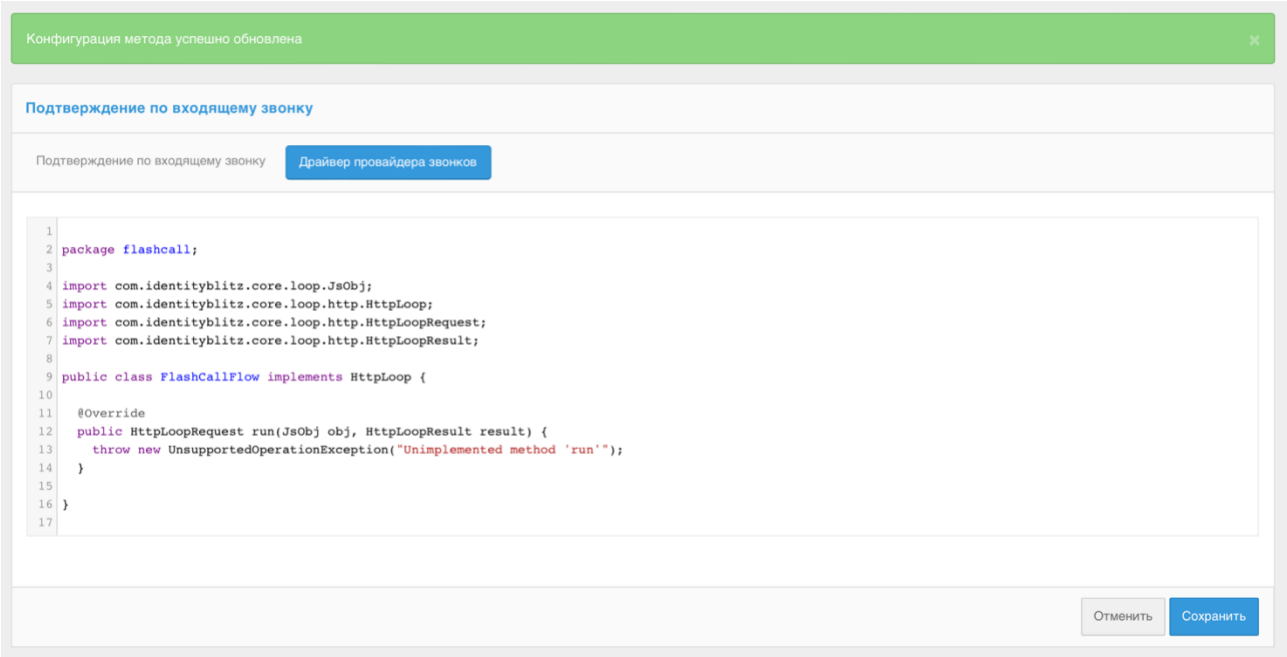

Рисунок 60 – Настройка драйвера провайдера звонков по умолчанию

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. Пример процедуры для интеграции с провайдером Flash Call:

```
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 {
                        return HttpLoop.error("wrong http status",
                                        Collections.<String, String>singletonMap("status", "" +
result.status()));
                }
        }
}
```

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

4.25. 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.

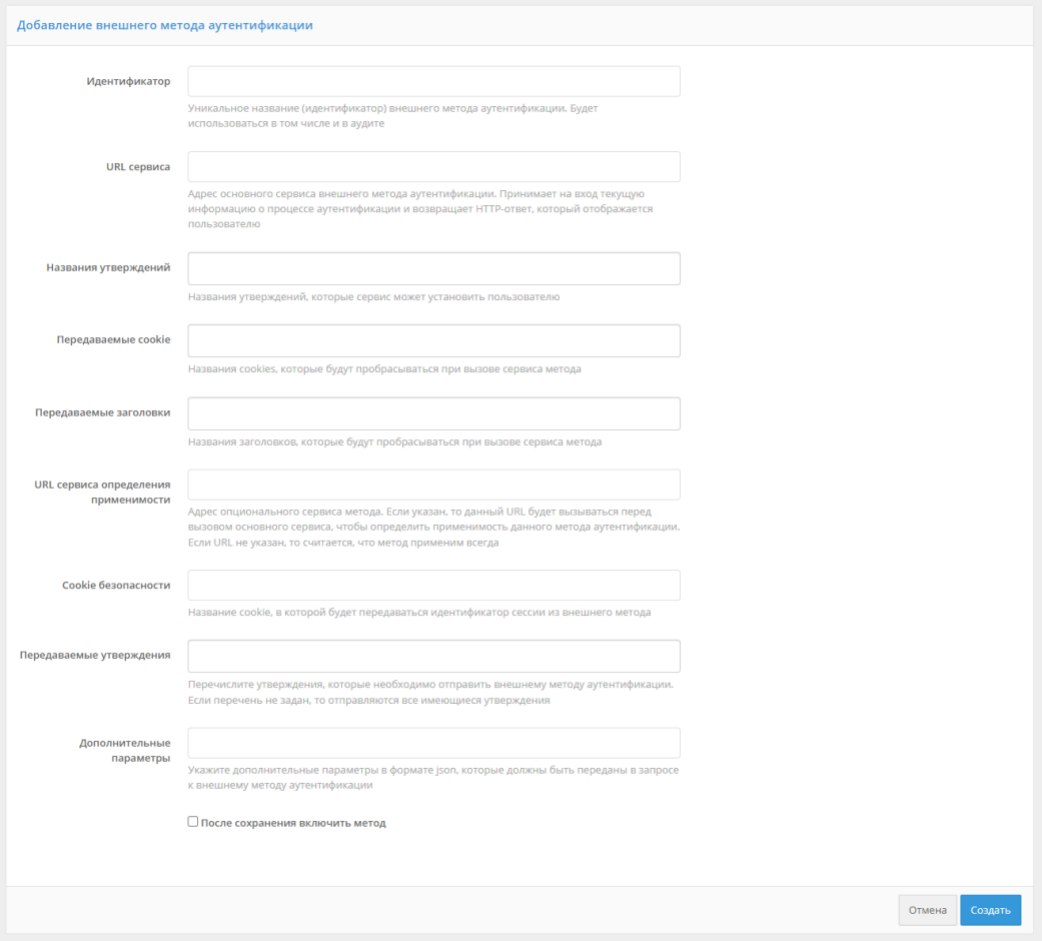

Рисунок 61 – Пример настройки внешнего метода

Для настройки использования Blitz Identity Provider внешнего метода 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 «Руководству по интеграции».

4.26. 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».

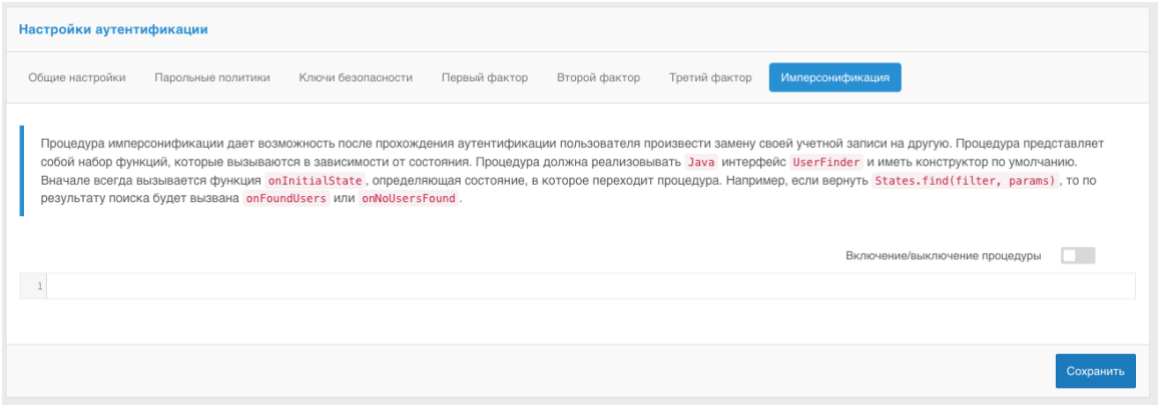

Рисунок 62 – Пример настройки процедуры имперсонификации

5. Регистрация приложений и сетевых служб

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 (см. Рисунок 63).

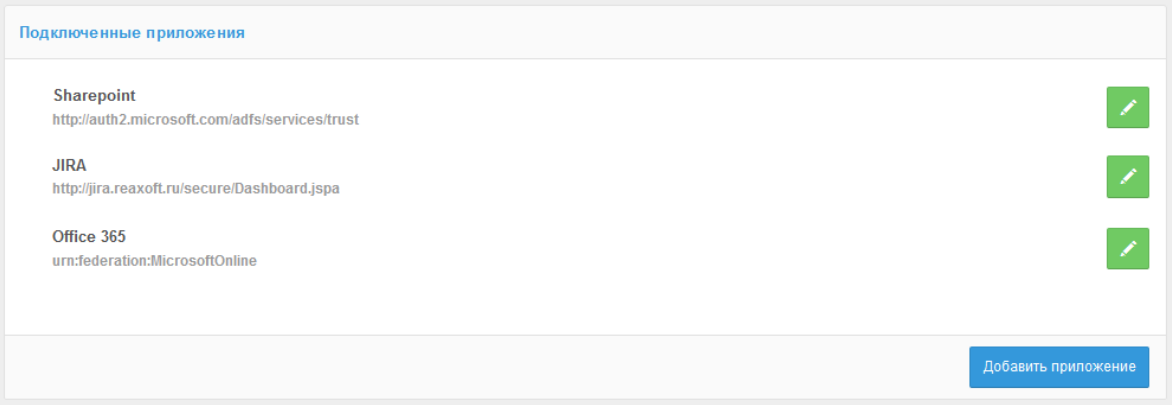

Рисунок 63 – Главный экран настройки приложений

5.1. 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, при задании идентификатора для OAuth 2.0 **недопустимо** использовать двоеточие и тильду), его название и домен, т.е. URL, по которому доступно данное приложение (см. Рисунок 64).

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.

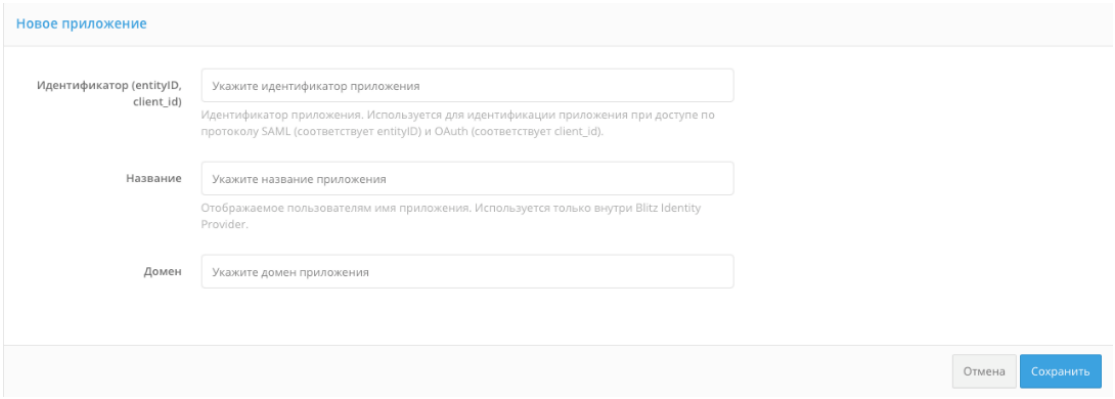

Рисунок 64 – Базовые настройки приложения

Шаг 2. Задание стартовой страницы приложения и выбор шаблона страницы входа (см. Рисунок 65). В поле «Стартовая страница приложения» рекомендуется задать ссылку на вход в приложение, инициирующую запрос идентификации и аутентификации.

В списке «Шаблон страниц» необходимо выбрать, на основе какого шаблона должна отображаться страница входа при попытке доступа пользователя в данное приложение. Инструкция по созданию нового шаблона входа приведена в п. 14.

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.

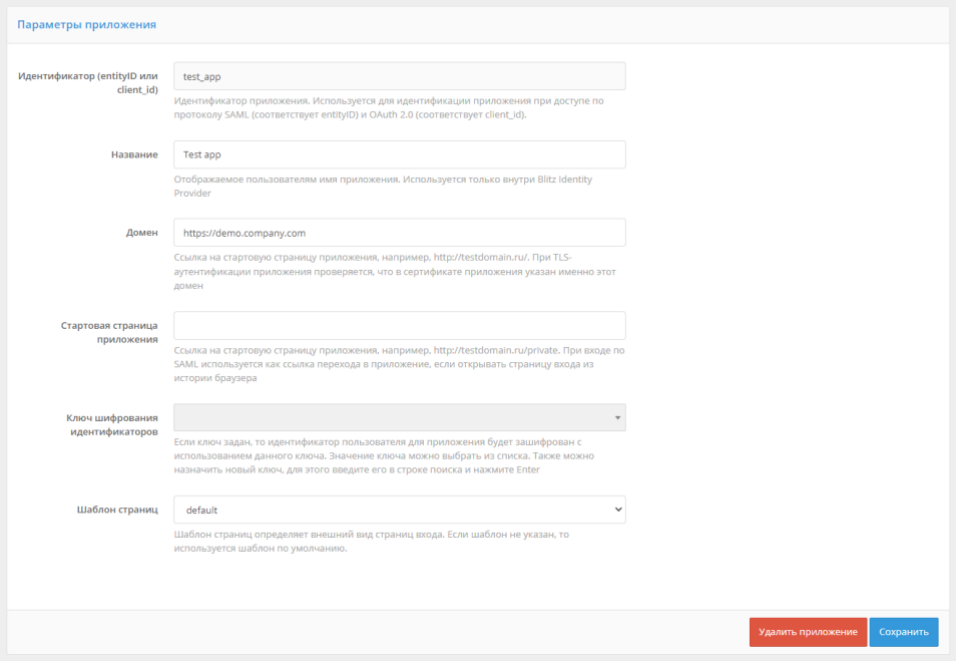

Рисунок 65 – Выбор шаблона страницы входа и настройка логаута

Шаг 3. Настройка правил доступа в приложения (см. Рисунок 66). Можно настроить правила, на основе которых Blitz Identity Provider будет принимать решение, впускать или нет пользователя в приложение.

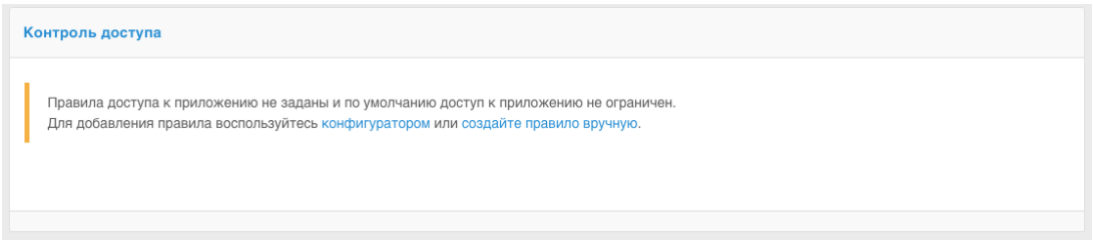

Рисунок 66 – Настройка правил контроля доступа

Правила контроля доступа можно добавить с помощью конфигуратора (см. Рисунок 67) или вручную с помощью RQL-выражения (см. Рисунок 68). 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).

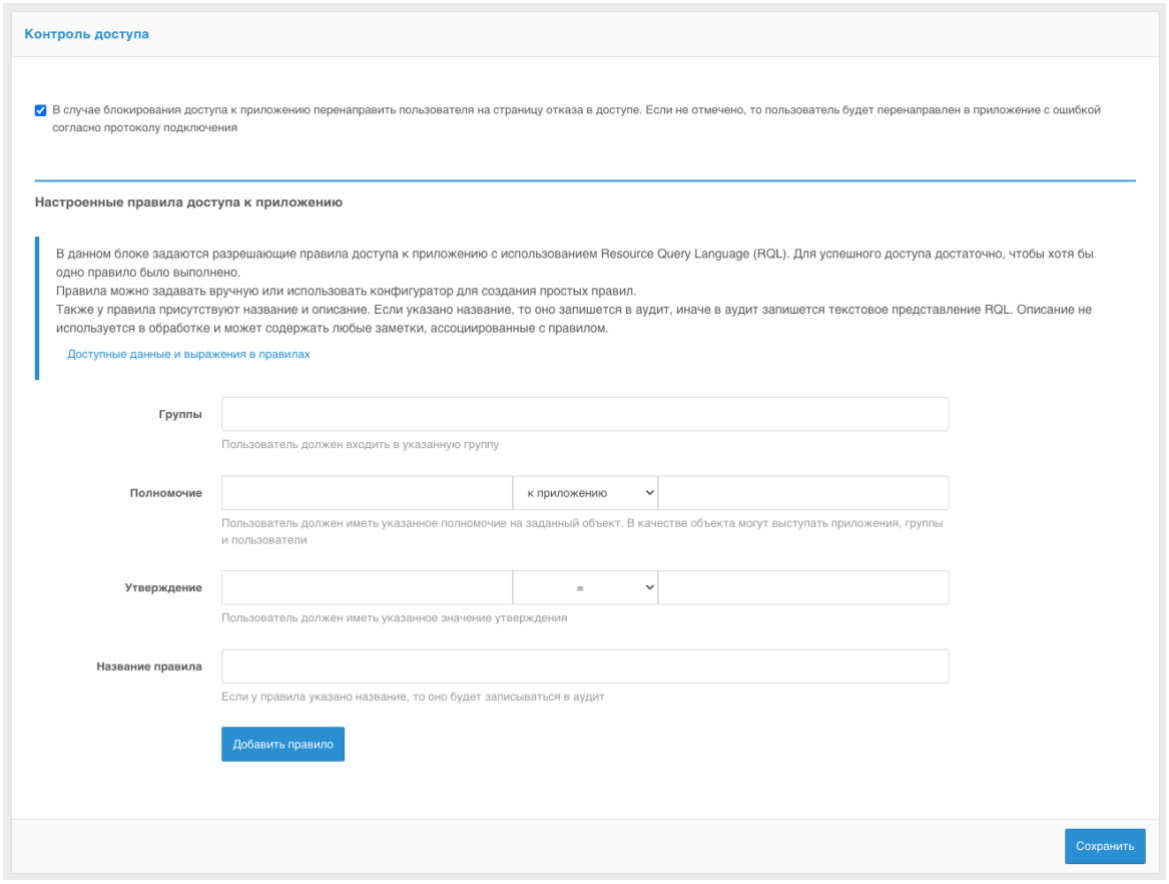

Рисунок 67 – Добавление правила доступа с помощью конфигуратора

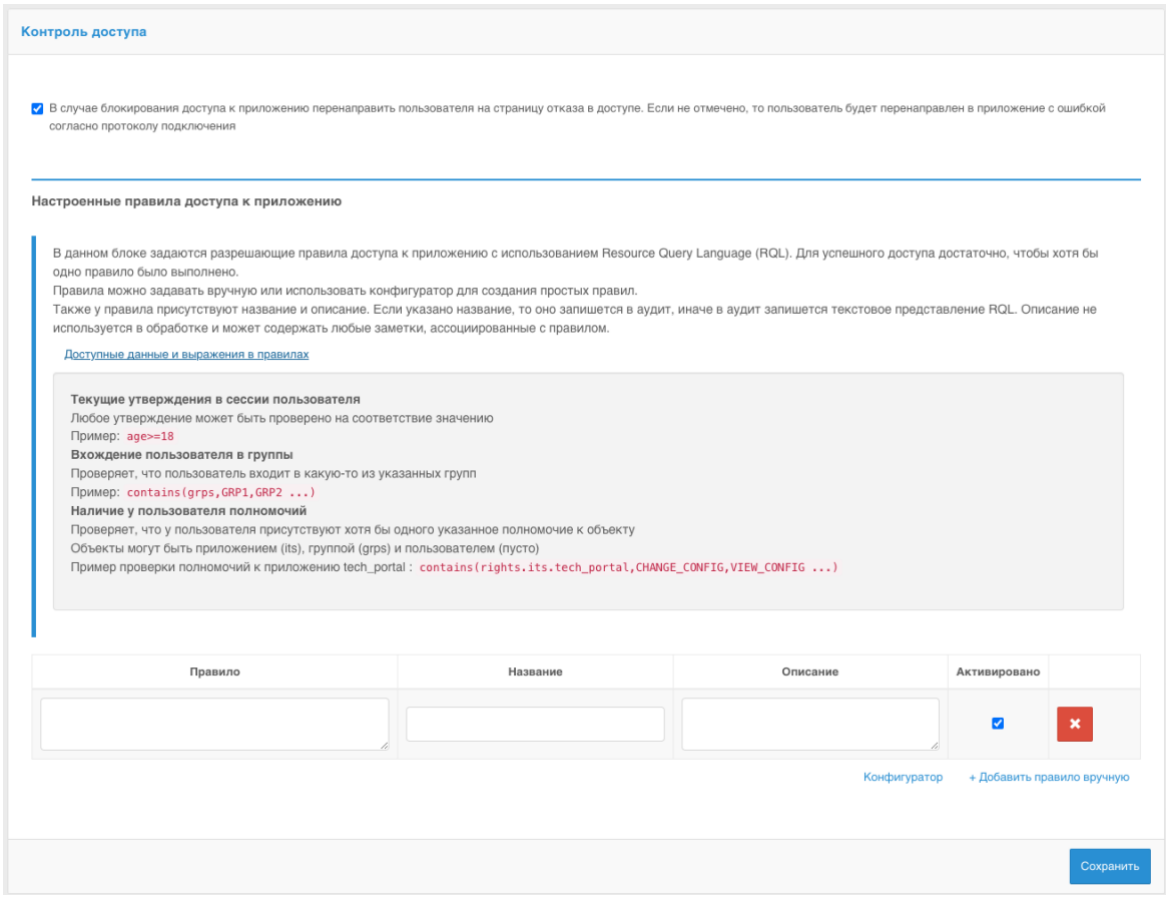

Рисунок 68 – Добавление правила доступа вручную

Шаг 4. Настройки протоколов подключения (см. Рисунок 69). Необходимо настроить один или несколько протоколов подключения приложения к Blitz Identity Provider.

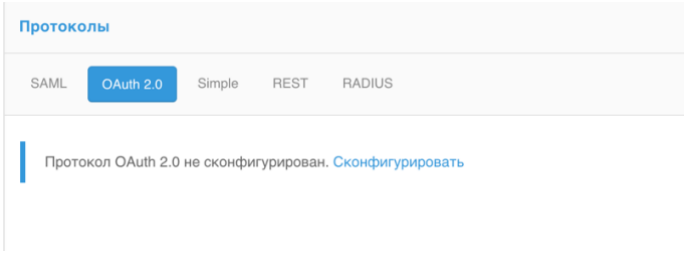

Рисунок 69 – Настройка протоколов подключения

Поддерживаются следующие протоколы подключения:

- SAML – для подключения приложений по SAML 1.0, 1.1, 2.0 и WS-Federation для идентификации и аутентификации пользователей;
- OAuth 2.0 – для подключения приложений по OAuth 2.0, OpenID Connect 1.0 (OIDC) для идентификации и аутентификации пользователей. В рамках этого протокола возможно конфигурирование динамической регистрации клиентов;
- Simple – для подключения веб-приложений для осуществления идентификации и аутентификации с помощью подстановки в приложение логина и пароля с proxy-сервера, если приложение не поддерживает возможности подключения по SAML/OIDC;
- REST – для подключения приложений, использующих REST-сервисы Blitz Identity Provider по регистрации/изменению учетных записей, управлению устройствами аутентификации пользователей.
- RADIUS – для подключения к сетевым службам по протоколу RADIUS.

Если организация планирует разработку или доработку собственных приложений для подключения их к Blitz Identity Provider, то разработчикам необходимо ознакомиться с документом «Руководство 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.

5.2. Настройка SAML и WS-Federation

5.2.1. Подключение по SAML 1.0/1.1/2.0

При подключении приложения по SAML необходимо задать следующие настройки (см. Рисунок 70):

- загрузить SAML-метаданные подключаемого приложения;
- 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 в разделе «SAML» консоли управления (см. п. 5.2.3).

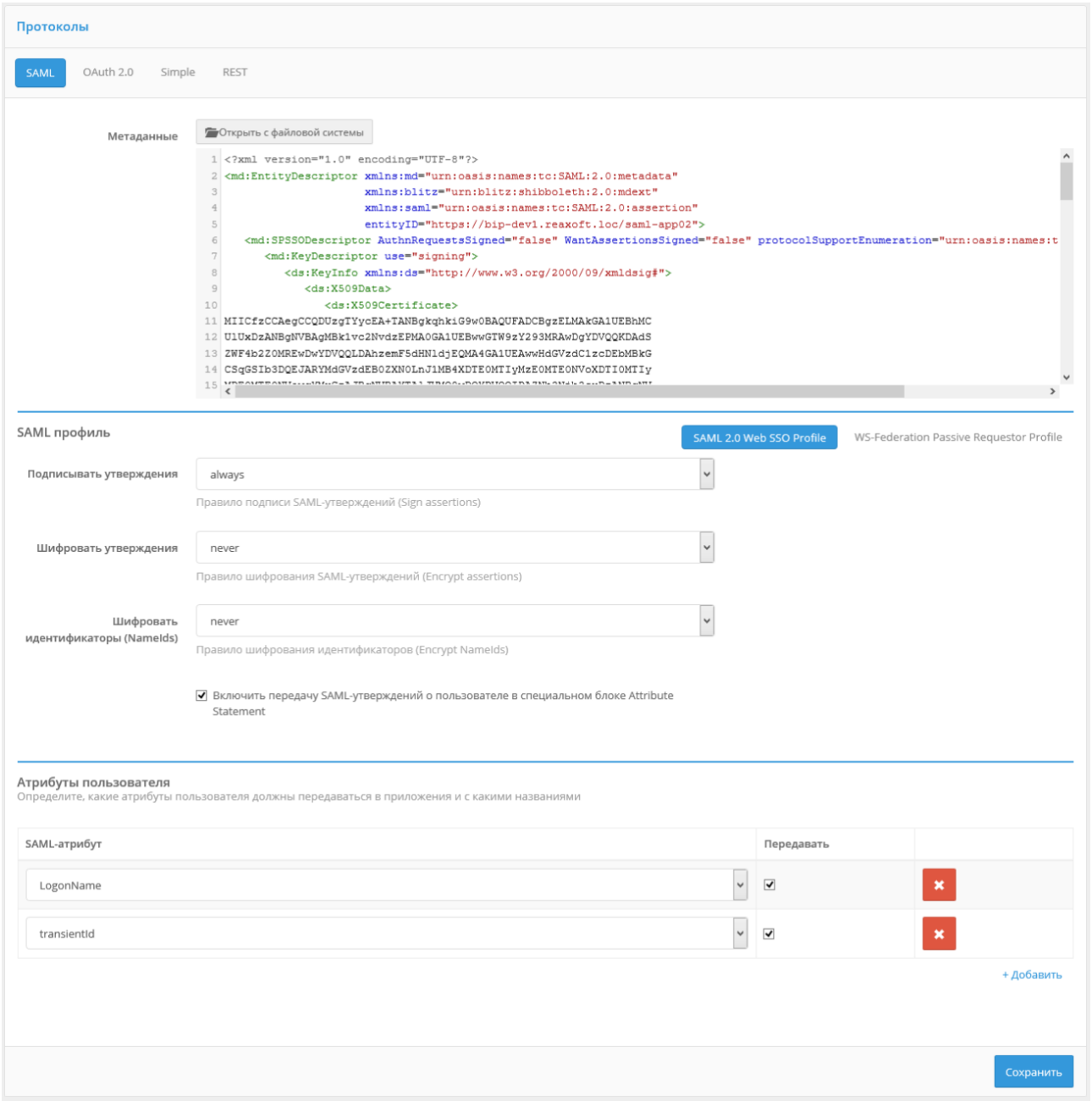

Рисунок 70 – Настройки протокола SAML для приложения

5.2.2. Подключение по WS-Federation

При подключении приложения по WS-Federation необходимо задать следующие настройки (см. Рисунок 71):

- загрузить метаданные подключаемого приложения;
- 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[29], 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 в разделе «SAML» консоли управления (см. п. 5.2.3).

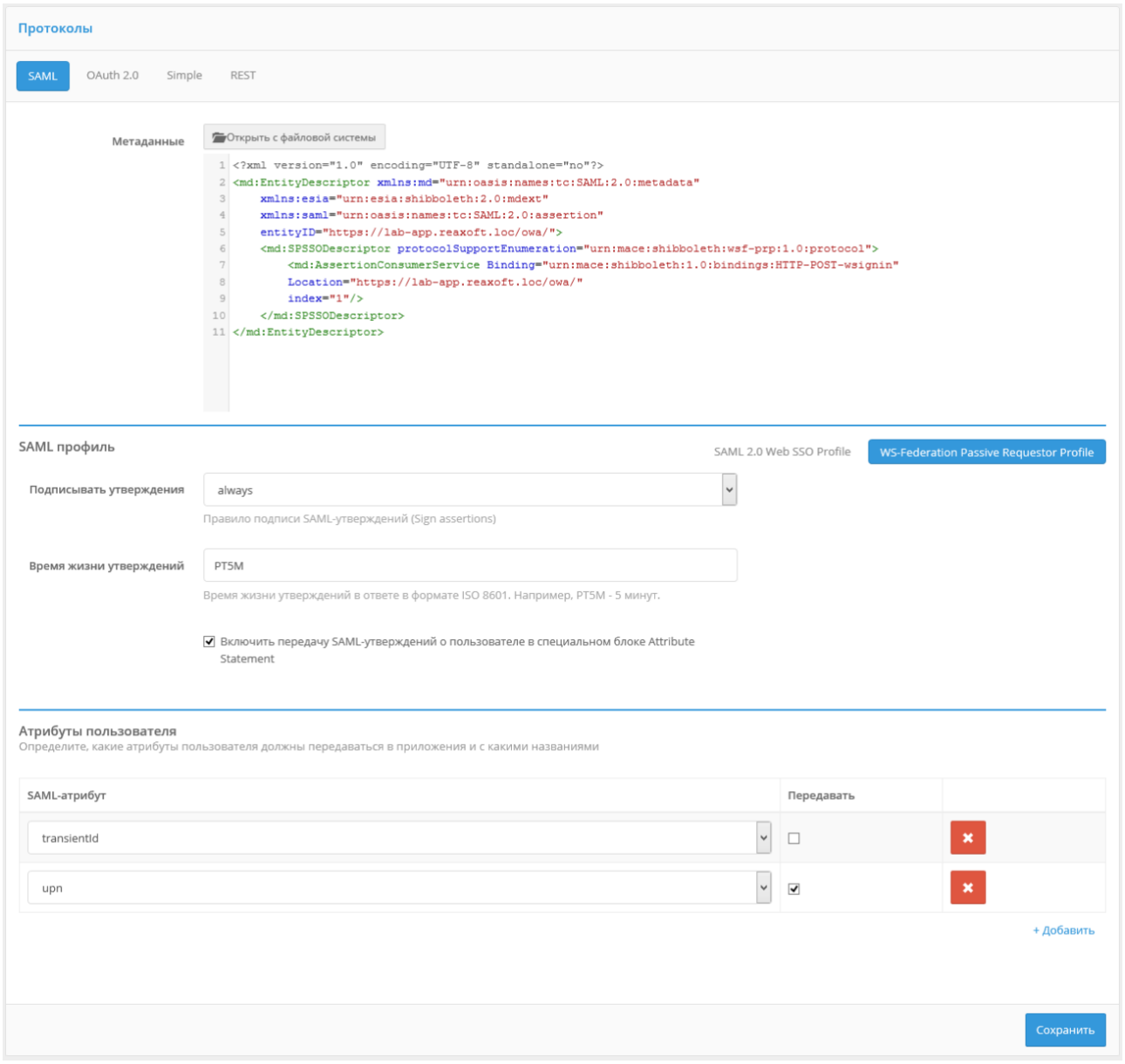

Рисунок 71 – Настройки протокола WS-Federation для приложения

[29] См.: http://www.ifap.ru/library/gost/86012001.pdf

5.2.3. Настройка SAML-атрибутов

Для регистрации SAML-атрибутов пользователя в Blitz Identity Provider используется раздел «SAML» консоли управления (см. Рисунок 72).

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 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).

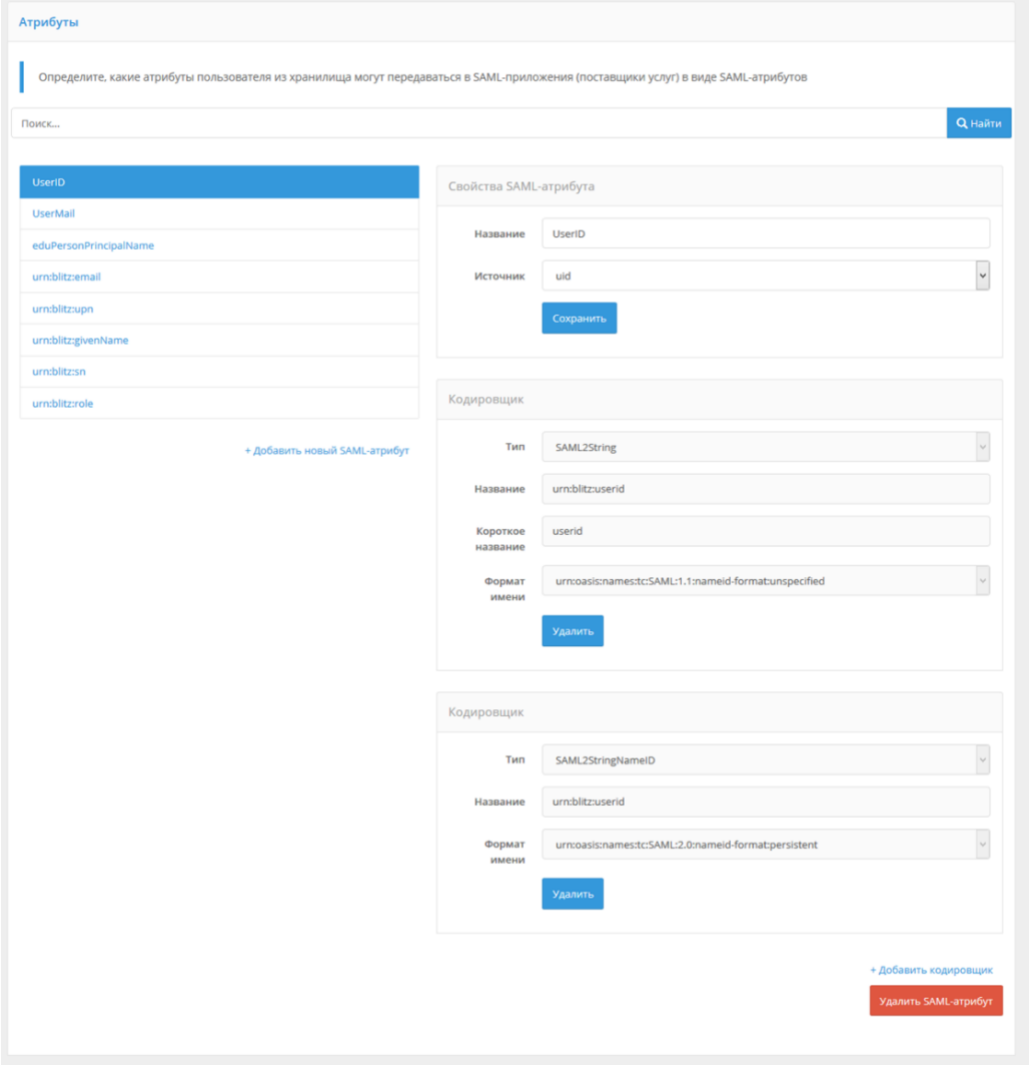

Рисунок 72 – Создание SAML-атрибутов

5.3. Настройка OAuth 2.0 и OpenID Connect 1.0

5.3.1. 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 (см. Рисунок 73):

- указать секретный ключ (или использовать сгенерированный по умолчанию ключ) подключаемого приложения (client_secret), который должен использоваться подключенным приложением при обращении к Blitz Identity Provider (если не указан, то аутентификация приложения-клиента должна производиться иначе, например, с использованием proxy TLS);
- указать дополнительный секретный ключ (client_secret) подключаемого приложения. Рекомендуется для случаев, когда нужно обеспечить плавную смену client_secret для данного приложения;

- указать предопределенную ссылку возврата (redirect_uri) – URL, на который по умолчанию будет переадресован пользователь после прохождения авторизации (redirect_uri);
- указать допустимые префиксы ссылок возврата – префикс используется для проверки ссылок возврата (redirect_uri), переданных в запросах на идентификацию от приложений. Если в запросе на аутентификацию указана ссылка возврата, и она не соответствует ни одному из указанных префиксов, то в аутентификации будет отказано;
- допустимые разрешения – разрешения (scope), которые имеет право запрашивать данное приложение;
- разрешения по умолчанию – разрешения (scope), которые будут по умолчанию выданы приложению после аутентификации. Если не указаны, то в запросе на аутентификацию всегда должны быть явно прописаны требуемые разрешения;
- отметить при необходимости опцию «Не требовать от пользователя согласие на предоставление доступа к данным о себе». Если она отмечена, то при первом входе пользователя в систему не будет отображена страница согласия на предоставление данных этой системе;
- отметить при необходимости опцию «Обязательное использование Proof Key for Code Exchange (RFC 7636) для Authorization code grant type», если запросы на аутентификацию должны выполняться согласно RFC 7636;
- выбрать при необходимости метод аутентификации при обращении к сервису выдачи маркеров. Указанные методы аутентификации должны использоваться при обращении к сервису выдачи маркеров (token endpoint). При пустом значении доступны все методы;
- выбрать при необходимости допустимые grant type. Параметр определяет список grant type, которые будут доступны приложению. При пустом списке доступны все grant type;
- выбрать при необходимости допустимые response type. Параметр определяет список response type, которые будут доступны приложению при обращении к URL авторизации (authorization endpoint). При пустом списке доступны все response type;
- указать время жизни маркера доступа (в секундах). Если параметр не задан, то берется из общих настроек из раздела «OAuth 2.0»;
- указать режим выдачи маркеров доступа по умолчанию. Blitz Identity Provider предусматривает два режима выдачи маркеров доступа (access_token):
  - offline-режим – при запросе маркера доступа будет выдан также бессрочный маркер обновления (refresh_token), которые может быть использован для

получения нового маркера доступа. Приложению рекомендуется использовать этот режим, если оно должно получать актуальные данные пользователя из Blitz Identity Provider за пределами времени действия пользовательской сессии. Например, если приложение делает почтовую рассылку и перед ее отправкой хочет получить актуальный адрес электронной почты из Blitz Identity Provider;

- online-режим – будет выдан только маркер доступа. Приложению рекомендуется использовать этот режим, если ему достаточно получать актуальные данные пользователя в момент входа (в течение активной сессии пользователя).

Режим выдачи маркеров доступа может быть явно указан в запросе на проведение идентификации; 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 код авторизации при вызове Token Endpoint будут выданы не только маркер доступа (access token) и маркер обновления (refresh token), но и маркер идентификации (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 сессионных атрибутов (подробнее см. п. 5.3.3);
- выбрать формат маркера доступа – можно выбрать opaque или JWT. Если параметр не задан, то берется из общих настроек из раздела «OAuth 2.0».

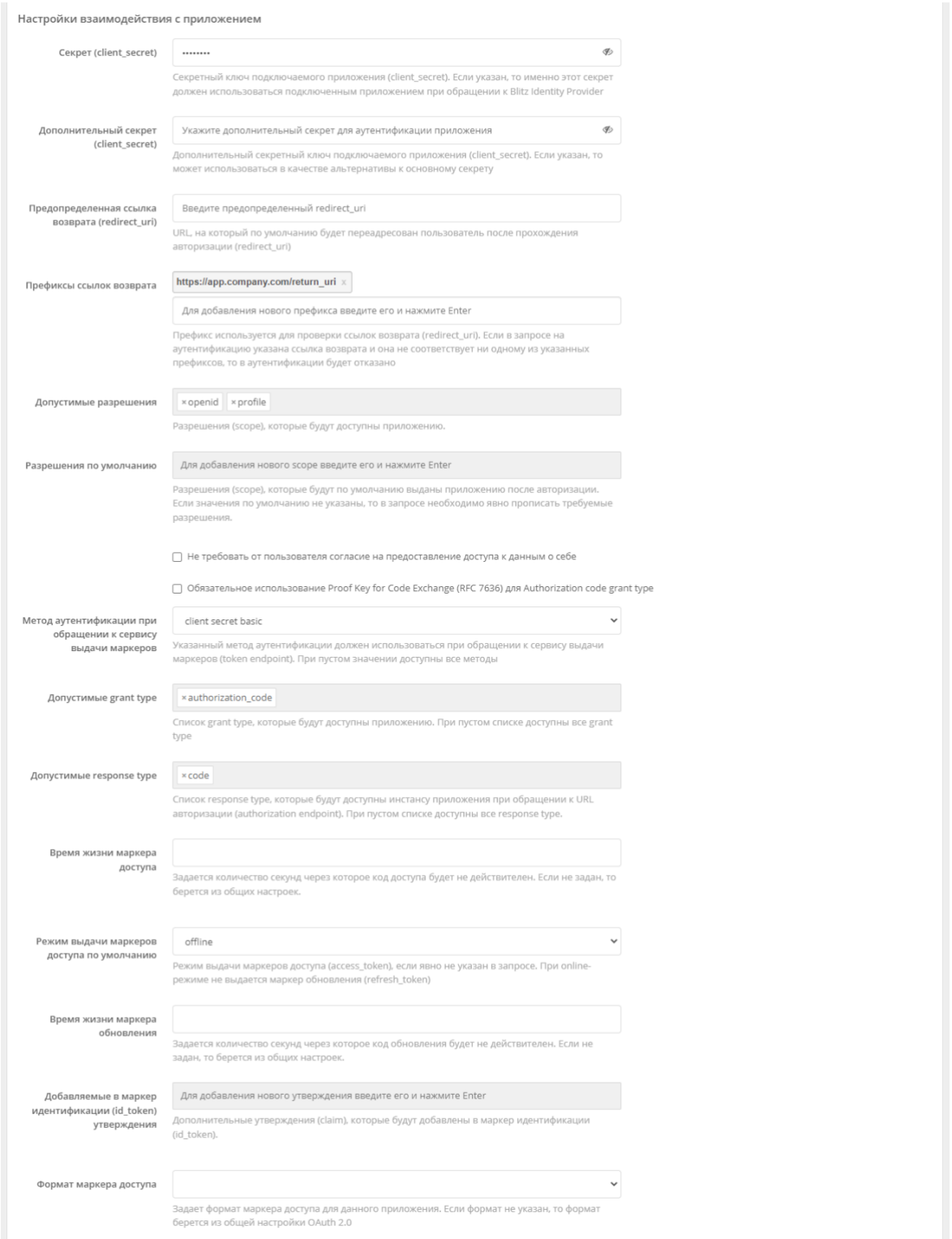

Рисунок 73 – Настройки взаимодействия с приложением по OAuth 2.0 и OIDC

При использовании в приложении функции логаута[30] в блоке «Выход из приложения» необходимо задать следующие настройки (см. Рисунок 74):

- указать префиксы ссылок возврата при выходе. Необходимо перечислить префиксы допустимых URL страниц перенаправления пользователя после инициирования

[30] См.: https://openid.net/specs/openid-connect-rpinitiated-1_0.html#RPLogout

приложением логаута. Допустимо задать один или несколько префиксов ссылок возврата;

- предопределенная ссылка возврата при выходе – ссылка, на которую будет перенаправлен пользователь после логаута из приложения, если в параметрах вызова логаута от приложения не был передан адрес возврата post_logout_redirect_uri;
- отметить при необходимости опцию «Не показывать пользователю экран с подтверждением выхода из системы» – если эту настройку не отметить, то пользователю будет показан экран с запросом подтверждения выхода из приложения;
- ссылка для очистки сессии пользователя в браузере (Front channel) – указанный адрес обработчика приложения будет вызван из фрейма браузера в случае инициирования логаута пользователя;
- отметить при необходимости опцию «Добавлять идентификатор сессии и эмитента в ссылку очистки сессии в браузере (Front channel)» – в этом случае в браузерный обработчик логаута приложения будет передан идентификатор сессии (sid);
- ссылка для очистки сессии пользователя в приложении (Back channel) – указанный адрес обработчика приложения будет вызван с сервера Blitz Identity Provider в случае инициирования логаута пользователя;
- отметить при необходимости опцию «Добавлять идентификатор сессии и эмитента в ссылку очистки сессии в приложении (Back channel)» – в этом случае на адрес обработчика приложения, вызванный с сервера Blitz Identity Provider в случае инициирования логаута пользователя, будет передан logout_token, содержащий идентификатор сессии пользователя (sid).

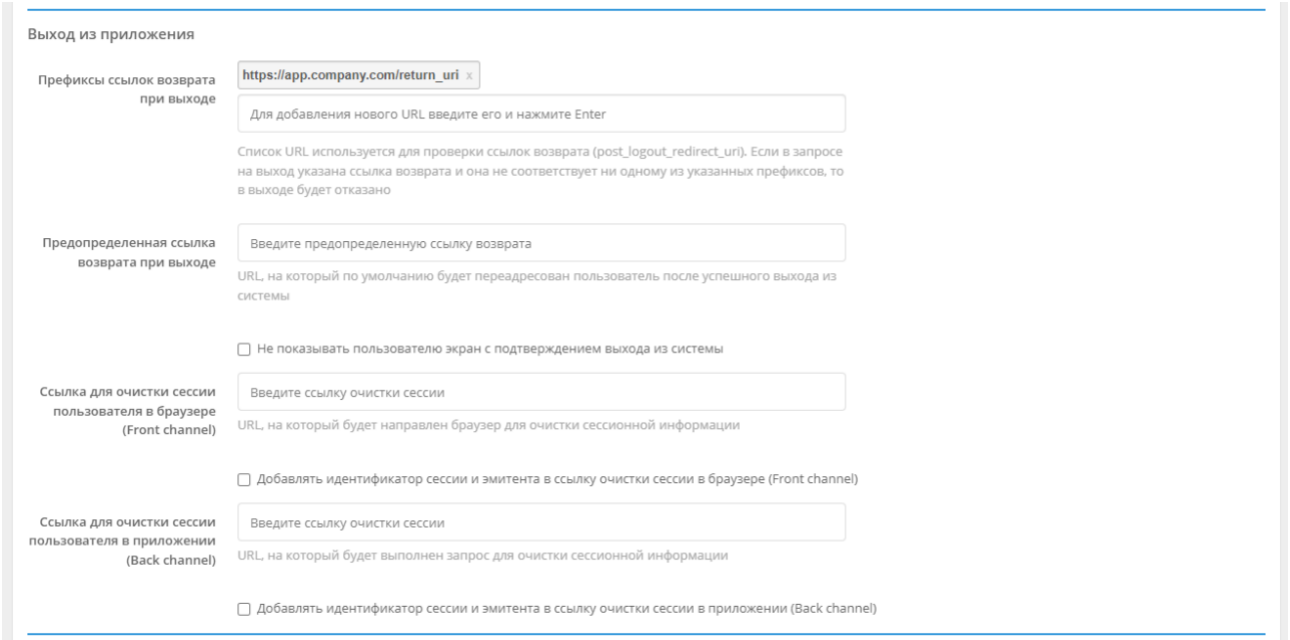

Рисунок 74 – Настройки выхода из приложения

При использовании приложением авторизации по спецификации Device Authorization Grant[31] (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 (см. Рисунок 75):

- формат пользовательского кода, для этого следует использовать регулярные выражения;
- время жизни пользовательского кода;
- ссылка на страницу ввода пользовательского кода;
- отметить при необходимости опцию «Добавлять в URL пользовательский код». В этом случае Blitz Identity Provider при авторизации будет возвращать не только ссылку на станицу ввода пользовательского кода (например, https://test.ru/device), но еще и ссылку с кодом в качестве параметра (например, https://test.ru/device?uc=676-267-324).

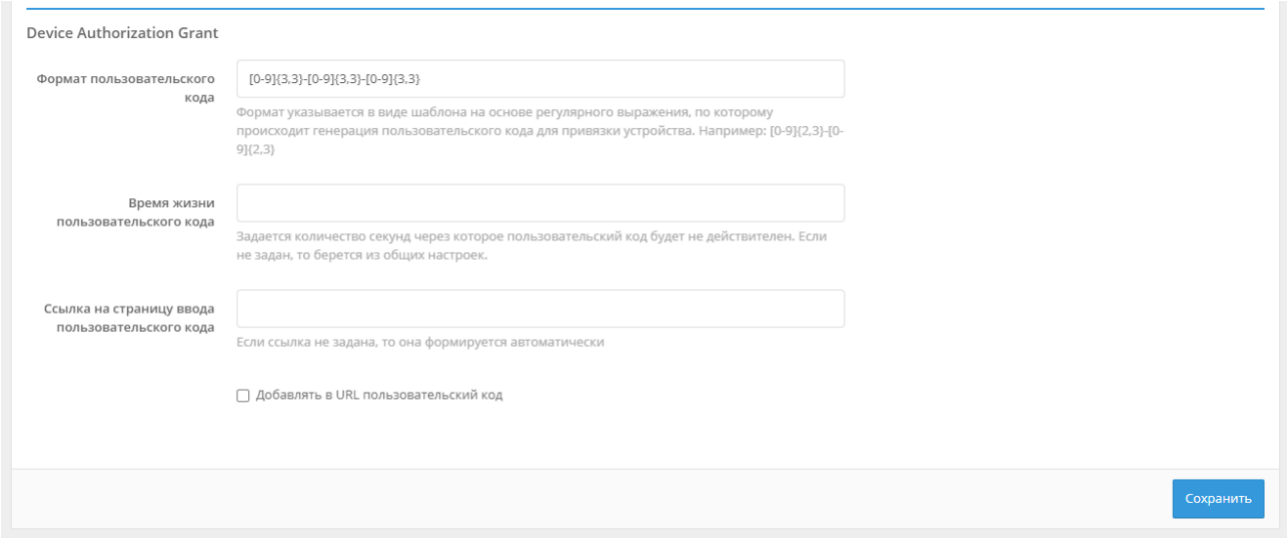

Рисунок 75 – Настройки Device Authorization Grant

5.3.2. 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 (см. Рисунок 76).

[31] См.: https://tools.ietf.org/html/rfc8628

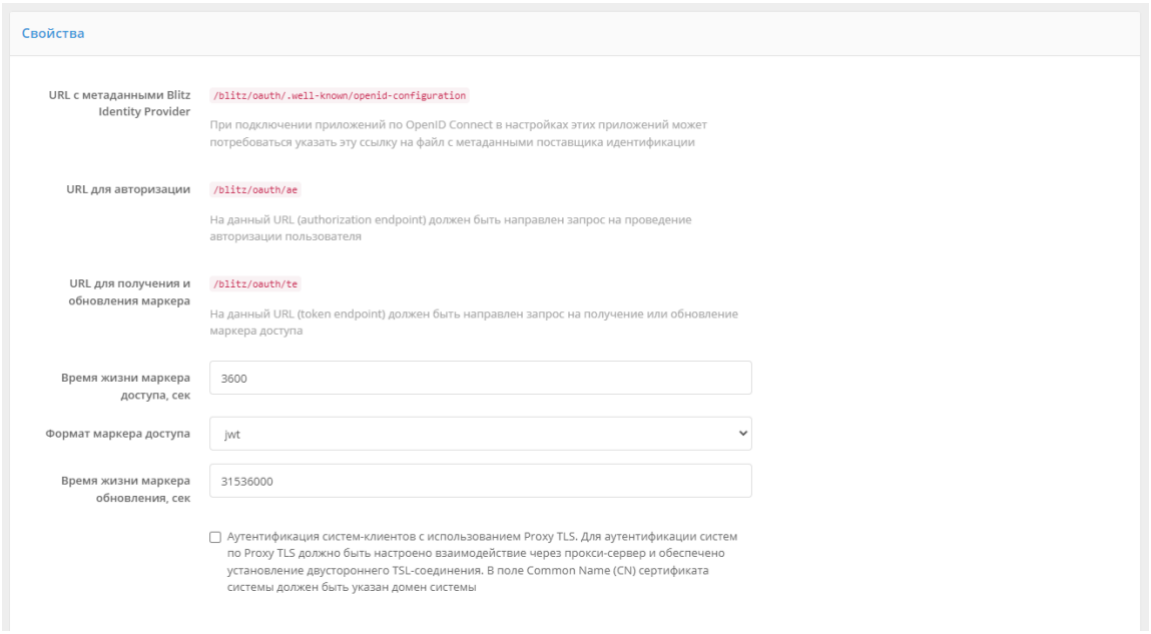

Рисунок 76 – Задание общих настроек OAuth 2.0 и OIDC

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[32]. Разработчики приложений могут не прописывать все указанные ниже URL в конфигурации своего приложения, а использовать в настойках единую ссылку на эти метаданные;
- «URL для авторизации» – адрес обработчика OAuth 2.0 Authorization Endpoint для запросов через браузер на получение кода авторизации;
- «URL для получения и обновления маркера» – адрес обработчика OAuth 2.0 Token Endpoint для получения маркеров безопасности (access_token, id_token, refresh_token).

При необходимости можно:

- изменить «Время жизни маркера доступа», используемое по умолчанию при выпуске маркеров для всех приложений;
- указать «Формат маркера доступа», используемый по умолчанию при выпуске маркеров для всех приложений: строка (opaque) или JWT;
- изменить «Время жизни маркера обновления», используемое по умолчанию при выпуске маркеров для всех приложений;
- отметить опцию «Аутентификация систем-клиентов с использованием Proxy TLS». В этом случае должно быть настроено взаимодействие приложений с Blitz Identity Provider через прокси-сервер с установкой двустороннего TSL-соединения. В поле

[32] См.: https://tools.ietf.org/html/draft-ietf-oauth-discovery-10

«Common Name (CN)» сертификата системы должен быть указан домен системы подключаемого приложения.

В разделе можно Device Authorization Grant можно определить общие настройки для взаимодействия с приложениями по этой спецификации:

- 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 индивидуальные настройки, связанные со спецификацией Device Authorization Grant (см. п. 5.3.1).

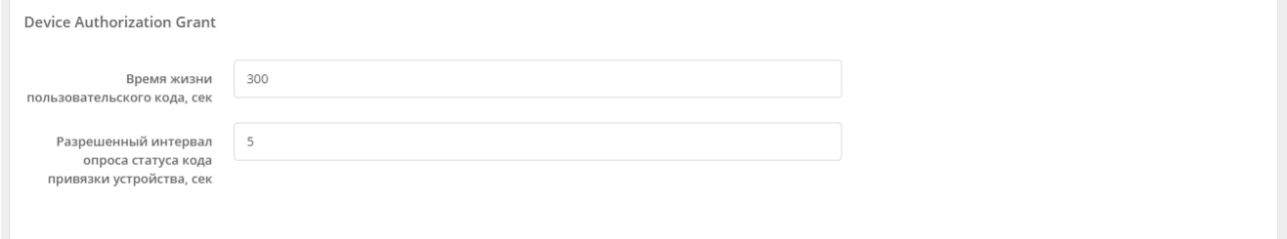

Рисунок 77 – Настройки Device Authorization Grant

Для корректной работы взаимодействия с приложениями по протоколу OAuth 2.0 необходимо определить разрешения (scope). Для этого нужно указать:

- название разрешения;
- описание разрешения (оно будет отображаться пользователю на странице согласия на предоставление доступа);
- атрибуты пользователя, которые будут предоставлены по данному разрешению (атрибуты должны быть определены в меню «Источники данных»);
- является ли разрешение системным – такие разрешения предоставляются приложениям только с использованием OAuth 2.0 Client Credentials Flow (не в контексте разрешения отдельного пользователя, а общие).

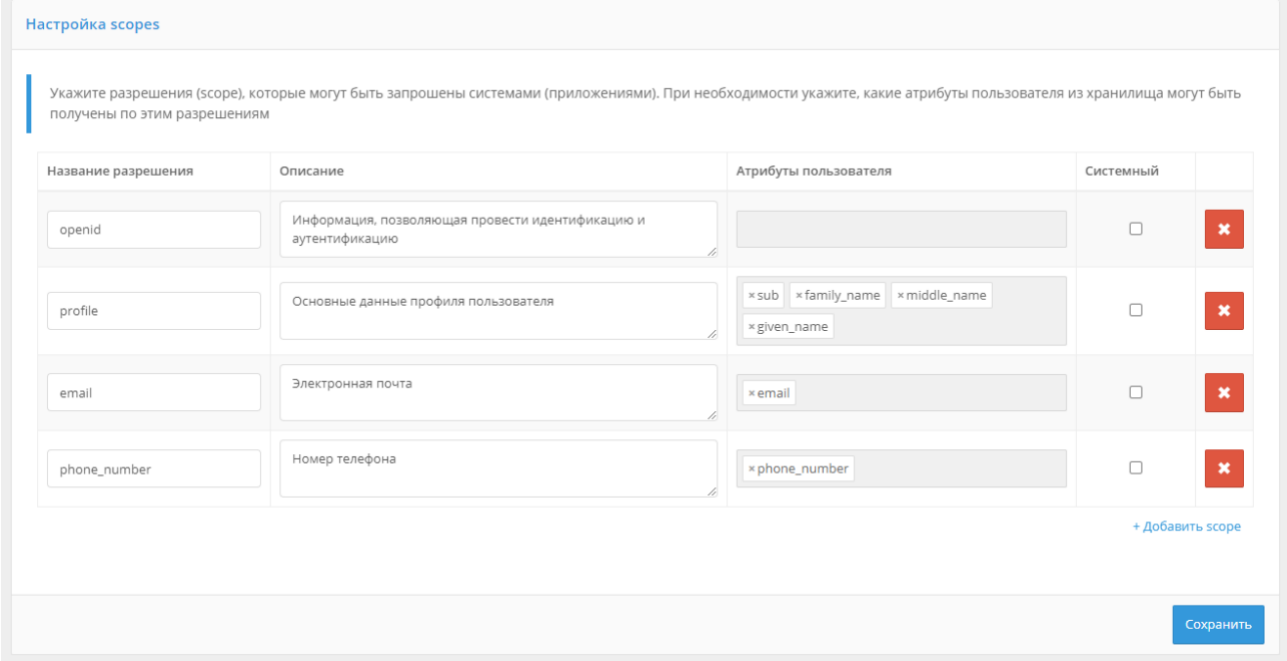

Рисунок 78 – Настройки scopes

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[33].

5.3.3. Добавление атрибутов в маркер идентификации

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 (см. Рисунок 73).

Помимо хранимых атрибутов, в маркер идентификации могут быть добавлены утверждения:

- полученные при входе пользователя по электронной подписи. Это могут быть данные о сертификате ключа электронной подписи, данные о физическом / юридическом лице из сертификата;
- полученные при входе через ЕСИА;
- определенные в процедуре входа.

Для получения утверждений из сертификата ключа электронной подписи необходимо отредактировать конфигурационный файл blitz.conf, добавив в блок настроек blitz.prod.local.idp.login.methods.x509 добавить структуру следующего содержания:

```
"claims" : [
  {
```

[33] В этом случае указанные данные могут быть получены по маркеру доступа (access token), выданному на разрешение openid.

```
    "name" : "attr_name",
    "value" : "cert attr name"
  }
]
```

В этой структуре attr_name – имя атрибута, которое будет использовано в маркере идентификации, а cert_attr_name – обозначение атрибута в сертификате (примеры доступных значении приведены в таблице).

Таблица 4

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 OID 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 (Таблица 5).

Таблица 5

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	Электронная почта
mobile	Моб. телефон

Пример добавляемой в конфигурационный файл структуры:

```
"claims" : [
   {
    "name" : "esia firstName",
    "value" : "firstName"
   },
   {
    "name" : "esia_lastName",
    "value" : "lastName"
   },
   {
    "name" : "esia_middleName",
    "value" : "middleName"
   },
   {
    "name" : "esia_birthDate",
    "value" : "birthDate"
   },
   {
    "name" : "esia_gender",
    "value" : "gender"
   },
   {
    "name" : "esia_snils",
    "value" : "snils"
   },
   {
    "name" : "esia_inn",
    "value" : "inn"
   },
   {
    "name" : "esia passport",
    "value" : "passport"
   },
   {
    "name" : "esia_birthPlace",
    "value" : "birthPlace"
   },
   {
    "name" : "esia_email",
    "value" : "email"
   },
   {
    "name" : "esia_mobile",
    "value" : "mobile"
   }
]
```

Чтобы иметь возможность определять сессионные утверждения в процедуре входа, соответствующие утверждения также должны быть определены в конфигурационной файле.

Для этого в раздел blitz.prod.local.idp.login конфигурационного файла необходимо добавить атрибут sessionClaims с перечнем утверждений, которые могут быть определены в процедуре.

Например, следующая запись позволяет определить атрибут custom_attr:

```
"sessionClaims" : [
   "custom attr"
]
```

5.3.4. 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. п. 5.3.2);
- в настройках OAuth 2.0 для данного приложения перейти на закладку «Динамические клиенты» (см. Рисунок 79).

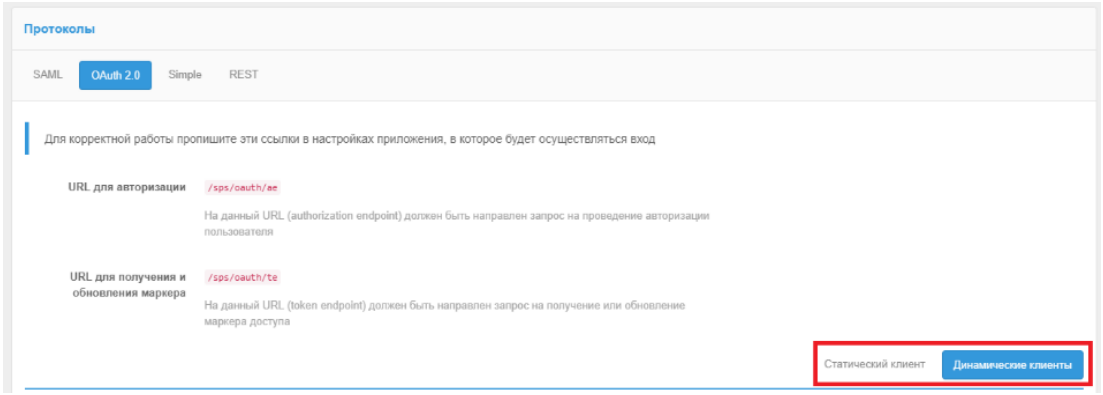

Рисунок 79 – Включение динамической регистрации клиентов

Указать базовые настройки динамической регистрации клиентов:

- разрешить динамическую регистрацию клиентов;
- указать допустимые к прямой передаче утверждения. Эти утверждения допускается указывать в запросе на регистрацию экземпляра приложения. В случае их наличия в метаданных приложения (software_statement), приоритет будет отдан значению из метаданных. Рекомендуется разрешить передачу только типа устройства (device_type).

Создать первичные маркеры для приложения. Первичные маркеры используются для авторизации 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 (см. Рисунок 80).

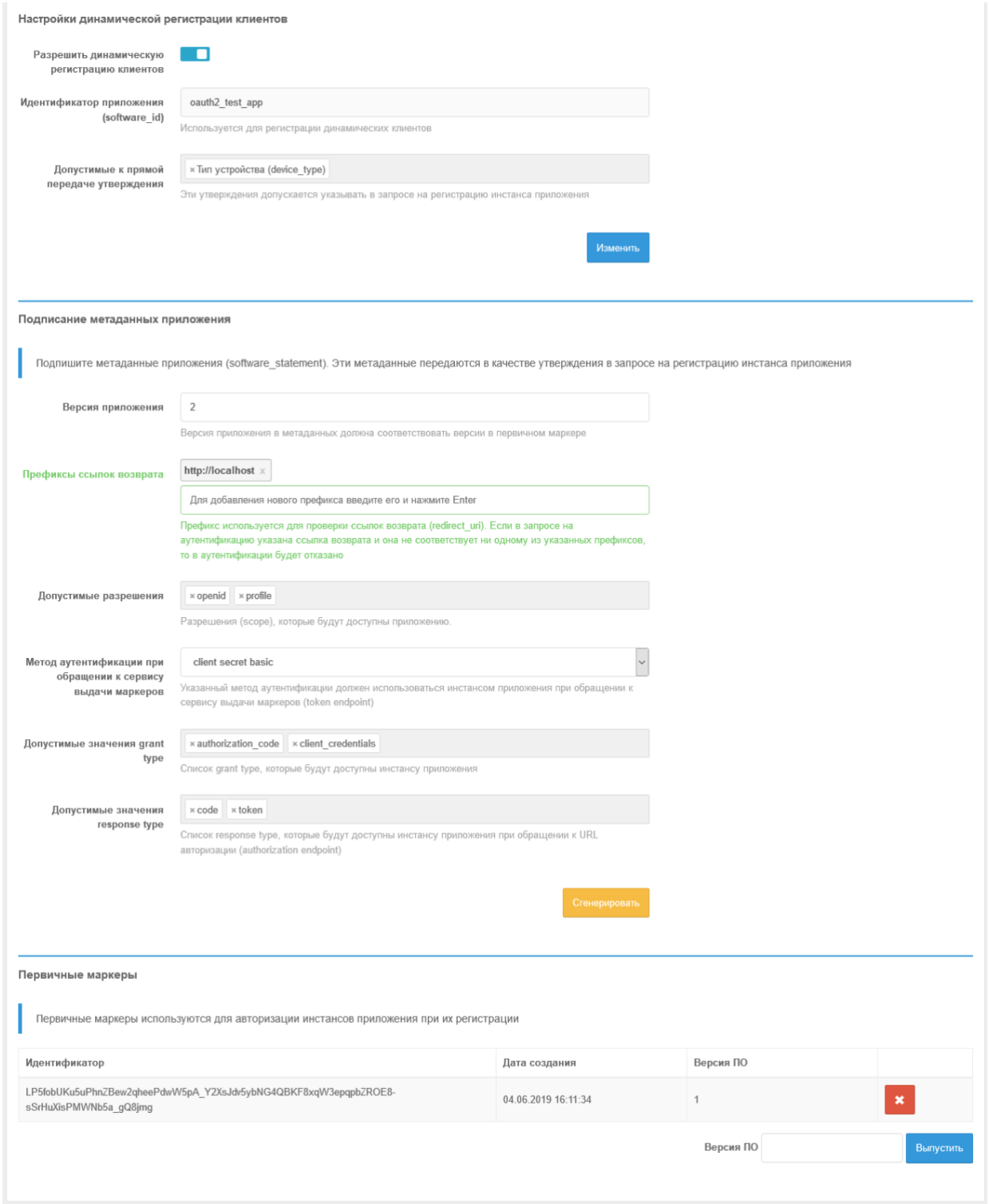

Рисунок 80 – Настройки динамической регистрации клиентов

5.4. Настройка Simple

Данный способ подключения приложения к Blitz Identity Provider можно применять при следующих условиях:

- Приложение нельзя подключить к Blitz Identity Provider с использованием стандартных протоколов SAML или OIDC.
- Приложение представляет собой веб-приложение, развернутое в собственной

инфраструктуре (On-Premise). Доступ пользователей к приложениям можно организовать через реверсивный прокси-сервер.

Чтобы подключить приложение к Blitz Identity Provider по протоколу Simple, необходимо:

1. В настройках приложения в консоль управления выбрать протокол Simple и задать его настройки:
   - SSL – настройка, указывающая, производится ли за прокси вызов подключаемого по Simple приложения по HTTP или по HTTPS. Рекомендуется в качестве прокси-сервера, защищающего приложение, использовать существующий веб-сервер приложения, и в таком случае соединение прокси-сервера с приложением будет осуществляться без TLS/SSL шифрования.
   - Селектор формы – задается CSS-селектор, позволяющий определить положение формы входа на странице подключаемого приложения.
   - Селектор поля с логином – задается CSS-селектор, позволяющий определить положение поля ввода логина на странице входа подключаемого приложения.
   - URL выхода по умолчанию (опциональная настройка) – указывает, какой адрес должен вызвать Blitz Identity Provider при необходимости инициировать логаут в подключенном по Simple приложении в случае единого логаута в Blitz Identity Provider.
   - URL для перехода после успешного выхода – указывает, какой адрес должен вызвать Blitz Identity Provider для перенаправления пользователя после успешного логаута, инициированного подключенным по Simple приложением.
   - JavaScript (опциональная настройка) – встраиваемый в страницу входа подключаемого по Simple приложения JS-код, позволяющий обработать полученный от приложения ответ с результатами входа (проверить, что вход произведен успешно) и показать об этом ошибку в Blitz Identity Provider.

Пример значения:

```
var fm = document.querySelector('form[name=login]');

if (fm) {
          document.body.style.display = "none";
          var err = document.getElementById('lost-password');
          var errKey = err && err.innerHTML.indexOf('Неправильный пароль.') !== -1 ?
'incorrect_password' : 'unknown_error';
          var kvp = document.location.search.substr(1).split('&');
          kvp.push([encodeURI('error'),encodeURI(errKey)].join('='));
          window.location.search = kvp.join('&');
}

var aLogout = document.querySelector('#logout');
var href = aLogout ? aLogout.getAttribute("href") : null;
if (href) {
    var lp = encodeURIComponent(href);
    var slp = document.createElement('script');
```

```
        slp.setAttribute('src', 'https://idp.company.com/blitz/simple/slp?app=app_id&lp=' + lp);
        document.head.appendChild(slp);
}
```

Пример настроек протокола Simple для приложения представлен на рисунке ниже (см. Рисунок 81).

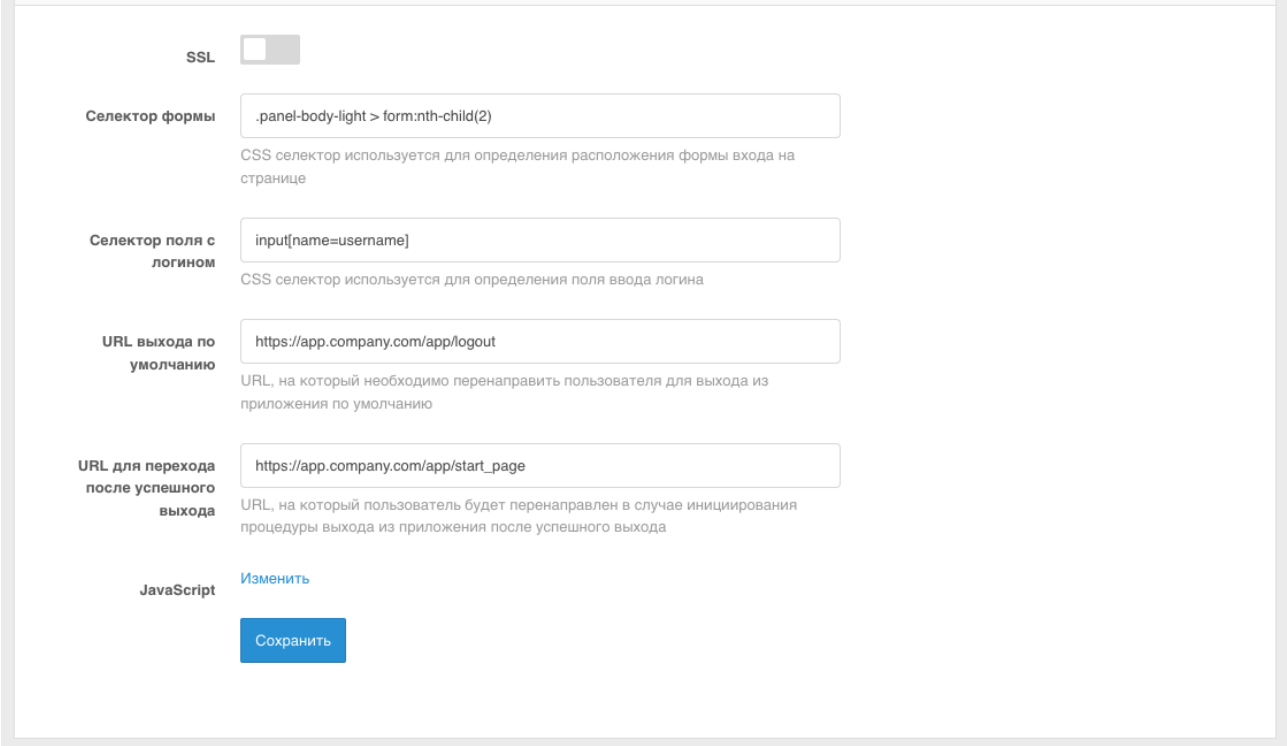

Рисунок 81 –Настройки взаимодействия с приложением по Simple

2. Задать настройки проксирования запросов к приложению на веб-сервере.

   Пример конфига для веб-сервера nginx:

```
map "" $idp_host {
        default <hostname сервера Blitz>: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 <hostname сервера приложения>:<порт приложения>;
}

server {
    listen 80;
    server name <доменное имя приложения>;
    # enforce https
    return 301 https://$server_name$request_uri;
}

server {
    listen         443 ssl;
    server_name    <доменное имя приложения>;

    resolver               172.27.0.20 172.25.0.50 valid=300s;
    #resolver               8.8.8.8 valid=300s;

    #ssl_certificate        /etc/nginx/cert/<путь к сертификату в случае подключения по SSL>.pem;
```

```
    #ssl_certificate_key    /etc/nginx/cert/<путь к контейнеру в случае подключения по SSL>.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 страницы входа в приложение>$ {
            #if ($http_referer ~* "/blitz/simple") {
            #    set $idp post login "1";
            #}
            if ($http_referer ~* "<доменное имя Blitz>") {
                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 ($activLogout = "1") {
            return 302 https://<доменное имя Blitz>/blitz/simple/active_logout?app=$host;
        }
        proxy pass http://oc-web$request uri;
    }
    location / {
        proxy_pass http://oc-web;
    }
}
```

5.5. Настройка клиента REST-сервисов Blitz Identity Provider

Для вызова REST-сервисов Blitz Identity Provider необходимо настроить приложение, которое будет выступать в качестве системы-клиента REST-сервисов. Для этого нужно зарегистрировать новое приложение в разделе «Приложения» (см. п. 5.1).

Далее перейти к настройкам приложения, в качестве протокола подключения указать REST и заполнить следующие данные (см. Рисунок 82):

- «Пароль» – будет использоваться при HTTP Basic авторизации. В качестве логина используется идентификатор приложения;
- «Допустимые CN» – перечень значений атрибута CN сертификата, используемого при TLS-аутентификация;

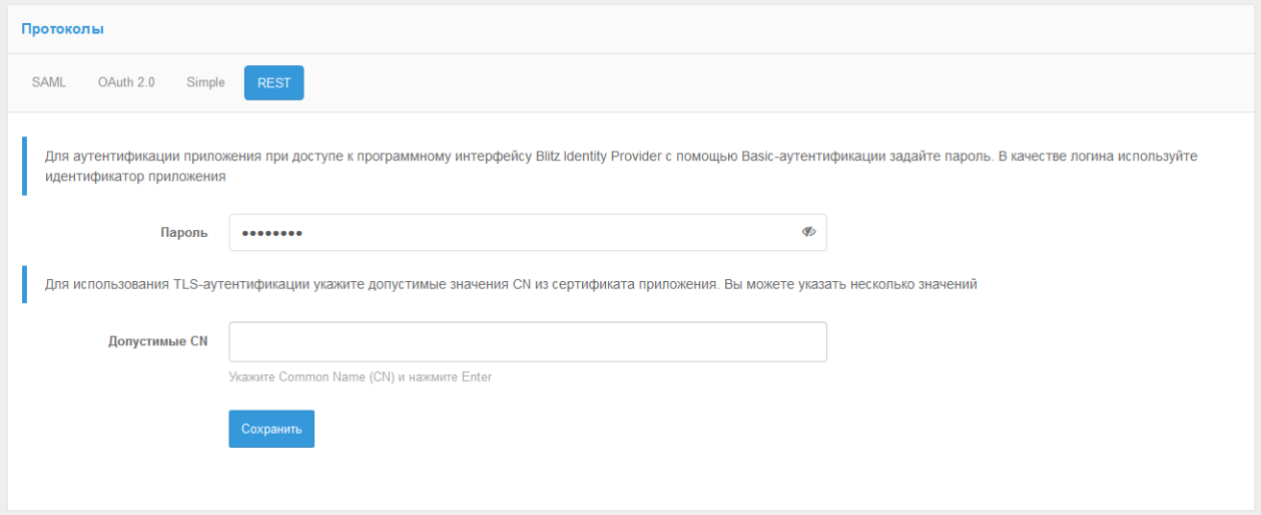

Рисунок 82 – Настройка приложения для работы с REST-сервисами

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.

5.6. 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.

Remote Authentication Dial In User Service (RADIUS)[34] — протокол, используемый для централизованного управления авторизацией, аутентификацией и учетом доступа в сетевые службы и оборудование. Через данный протокол выполняется взаимодействие между сервером и клиентом RADIUS. После запроса пользователем доступа в сетевую службу соответствующий клиент отправляет серверу запрос, в результате которого сервер проверяет наличие пользователя в базе данных. Если пользователь найден, сервер отправляет клиенту разрешение на его аутентификацию.

Сервером RADIUS выступает Blitz Identity Provider, клиентом — подключенная сетевая служба. Сервер поддерживает как первый, так и второй факторы аутентификации. В текущей реализации сервер выполняет поиск пользователей во всех подключенных хранилищах. Сетевые службы настраиваются в Blitz Identity Provider как приложения.

5.6.1. Конфигурирование сервера RADIUS

Для конфигурирования сервера RADIUS в Blitz Identity Provider выполните следующие действия:

[34] https://datatracker.ietf.org/doc/html/rfc2865

1) В консоли управления перейдите в раздел **RADIUS**. Последовательно настройте конфигурацию сервера.
2) **Общие настройки**: на данной вкладке указываются общие настройки сервера RADIUS.

   – **Статус**: включение сервера.
   – **Сетевой адрес привязки**: список адресов, с которых сервер обрабатывает запросы. 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. Данное время должно быть согласовано с RADIUS-клиентом за счет корректной настройки времени ожидания ответа RADIUS-сервера.

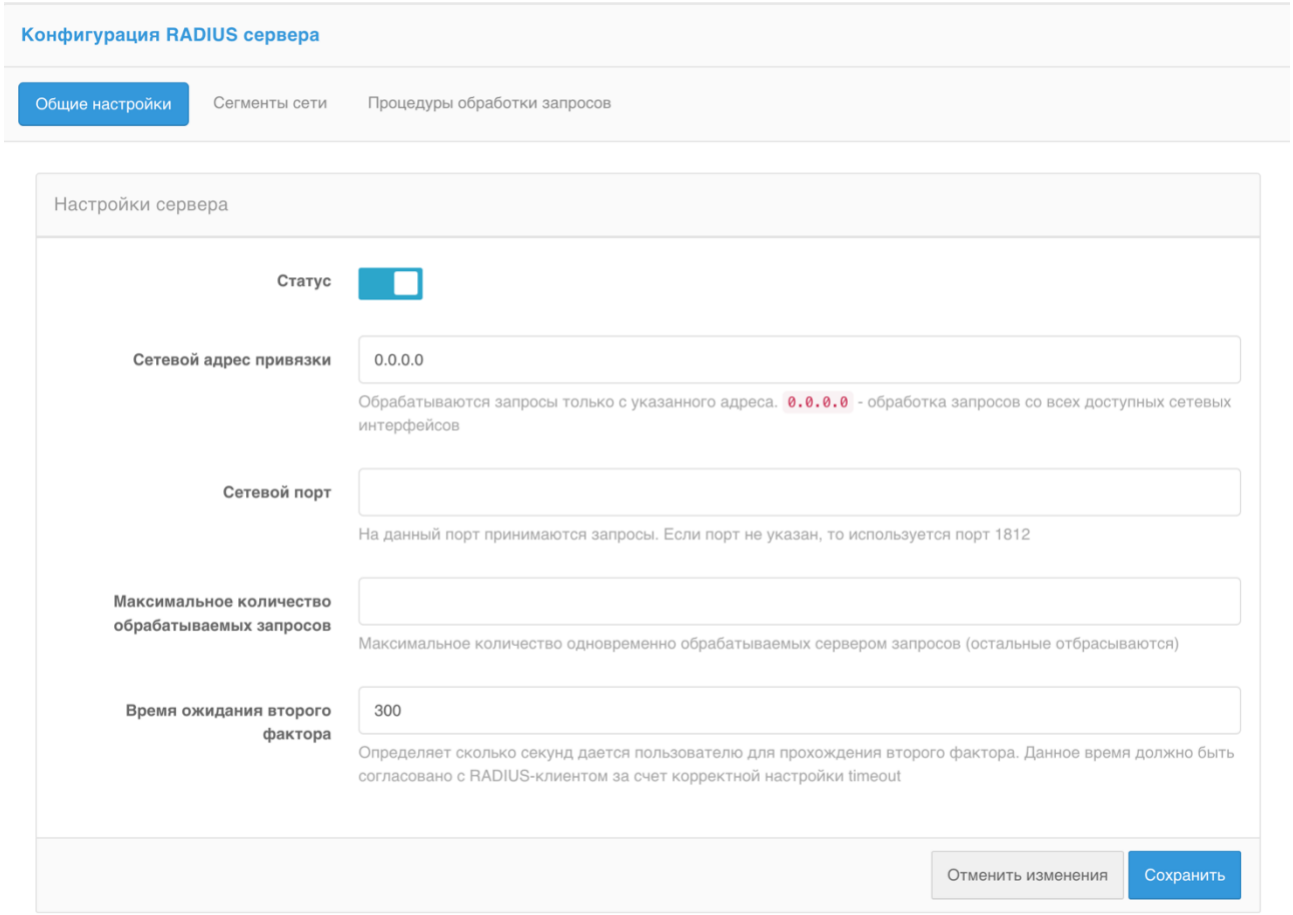

Рисунок 83 – Общие настройки сервера RADIUS

Нажмите **Сохранить**.

3) **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**. 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.
   - **Приложение по умолчанию**: выберите приложение, с которым будет ассоциироваться запрос из данной подсети. Если приложений несколько, оно будет выступать приложением по умолчанию.
   - Соответствие **NasId** и приложений: если предполагается, что из одной подсети запрашивать аутентификацию будет несколько приложений, задайте **NasId**, по которым сервер RADIUS будет их идентифицировать.

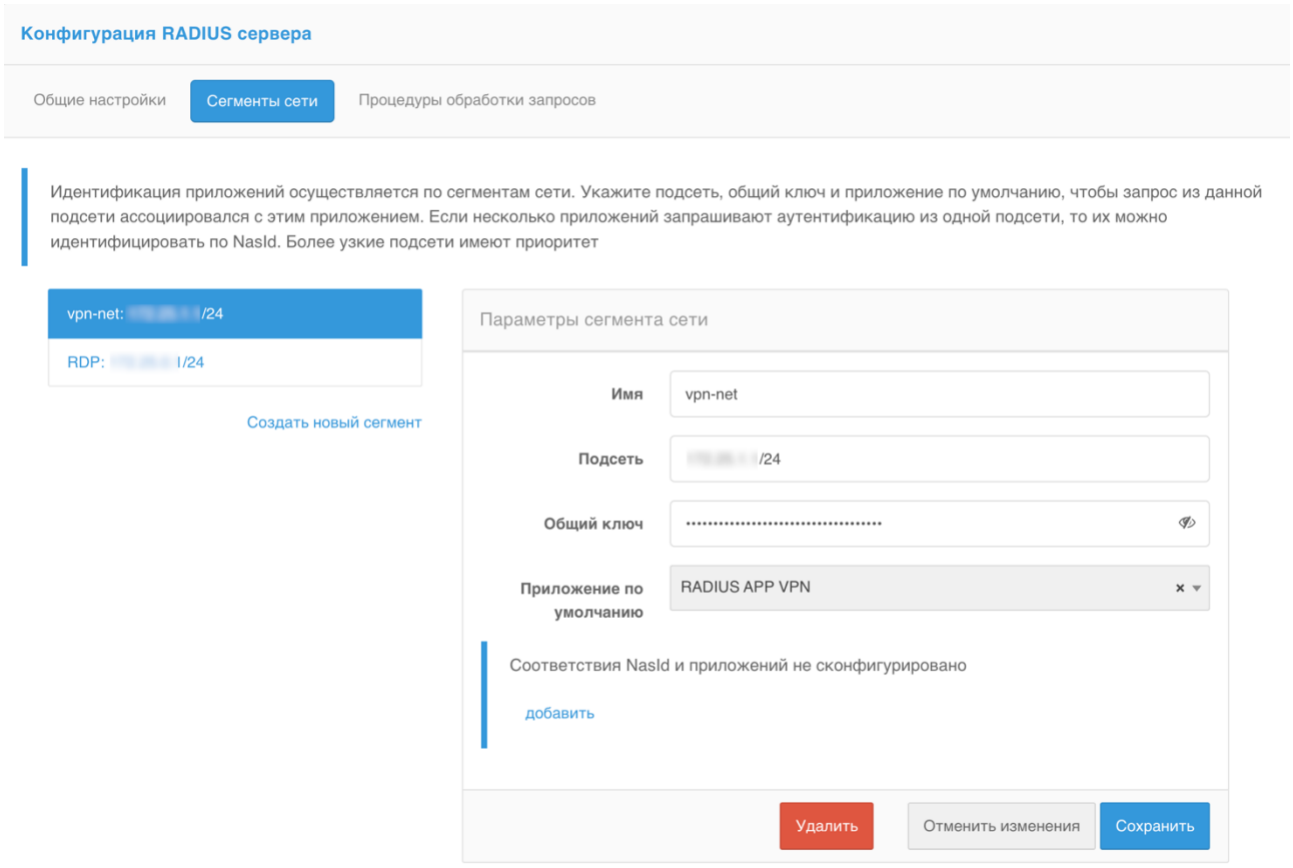

Рисунок 84 – Сегменты сети

Нажмите **Сохранить**.

4) **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:

   – Нажмите **Создать новую процедуру обработки запросов**.
   – Задайте настройки:
     - **Статус**: включение процедуры.
     - **Идентификатор процедуры**: задайте идентификатор процедуры. Java класс, описывающий процедуру обработки запросов, должен иметь такое же название.
     - **Описание**: введите описание процедуры.

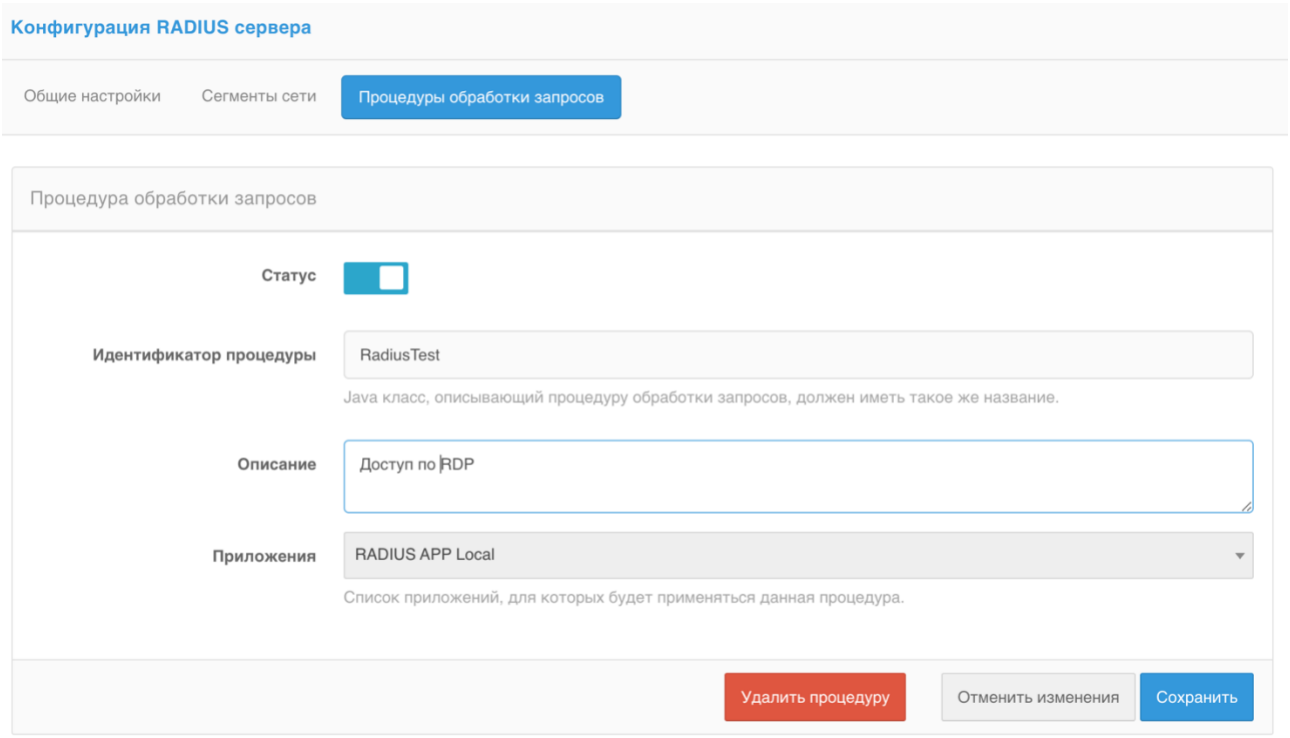

Рисунок 85 – Процедуры обработки запросов

– Нажмите **Сохранить**.
– Введите исходный код процедуры. Для управления процессом обработки RADIUS запросов необходимо написать на языке Java класс, реализующий интерфейс RadiusFlow.

В случае использования второго фактора аутентификации будет достаточно процедуры по умолчанию. В случае использования второго фактора аутентификации будет достаточно процедуры по умолчанию. Вызовите RadiusResult.more("method"), где method принимает одно из следующих значений: sms, push, totp, hotp, email, prfc (подтверждение в Личном кабинете пользователя). При подтверждении через Личный кабинет в нем появляется сообщение о попытке входа, в котором пользователю требуется нажать **Подтвердить**. Для того чтобы фактор сработал, Личный кабинет должен быть открыт с обязательным прохождением двух факторов аутентификации.

Пример процедуры 2FA через подтверждение в Личном кабинете:

```
package com.identityblitz.idp.radius.flow;
public class TestRadius implements RadiusFlow {
  public String loginN12(final String login) {
    return login;
  }
  public RadiusResult next(final RadiusContext context) {
    if (context.factor() == 1) {
      return RadiusResult.more("prfc");
```

```
    }
    return  RadiusResult.authenticated(context.subject());
  }
}
```

В случае использования первого фактора, деактивируйте условие *if (context.factor() == 1)*.

```
package com.identityblitz.idp.radius.flow;
public class TestRadius implements RadiusFlow {
  public String loginN12(final String login) {
    return login;
  }
  public RadiusResult next(final RadiusContext context) {
    return  RadiusResult.authenticated(context.subject());
  }
```

Для компиляции нажмите **Сохранить**.

5.6.2. Настройка приложения

Для настройки приложения выполните следующие действия:

1) В консоли управления перейдите в раздел **Приложения**. Создайте приложение с базовыми настройками.

   – **Идентификатор (entityID или client_id)**,
   – **Название**,
   – **Домен**: домен сетевой службы.

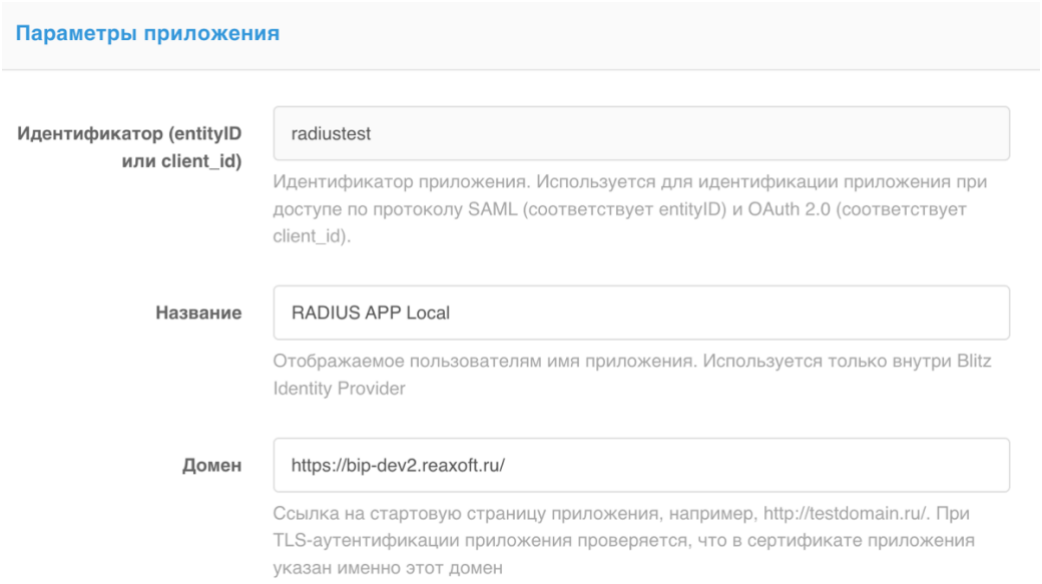

Рисунок 86 – Параметры приложения сетевой службы

Нажмите **Сохранить**.

2) В секции **Протоколы** приложения на вкладке **RADIUS** задайте следующие настройки:

   – Поставьте флажок **Пароль проверяется приложением самостоятельно**, если Blitz Identity Provider будет использоваться для второго фактора аутентификации.

   – **Время ожидания второго фактора**: время в секундах, которое дается пользователю для прохождения второго фактора. Если параметр не задан, будет взято значение из настроек сервера RADIUS. 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 на сервере RADIUS процедуры.

   Внимательно настраивайте интеграцию на стороне сетевой службы (п. 5.6.3). Если в приходящих от приложения запросах не определен **NasId**, приложение узнается 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.

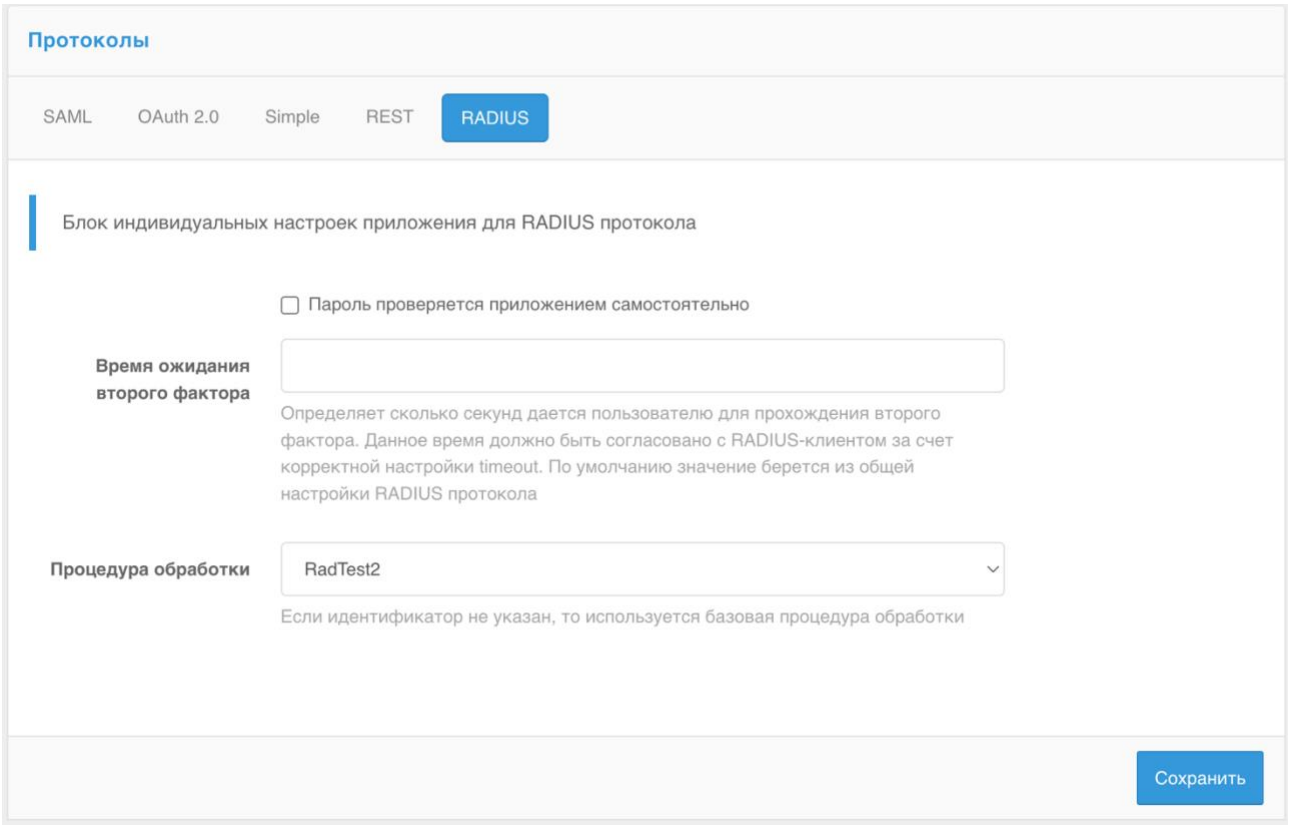

Рисунок 87 – Протокол RADIUS в настройках приложения

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

6. Кастомизация работы Blitz Identity Provider посредством программирования на Java

6.1. Создание процедур входа

Процедуры входа 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 (см. Рисунок 88).

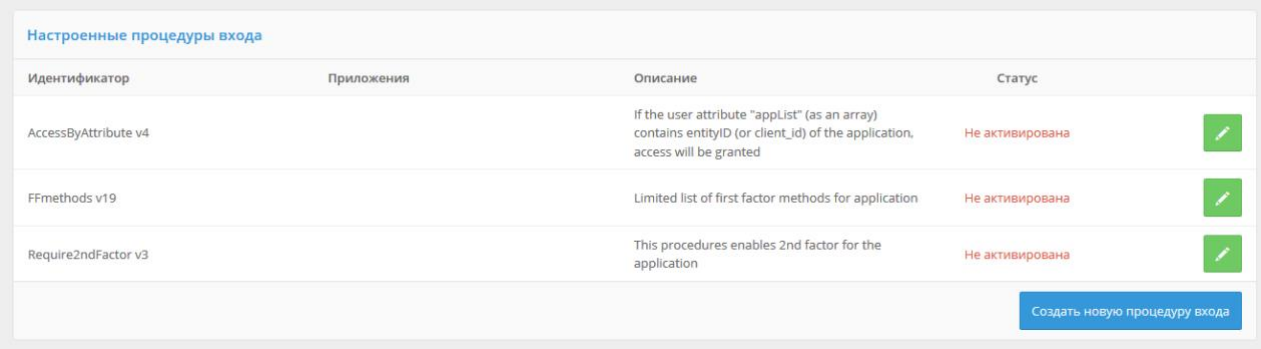

Рисунок 88 – Экран настроек процедур входа

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.

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.

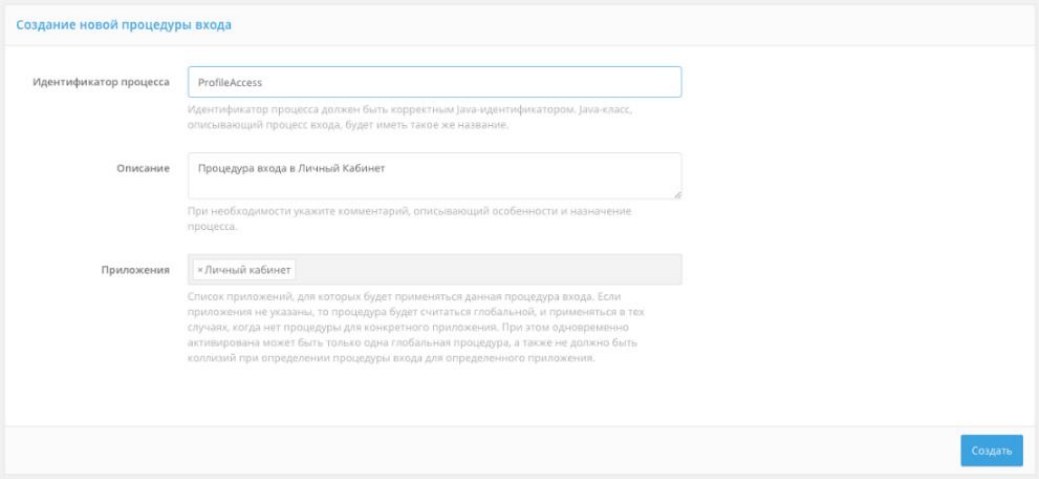

Рисунок 89 – Экран создания новой процедуры входа

2. Написание исходного кода процедуры (см. Рисунок 90). 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 случаях авторизовать пользователя 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.

Если процедура активирована, то сохранить можно только ту процедуру, которую удается скомпилировать. Иными словами, если при редактировании активированной процедуры была выявлена ошибка, то кнопка «Сохранить» работать не будет, а при перезагрузке страницы изменения будут утеряны.

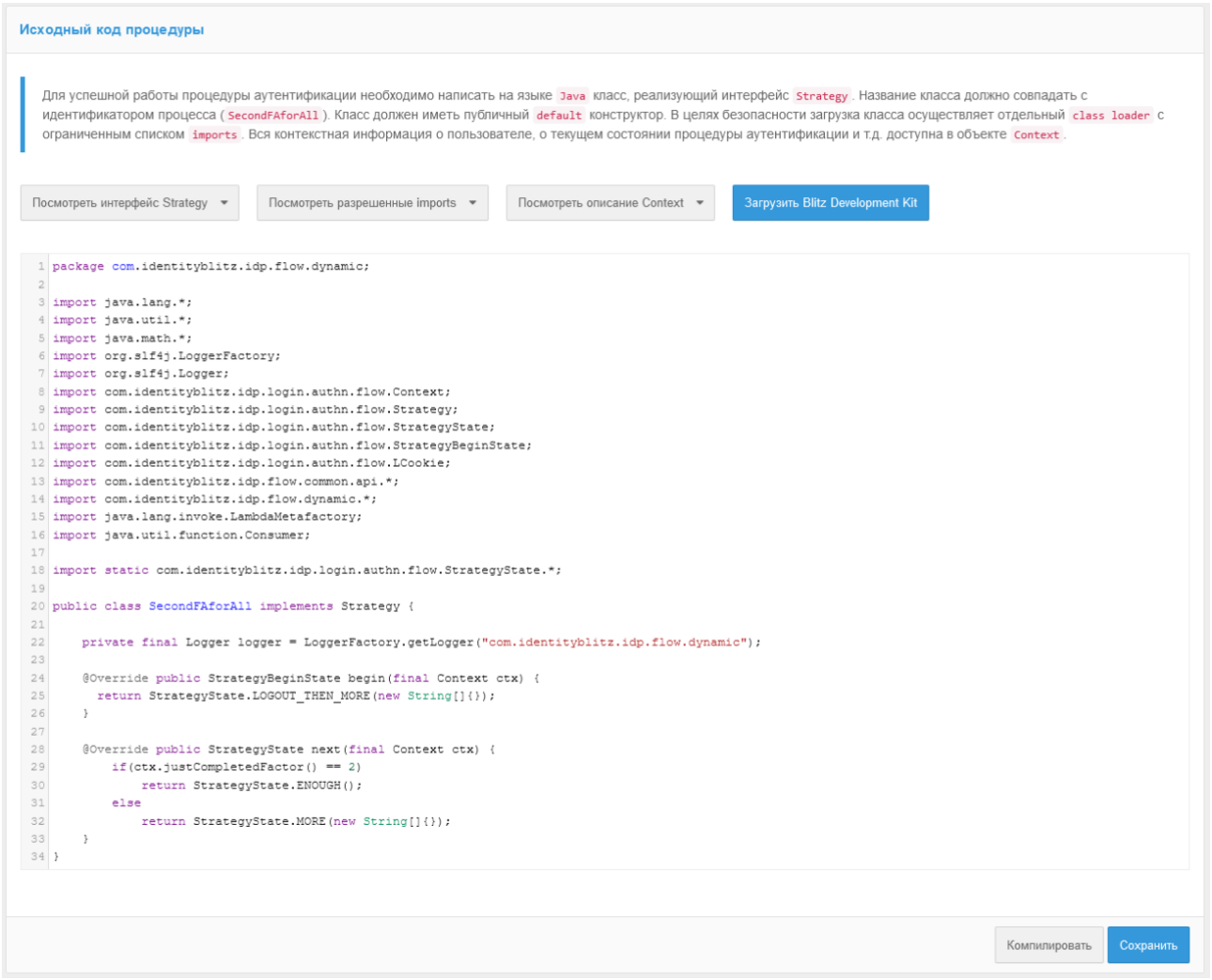

Рисунок 90 – Экран редактирования исходного кода процедуры входа (фрагмент)

6.2. Примеры процедур входа

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

- принудительная двухфакторная аутентификация в приложение (Require2ndFactor);
- ограничение перечня доступных методов первого фактора при входе в приложение (FFmethods);
- предоставление доступа к приложению только при определенном значении атрибута (AccessByAttribute);
- запрет входа в приложение после истечения срока действия учетной записи (AccountExpiresCheck);
- разрешение входа в приложение только из определенных сетей (AllowedIPs);
- запрет работы в нескольких одновременных сессиях (RestrictSessions);
- сохранение в утверждениях (claims) перечня групп пользователя (AddGroupsToToken);
- отображение пользователю объявления при входе (InfoPipe);

- запрос ввода пользователем атрибута или актуализации телефона и email (PipeAttrActAdd);
- запрос ввода пользователем контрольного вопроса, если он не задан в учетной записи (PipeSecQuestion);
- регистрация ключа безопасности (WebAuthn, Passkey, FIDO2) при входе (PipeWebAuthn).

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"));
```

6.2.1. Принудительная двухфакторная аутентификация в приложение

Процедура Require2ndFactor требует двухфакторной аутентификации для доступа к приложению. Если пользователь переходит в приложение в рамках единой сессии, то при наличии одного пройденного фактора у него будет дополнительно проверен второй фактор, т.е. SSO в этом случае не сработает.

```
public class Require2ndFactor implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.flow.dynamic");

    @Override public StrategyBeginState begin(final Context ctx) {
        if(ctx.claims("subjectId") != null){
            if (ctx.sessionTrack().split(",").length < 2)
                return StrategyState.MORE(new String[]{});
            else
                return StrategyState.ENOUGH();
        }
        else {
            return StrategyState.MORE(new String[]{});
        }
    }

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

6.2.2. 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,

ЕСИА);

- spnego – вход по сеансу операционной системы;
- sms – вход по коду подтверждения из SMS-сообщения;
- 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[]{});
        }
    }
}
```

6.2.3. Разрешить вход в приложение только при определенном значении атрибута у пользователя

Процедура AccessByAttribute использует атрибут appList для принятия решения о доступе пользователя к приложению. Для работы этой процедуры необходимо создать атрибут appList в виде массива (Array of strings). В качестве значений элементов этого массива следует использовать идентификаторы приложений. В результате доступ к приложению будет предоставлен, если среди значений appList у данного пользователя будет идентификатор этого приложения. Такая архитектура процедуры позволяет назначить ее сразу нескольким приложениям и регулировать доступ к ним при помощи одного атрибута.

```
public class AccessByAttribute implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.flow.dynamic");

    @Override public StrategyBeginState begin(final Context ctx) {
        if(ctx.claims("subjectId") != null){
            int appListIdx = 0;
            boolean hasAccess = false;
            while (appListIdx > -1) {
                String app = ctx.claims("appList.[" + appListIdx + "]");
                logger.debug("app [" + appListIdx + "] = " + app);
                if (app == null){ appListIdx = -1; }
                else if (app.equals(ctx.appId())) { appListIdx = -1; hasAccess = true; }
                else { appListIdx ++; logger.debug("AppList index = " + appListIdx); }
            }
            if(hasAccess)
                return StrategyState.ENOUGH();
            else
```

```
                return StrategyState.DENY;
        }
        else
            return StrategyState.MORE(new String[]{});
    }

    @Override public StrategyState next(final Context ctx) {
        int appListIdx = 0;
        boolean hasAccess = false;
        while (appListIdx > -1) {
            String app = ctx.claims("appList.[" + appListIdx + "]");
            logger.debug("app [" + appListIdx + "] = " + app);
            if (app == null){ appListIdx = -1; }
            else if (app.equals(ctx.appId())) { appListIdx = -1; hasAccess = true; }
            else { appListIdx ++; logger.debug("AppList index = " + appListIdx); }
        }
        if(!hasAccess)
            return StrategyState.DENY;
          Integer reqFactor = 0;
          if (ctx.user() != null) {
            reqFactor = ctx.user().requiredFactor();
        }
        if (reqFactor == 0)
            return StrategyState.ENOUGH();
        else {
            if (reqFactor == ctx.justCompletedFactor())
                return StrategyState.ENOUGH();
            else
                return StrategyState.MORE(new String[]{});
        }
    }
}
```

Пример упрощенного варианта процедуры – допуск пользователя в приложение при условии, что адрес его электронной почты равен ivanov@company.ru:

```
@Override public StrategyBeginState begin(final Context ctx) {
  if(ctx.claims("subjectId") != null){
    if("ivanov@company.ru".equals(ctx.claims("email")))
      return StrategyState.ENOUGH();
    else
      return StrategyState.DENY;
  }
  else
    return StrategyState.MORE(new String[]{});
}

@Override public StrategyState next(final Context ctx) {
  if(!"ivanov@company.ru".equals(ctx.claims("email")))
    return StrategyState.DENY;
  Integer reqFactor = (ctx.user() == null) ? null : ctx.user().requiredFactor();
  if(reqFactor == null)
    return StrategyState.ENOUGH();
  else {
    if(reqFactor == ctx.justCompletedFactor())
      return StrategyState.ENOUGH();
    else
      return StrategyState.MORE(new String[]{});
  }
}
```

6.2.4. Запрет входа в приложение после истечения срока действия учетной записи

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);
        }
    }
}
```

6.2.5. Разрешение входа в приложение 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;
       }
}
```

6.2.6. 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[]{});
    }
}
```

6.2.7. 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. п. 5.3.3).

При входе в приложение будет проверено наличие групп у пользователя в атрибуте memberOf, и если они там присутствуют, то они будут добавлены в утверждение grps.

```
public class AddGroupsToToken implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.flow.dynamic");

    @Override public StrategyBeginState begin(final Context ctx) {
        if ("login".equals(ctx.prompt())){
            List<String> methods = new ArrayList<String>(Arrays.asList(ctx.availableMethods()));
```

```
            methods.remove("cls");
            return StrategyState.MORE(methods.toArray(new String[0]), true);
        } else {
            if(ctx.claims("subjectId") != null)
                return StrategyState.ENOUGH();
            else
                return StrategyState.MORE(new String[]{});
        }
    }

    @Override public StrategyState next(final Context ctx) {
        Integer reqFactor = (ctx.user() == null) ? null : ctx.user().requiredFactor();
        if(reqFactor == null || reqFactor == ctx.justCompletedFactor()) {
            List<String> grps = new ArrayList<String>();
            int groupListIdx = 0;
            while (groupListIdx > -1) {
              String group = ctx.claims("memberOf.[" + groupListIdx + "]");
              logger.debug("### group [" + groupListIdx + "] = " + group);
              if (group == null) {
                groupListIdx = -1;
              } else {
                grps.add(ctx.claims("memberOf.[" + groupListIdx + "]"));
                groupListIdx ++;
              }
            }
                        LClaimsBuilder  claimsBuilder = ctx.claimsBuilder();
                   if (grps.size() > 0) {
                   claimsBuilder.addClaim("grps", grps);
            }
            LClaims claims = claimsBuilder.build();
            return StrategyState.ENOUGH BUILDER()
                .withClaims(claims)
                .build();
        } else
            return StrategyState.MORE(new String[]{});
    }
}
```

6.2.8. Отображение пользователю объявления при входе

Процедура InfoPipe позволяет с периодичностью в 30 дней показывать пользователю при входе объявления. Перед использованием в процедуру нужно внести 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 в конфигурационном файле тип уведомления – см. п. 16.1.28;
- настроить в сообщениях текст уведомления и названия кнопок – см. п. 16.2.8.

```
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();
    }
}
```

6.2.9. 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 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. п. 16.2.8.

```
public class PipeAttrActAdd implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.flow.dynamic");
    private final static String DOMAIN = "example.com";
    private final static String MOBILE_ATTR = "phone_number";
    private final static String EMAIL_ATTR = "email";
    private final static String COMMON_ATTR = "family_name";
    private final static Integer SKIP_TIME_IN_SEC = 30*86400;
    private final static Integer ACT_TIME_IN_SEC = 30*86400;
    private final static Boolean ASK_AT_1ST_LOGIN = false;

    @Override public StrategyBeginState begin(final Context ctx) {
        if ("login".equals(ctx.prompt())){
            List<String> methods = new ArrayList<String>(Arrays.asList(ctx.availableMethods()));
            methods.remove("cls");
            return StrategyState.MORE(methods.toArray(new String[0]), true);
        } else {
            if(ctx.claims("subjectId") != null)
                return StrategyState.ENOUGH();
            else
                return StrategyState.MORE(new String[]{});
        }
    }

    @Override public StrategyState next(final Context ctx) {
        Instant instant = Instant.now();
        Boolean new_device = false;
        if (ctx.ua().getNewlyCreated() && ctx.justCompletedFactor() == 1 && !ASK_AT_1ST_LOGIN){
            logger.debug("User with sub={} is signing in, pid={}, on a new device",
                ctx.claims("subjectId"), ctx.id());
            new_device = true;
        }
        Integer reqFactor = ctx.user().requiredFactor();
        if(reqFactor == null || reqFactor == ctx.justCompletedFactor()) {
            Enough.Builder en_builder = StrategyState.ENOUGH_BUILDER();
            if (MOBILE_ATTR !=null && !new_device && requireActualizeAttr(MOBILE_ATTR, ctx)) {
                String uri = "https://"+DOMAIN+"/blitz/pipes/attr/act?attr="
                    +MOBILE_ATTR+"&canSkip=true&appId=_blitz_profile&verified=true";
                Set<String> clms = new HashSet<String>(){{
                    add("instanceId");
                    add(MOBILE ATTR);
                }};
                Set<String> scps = new HashSet<String>(){{
                    add("openid");
                }};
                logger.debug("User has no {} or a non-actualzed {}, so opening pipe",
                    MOBILE ATTR, MOBILE ATTR);
                en_builder = en_builder.withPipe(uri, "_blitz_profile", scps, clms);
            } else if (EMAIL_ATTR !=null && !new_device && requireActualizeAttr(EMAIL_ATTR, ctx)) {
                String uri = "https://"+DOMAIN+"/blitz/pipes/attr/act?attr="
                    +EMAIL_ATTR+"&canSkip=true&appId=_blitz_profile&verified=true";
                Set<String> clms = new HashSet<String>(){{
                    add("instanceId");
                    add(EMAIL_ATTR);
                }};
                Set<String> scps = new HashSet<String>(){{
                    add("openid");
                }};
                logger.debug("User has no {} or a non-actualzed {}, so opening pipe",
                    EMAIL_ATTR, EMAIL_ATTR);
                en_builder = en_builder.withPipe(uri, "_blitz_profile", scps, clms);
            } else if (COMMON_ATTR !=null && !new_device &&
                        requireActualizeAttr(COMMON_ATTR, ctx)) {
                String uri = "https://"+DOMAIN+"/blitz/pipes/attr/act?attr="
                    +COMMON_ATTR+"&canSkip=true&appId=_blitz_profile";
                Set<String> clms = new HashSet<String>(){{
                    add("instanceId");
                    add(COMMON_ATTR);
                }};
                Set<String> scps = new HashSet<String>(){{
```

```
                    add("openid");
                }};
                logger.debug("User has no {}, so opening pipe", COMMON ATTR);
                en_builder = en_builder.withPipe(uri, "_blitz_profile", scps, clms);
            }
            return en builder.build();
        } else {
            return StrategyState.MORE(new String[]{});
        }
    }

    private Boolean requireActualizeAttr(final String attrName, final Context ctx) {
        if (attrName.equals(MOBILE_ATTR) && (ctx.passedTrack().startsWith("1:sms") ||
            ctx.passedTrack().endsWith("sms"))) {
            logger.debug("User subjectId = {}, pid = {} used SMS, so no actualization needed",
                ctx.claims("subjectId"), ctx.id());
            return false;
        }
        if (attrName.equals(EMAIL ATTR) && ctx.passedTrack().endsWith("email")) {
            logger.debug(
                "User subjectId = {}, pid = {} used EMAIL while auth, so no actualization needed",
                ctx.claims("subjectId"), ctx.id());
            return false;
        }
        Long skpTime = null;
        Long actTime = null;
        long now = Instant.now().getEpochSecond();
        if (ctx.user().userProps().numProp("pipes.act."+attrName+".skippedOn") != null) {
            skpTime = ctx.user().userProps().numProp("pipes.act."+attrName+".skippedOn");
        }
        if (skpTime != null && ((now - skpTime) < SKIP TIME IN SEC)) {
            logger.debug(
                "User subjectId = {}, pid = {} has skipped update '{}' only '{}' seconds ago, no
actualization needed", ctx.claims("subjectId"), ctx.id(), attrName, (now - skpTime));
            return false;
        }
        if (ctx.claims(attrName) == null) return true;
        else {
            if (ctx.user().attrsCfmTimes() != null) {
                actTime = ctx.user().attrsCfmTimes().get(attrName);
            }
            if (actTime == null) return true;
            else {
                logger.debug(
                    "User subjectId = {}, pid = {} has updated '{}' '{}' seconds ago, actualization
needed = {}", ctx.claims("subjectId"), ctx.id(), attrName, (now - actTime), ((now - actTime) >
ACT TIME IN SEC));
                return ((now - actTime) > ACT_TIME_IN_SEC);
            }
        }
    }
}
```

6.2.10. 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
                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();
    }
}
```

6.2.11. 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 сделана настройка, чтобы

пользователю при первом входе не предлагали сразу заполнить данные);

- в теле процедуры вместо _blitz_profile указать идентификатор другого приложения, если изменение атрибутов должно делаться от приложения, отличного от Личного кабинета.

```
public class PipeWebAuthn implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.flow.dynamic");
    private final static String DOMAIN = "example.com";
    private final static Integer SKIP_TIME_IN_SEC = 30*86400;
    private final static Boolean ASK_AT_1ST_LOGIN = true;

    @Override public StrategyBeginState begin(final Context ctx) {
        if ("login".equals(ctx.prompt())){
            List<String> methods = new ArrayList<String>(Arrays.asList(ctx.availableMethods()));
            methods.remove("cls");
            return StrategyState.MORE(methods.toArray(new String[0]), true);
        } else {
            if(ctx.claims("subjectId") != null)
                return StrategyState.ENOUGH();
            else
                return StrategyState.MORE(new String[]{});
        }
    }

    @Override
    public StrategyState next(Context ctx) {
        Boolean new_device = false;
        if (ctx.ua().getNewlyCreated() && ctx.justCompletedFactor() == 1 && !ASK_AT_1ST_LOGIN){
            logger.debug("User with sub={} is signing in, pid={}, on a new device",
                ctx.claims("subjectId"), ctx.id());
            new_device = true;
        }
        if (ctx.user() == null || ctx.user().requiredFactor() == null ||
            ctx.user().requiredFactor().equals(ctx.justCompletedFactor()))
            if (!new device && requiredWebAuthn(ctx))
                return webAuthn(ctx);
            else
                return StrategyState.ENOUGH();
        else
            return StrategyState.MORE(new String[] {});
    }

    private boolean requiredWebAuthn(final Context ctx) {
        LBrowser br = ctx.ua().asBrowser();
        String deviceType = br.getDeviceType();
        String os = br.getOsName();
        List<WakMeta> keyList = null;
        logger.trace("User subjectId = {}, pid = {} is logging using device '{}' and OS '{}',
checking configured webAuthn keys", ctx.claims("subjectId"), ctx.id(), deviceType, os);
        ListResult<WakMeta> keys = ctx.dataSources().webAuthn().keysOfCurrentSubject();
        if (keys != null) {
            keyList = keys.filter(k -> deviceType.equals(k.addedOnUA().deviceType()))
                .filter(k -> os.equals(k.addedOnUA().osName())).list();
        }
        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();
    }
}
```

6.2.12. Отображение пользователю списка выбора значений при входе

Процедура ChoicePipe позволяет показывать пользователю при входе страницы выбора списка значений. Перед использованием в процедуру нужно 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 – см. п. 16.1.28;
- настроить в сообщениях текст уведомления и названия кнопок – см. п. 16.2.8.

```
public class ChoicePipe implements Strategy {

    private final Logger logger = LoggerFactory.getLogger("com.identityblitz.idp.flow.dynamic");

    private final static String DOMAIN = "<BLITZ-HOST>";
    private final static String CLIENT_ID = "<CLIENT_ID>";

    @Override public StrategyBeginState begin(final Context ctx) {
        if ("login".equals(ctx.prompt())){
            List<String> methods = new ArrayList<String>(Arrays.asList(ctx.availableMethods()));
            methods.remove("cls");
            return StrategyState.MORE(methods.toArray(new String[0]), true);
        } else {
            if(ctx.claims("subjectId") != null)
                return StrategyState.ENOUGH();
            else
                return StrategyState.MORE(new String[]{});
        }
    }

    @Override
    public StrategyState next(Context ctx) {
        List<List<String>> choice = new ArrayList<List<String>>(){};
        choice.add(Arrays.asList("Value 1"));
        choice.add(Arrays.asList("Value 2"));
        try {
            if (ctx.user() == null || ctx.user().requiredFactor() == null
                    || ctx.user().requiredFactor().equals(ctx.justCompletedFactor())) {
                String res = new ObjectMapper().writeValueAsString(choice);
                String choiceJson = Base64.getUrlEncoder().encodeToString(res.getBytes("UTF-8"));
                return choice(ctx, choiceJson);
            }
            else
                return StrategyState.MORE(new String[] {});
```

```
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }

    private StrategyState choice(final Context ctx, final String choiceJson) {
        String uri = "https://" + DOMAIN + "/blitz/pipes/choice/start?appId=" + CLIENT ID +
"&pipeId=select_value&choices=" + choiceJson;
        Set<String> claims = new HashSet<String>(){{
            add("instanceId");
        }};
        Set<String> scopes = new HashSet<String>(){{
            add("openid");
        }};
       return StrategyState.ENOUGH_BUILDER()
         .withPipe(uri, CLIENT_ID, scopes, claims)
         .build();
    }
}
```

6.2.13. Получение геоданных пользователя

Процедуру входа можно использовать для получения данных о стране и городе, в которых находится пользователь, и на основании этого гибко настраивать правила входа, например, вводить запрет на вход из-за рубежа, активировать второй фактор аутентификации и др.

Для этого в процедурах входа используйте следующие классы и методы:

1) Класс **LGeoData** с функциями **getCountry()** и **getCity()**.

```
public class LGeoData {
    /**
    *  Get IP address country
    *
    * @return - country or null if country not specified.
    */
    public final String getCountry();
    /**
    *  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();
```

Для работы метода необходимо импортировать класс **LGeoData**.

```
import com.identityblitz.idp.login.authn.flow.LGeoData
```

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());
```

6.3. Кастомизация логики операций с хранилищами данных

6.3.1. 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

6.3.2. 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.

2) Откройте файл конфигурации **/usr/share/identityblitz/blitz-config/blitz.conf**.

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

3) Добавьте новый блок **logic** в блок конфигурации **blitz.prod.local.idp.id-stores**. Блок должен содержать имена кастомизируемых операций, указанных в качестве ключа и секцию **{ "enabled": true}** в качестве значения ключа.

Пример кастомизации операций searchUsers и bindUser

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

```
    }
}
```

6.3.3. 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.

7. Настройка сервисов самообслуживания пользователей

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. При включении доступно 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.

Администратор консоли управления должен самостоятельно проверять корректность помещаемых на страницу входа JS-скриптов и содержимое страниц регистрации и личного кабинета на предмет возможных уязвимостей.

7.1. 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.

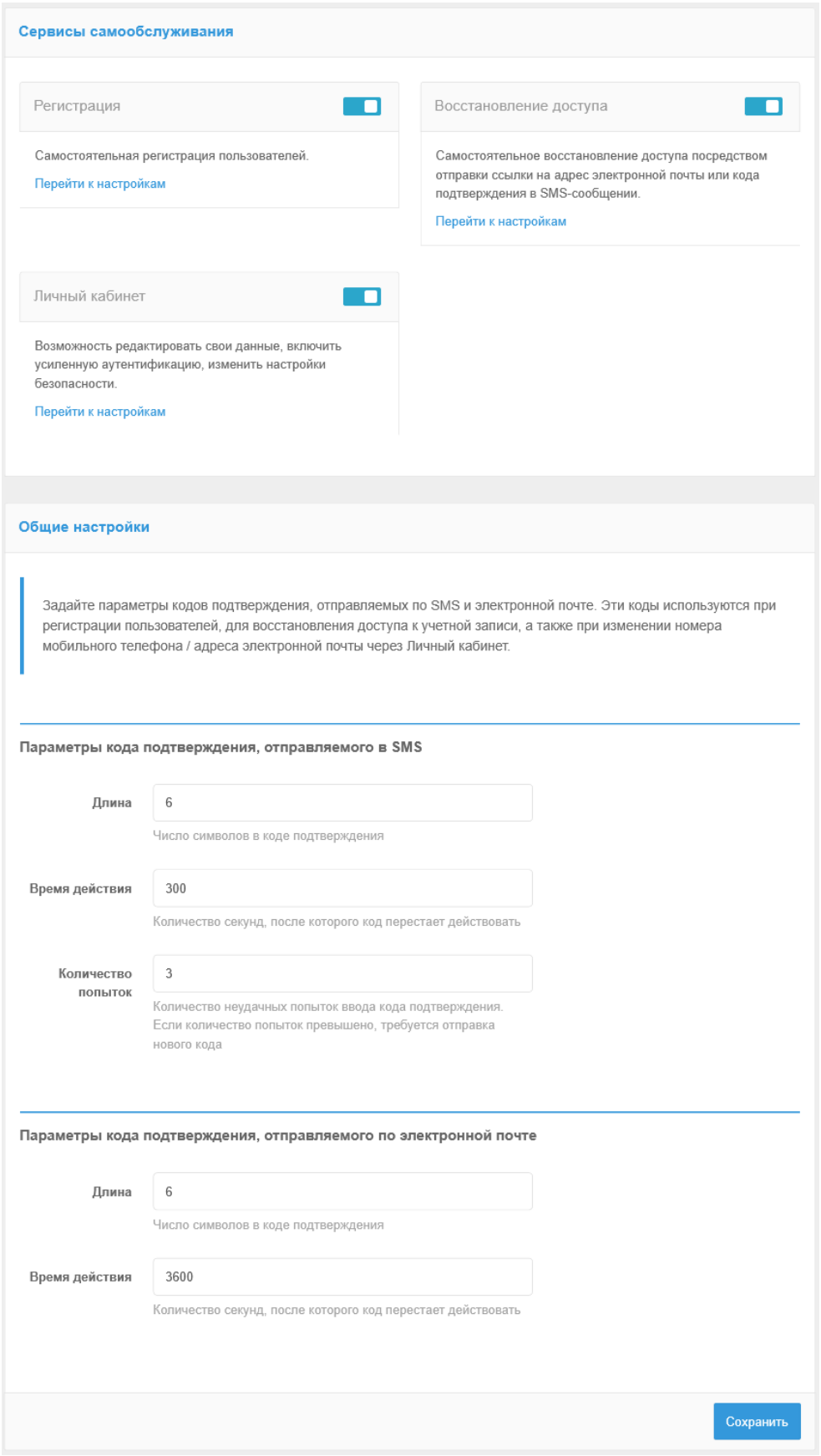

Рисунок 91 – Сервисы самообслуживания и их общие настройки

В подразделах осуществляется настройка каждого сервиса самообслуживания в отдельности.

7.2. 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).

7.2.1. Форма регистрации

Перечень запрашиваемых данных пользователя определяется HTML-шаблоном. Шаблон представляет собой текстовый файл, который компилируется с помощью шаблонизатора Twirl. В шаблоне необходимо разместить функции, позволяющие пользователю при регистрации вводить данные о себе.

Примеры функций, доступных в шаблоне:

- @attrInput("email", msg("reg.email"), Map("placeholder" -> "mail@example.com", "error-messages" -> msg("reg.email.wrong"), "input-type" -> "mail")) – отображает на странице поле для ввода атрибута email, описанного в системе. msg("reg.email") - это название атрибута, которое берется из файла сообщений в соответствии с текущими языковыми настройками. При пустом поле ввода в нем отображается "mail@example.com" в качестве подсказки, а при некорректном вводе – сообщение msg("reg.email.wrong") из файла сообщений. Для элемента задается input-type равный mail;
- @attrInput("family_name", "Фамилия", Map("placeholder" -> "Фамилия", "error-messages" -> "Ошибка")) – отображает на странице поле для ввода фамилии пользователя в переменную family_name. Эту переменную далее можно использовать при исполнении процедуры регистрации.
- @securityQuestionInput – отображает на странице поля ввода контрольного вопроса и ответа на контрольный вопрос;
- @passwordsInput – отображает на странице поля ввода пароля и его подтверждение;
- @agreement – отображает ссылку на условия использования;
- @attrExpr – функция, позволяющая создать вычисляемый атрибут (или присвоить атрибуту константное значение);
- @submitButton – отображает кнопку «Зарегистрироваться».

Пример шаблона для регистрации:

```
@attrInput("family_name", "Фамилия", Map("placeholder" -> "Фамилия", "error-messages" -> "Ошибка"))
@attrInput("given_name", "Имя", Map("placeholder" -> "Имя", "error-messages" -> "Ошибка"))
@attrInput("phone number", "Номер мобильного телефона", Map("placeholder" -> "+7(999)9999999",
"error-messages" -> "reg.page.mobile.req.err.msg"))
@attrInput("email", "Адрес электронной почты", Map("placeholder" -> "name@example.com", "error-
messages" -> "reg.page.email.req.err.msg", "input-type" -> "mail"))
@passwordsInput
@agreement
@attrExpr("sub","BIP-${&random(4)}")
```

```
@submitButton
```

Для автогенерации GUID создаваемых учетных записей используйте следующую формулу @attrExpr:

```
@attrExpr("sub","${&rUUID()}")
```

Пример отображения указанного шаблона в интерфейсе веб-приложения «Регистрация пользователя» представлен на рис. 92.

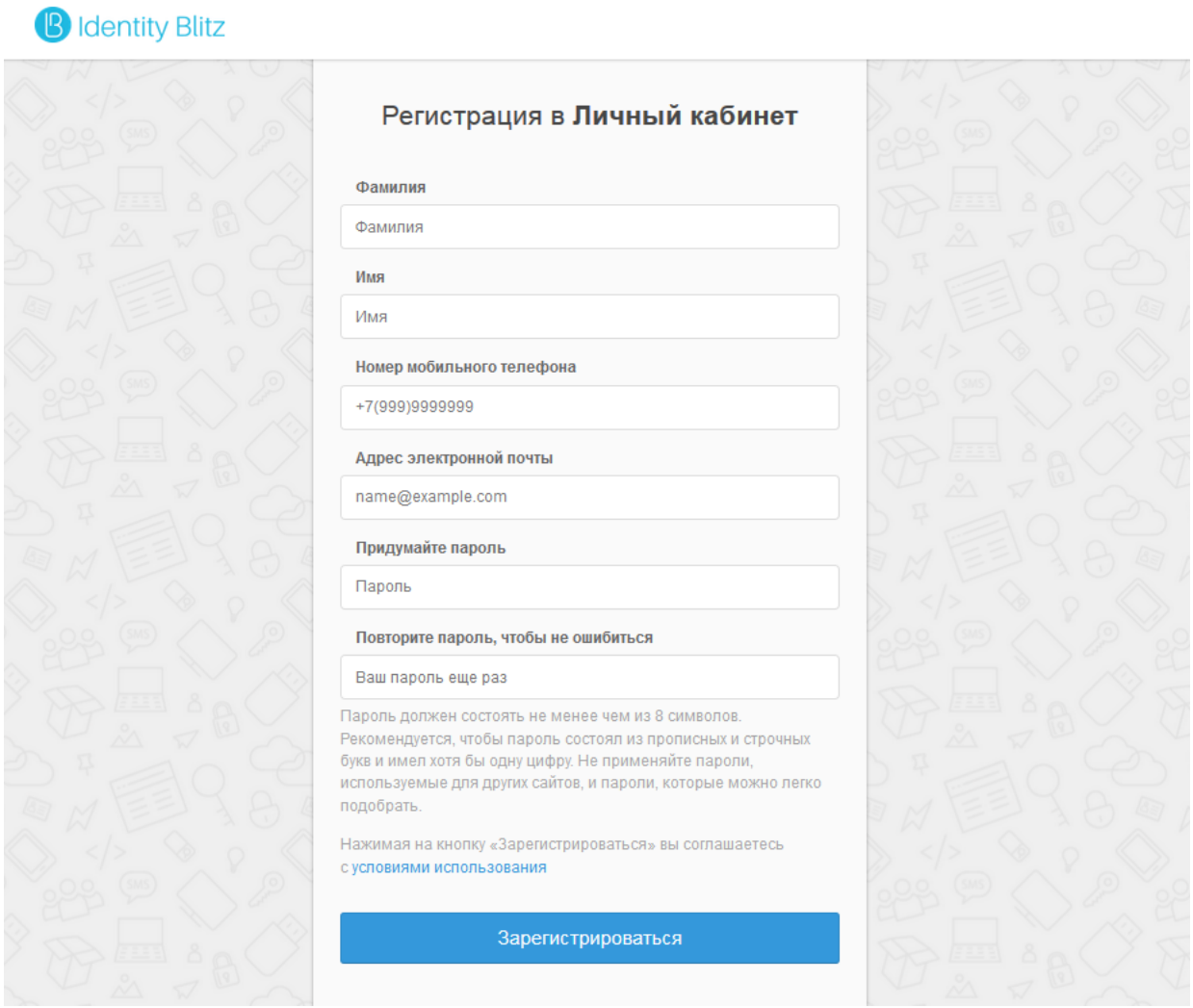

Рисунок 92 – Пример отображения регистрационной формы

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"))
```

7.2.2. 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 рис. 93.

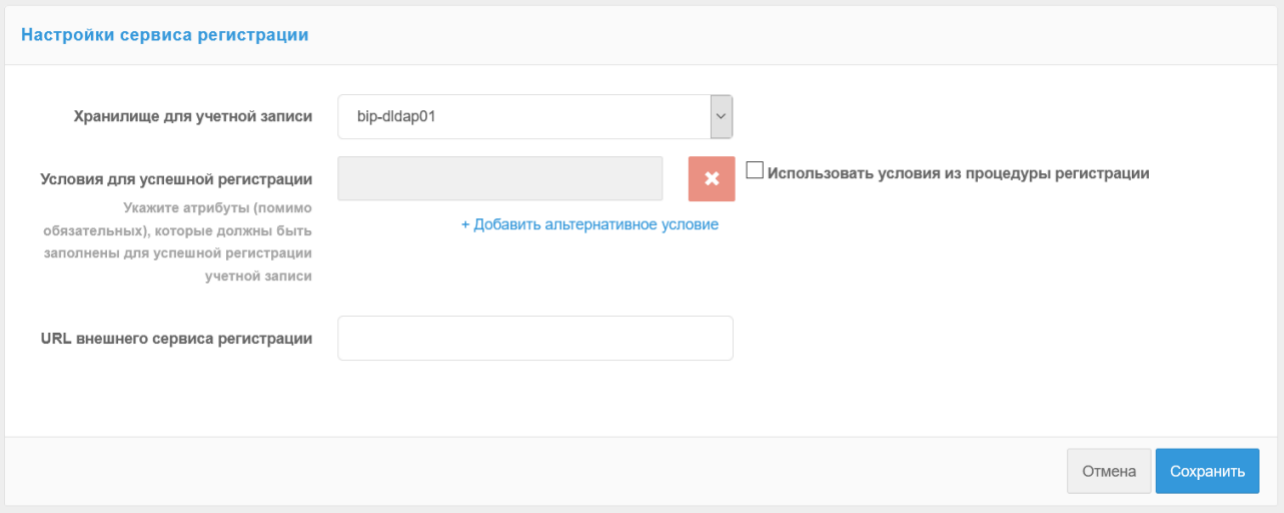

Рисунок 93 – Скриншот настроек сервиса регистрации

7.2.3. 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.

7.2.4. 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 строку reg.page.reg.action.agreement и setPswd.page.agreement (см.п. 16.2.2). Такой способ рекомендуется применять, если правила использования размещены на внешнем ресурсе, например, в виде отдельной веб-страницы.

7.3. 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.

7.3.1. Отображение атрибутов пользователя

На основной странице Личного кабинета отображается блок с данными учетной записи. Пример этого блока представлен на рис. 94.

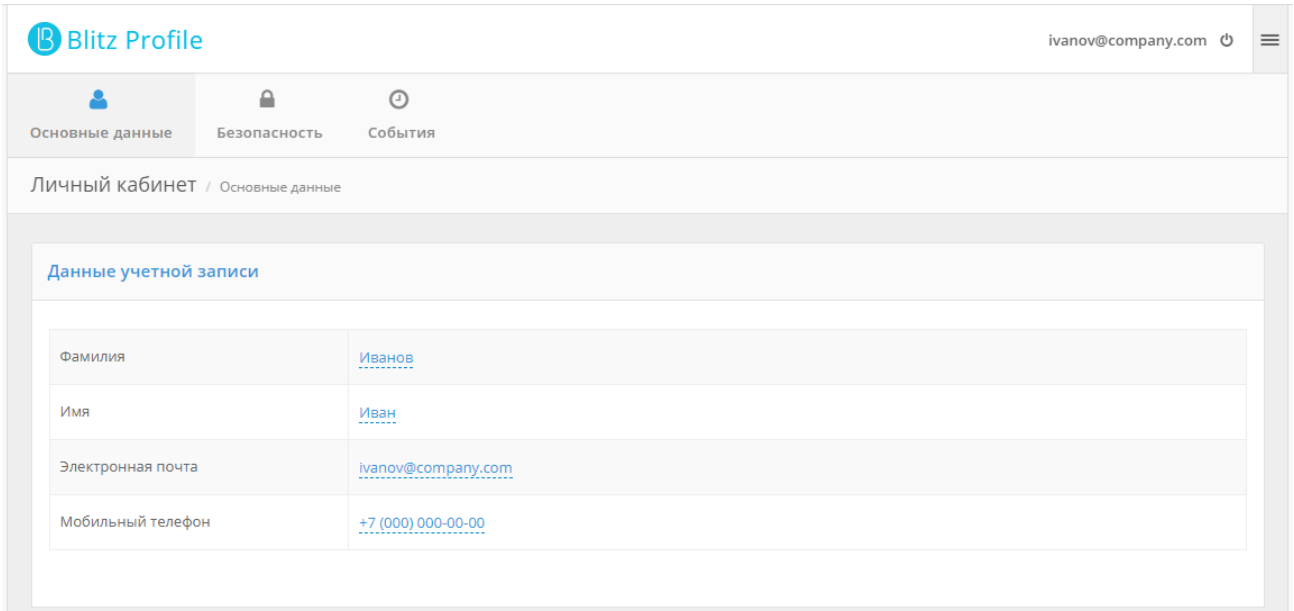

Рисунок 94 – Личный кабинет: данные учетной записи

Отображение данных пользователя определяется HTML-шаблоном. Шаблон представляет собой текстовый файл, который компилируется с помощью шаблонизатора

Twirl[35]. 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 «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. п. 16.2.2 документа. Рекомендуется использовать файлы сообщений при необходимости поддержки мультиязычности.

Примеры функций:

Отображение атрибута email:

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

Отображение атрибута phone_number с возможностью его редактировать:

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

Отображение булевого атрибута info с возможностью его редактировать:

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

Отображение массива строк subscription_array с возможностью его редактировать (выбор значений):

```
@editAsStrings("subscription_array", "Подписки", ListMap("a" -> "Акции и бонусные программы", "b" -> "Новости компании", "c" -> "Дайджест событий за месяц"))
```

Пример отображения массива строк в интерфейсе Личного кабинета представлен на рис. 95.

[35] См.: https://github.com/playframework/twirl

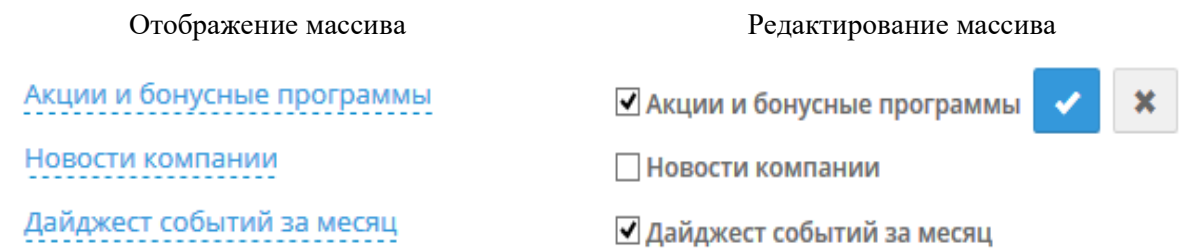

Рисунок 95 – Личный кабинет: массив строк (в режиме отображения и редактирования)

7.3.2. Дополнительные параметры

В качестве дополнительных параметров (см. Рисунок 96) 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. п. 16.1.27). 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.

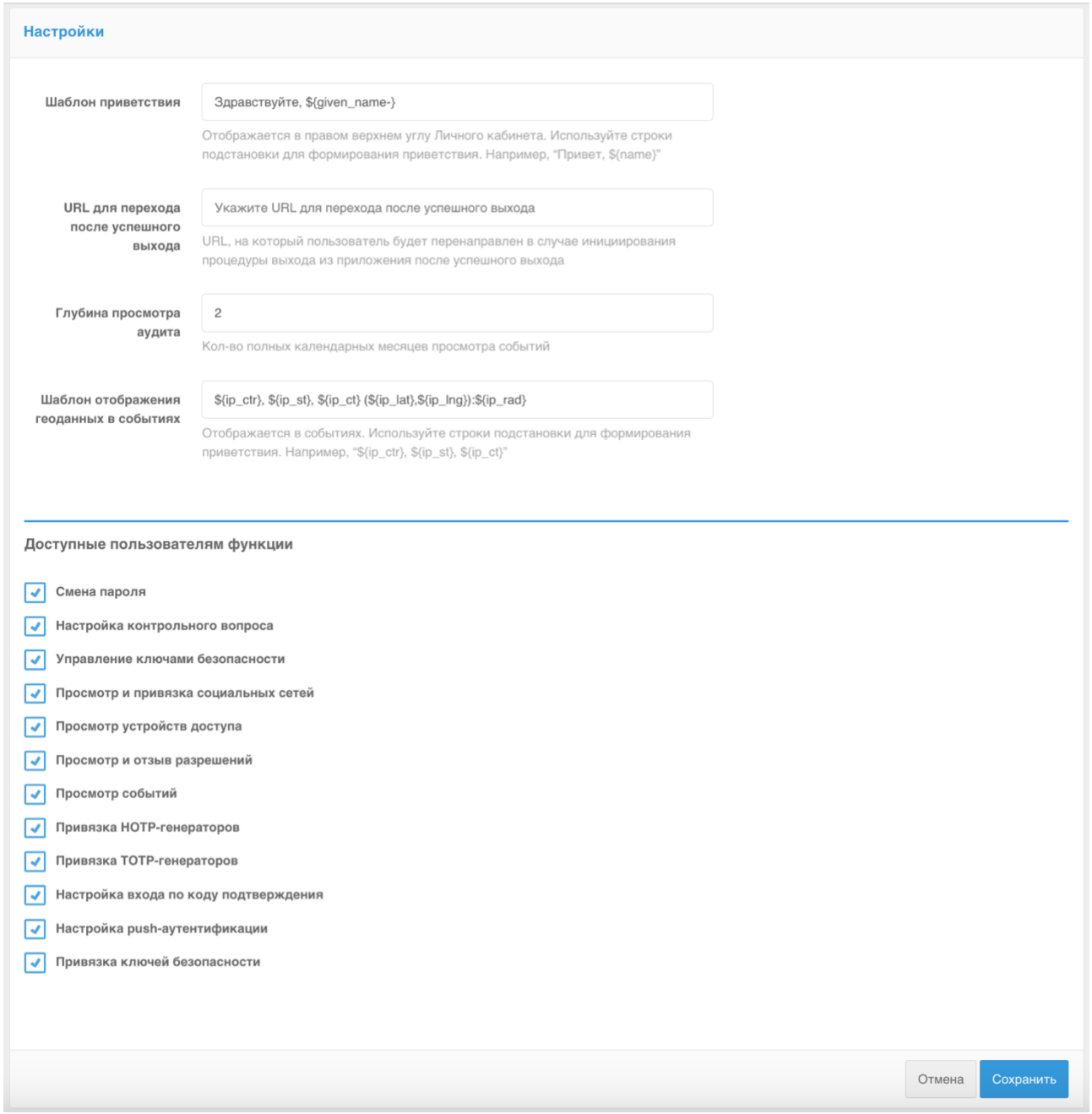

Рисунок 96 – Настройка дополнительных параметров Личного кабинета

7.4. Восстановление доступа

Настройка «Возможные атрибуты для поиска» сервиса восстановления доступа определяет атрибуты, по которым будет производиться поиск учетной записи (см. Рисунок 97).

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.

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.

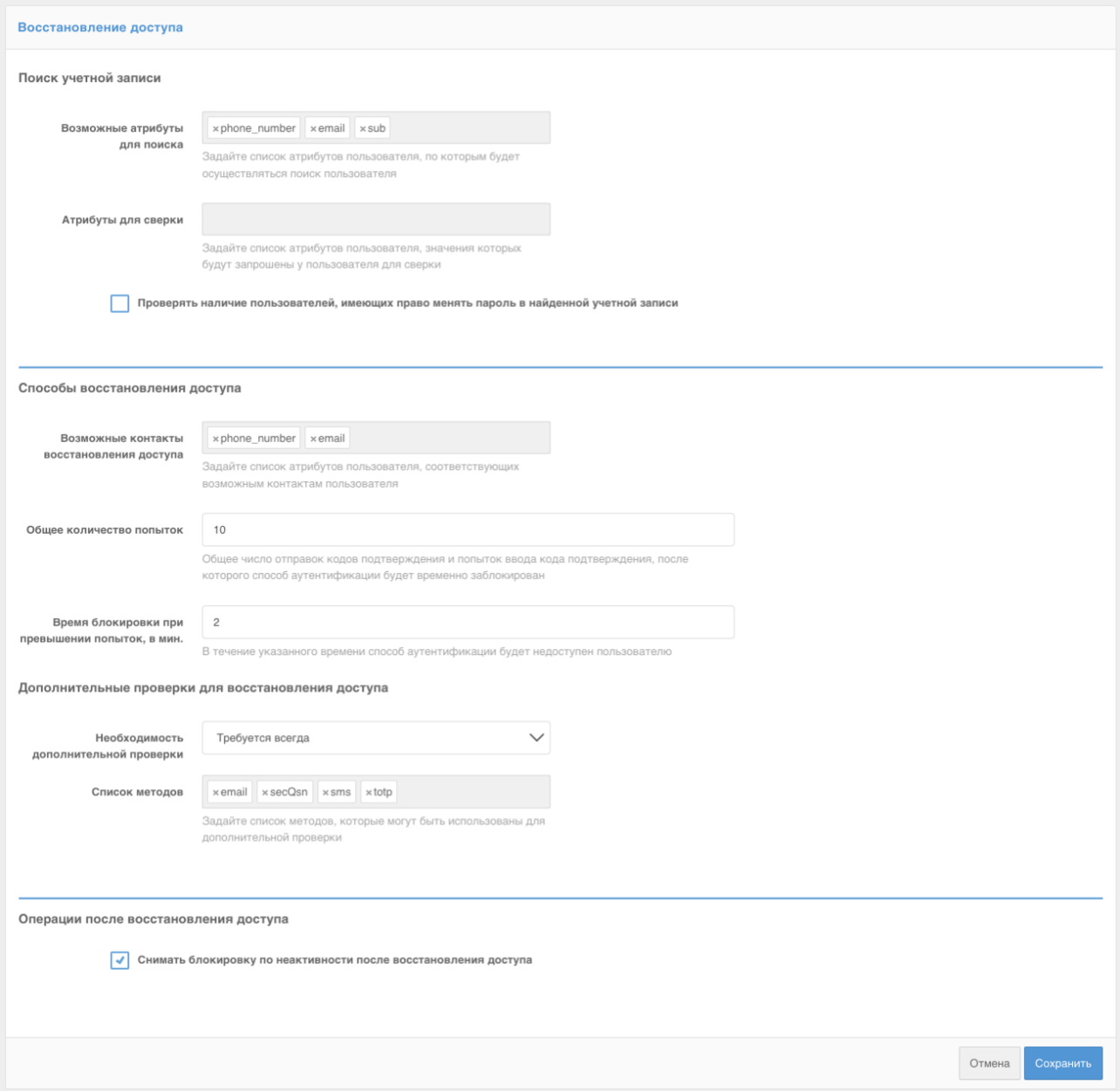

Рисунок 97 – Восстановление доступа

После определения набора атрибутов для сверки необходимо задать соответствующие им тексты в форме восстановления доступа. Для этого используйте стандартный алгоритм из п. 16.2.2. 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.

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
```

8. Вход через внешние поставщики идентификации

Настройка входа через внешние поставщики идентификации включает в себя следующие шаги:

1. Выполнить настройки в разделе «Поставщики идентификации» в консоли управления Blitz Identity Provider.
2. Выполнить настройки на стороне поставщика идентификации.
3. Включить возможность входа через данный поставщик идентификации в разделе «Аутентификации» (см. п. 4.6).

Для настройки используется раздел «Поставщики идентификации» в консоли управления. Начальный экран показывает настроенные поставщики и позволяет выбрать для настройки требуемый тип поставщика идентификации (см. Рисунок 98).

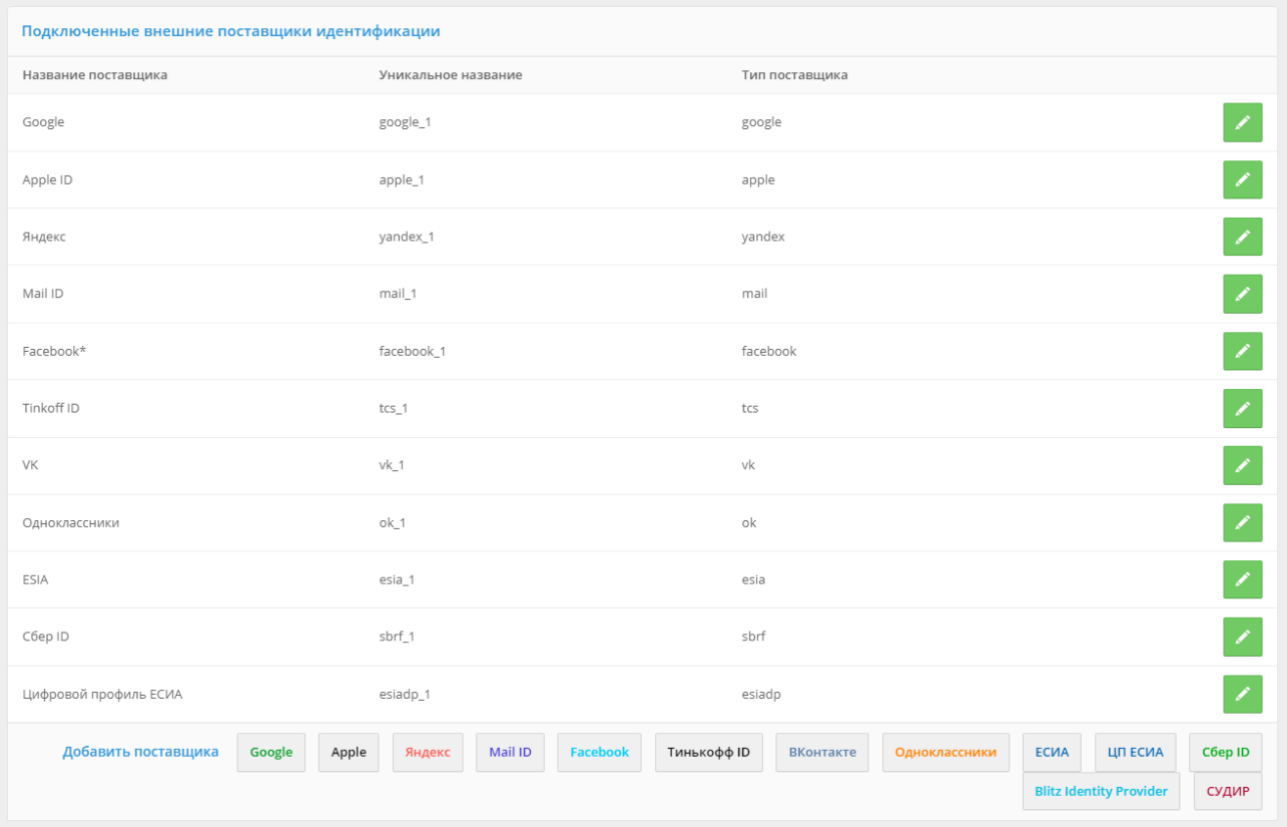

Рисунок 98 – Настройка поставщиков идентификации

Настройка поставщика идентификации состоит из следующих шагов:

1. Задание идентификатора поставщика и имени поставщика (см. Рисунок 99):

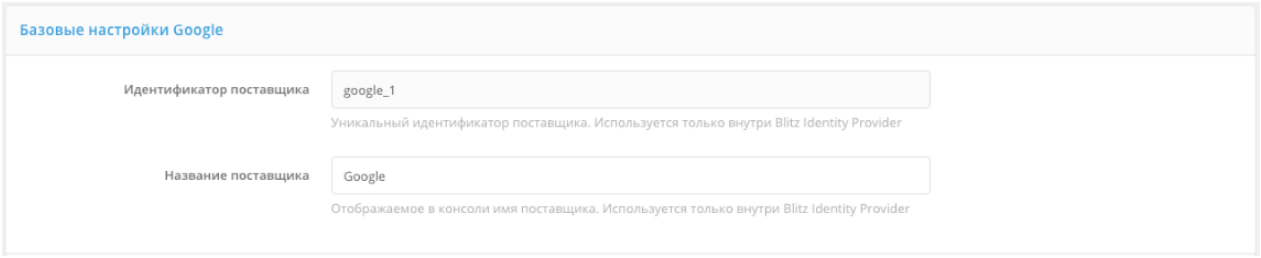

Рисунок 99 – Задание идентификатора и названия поставщика (на примере Google)

2. Задание настроек подключения к поставщику (описаны в п. 8.1 – п. 8.17 по каждому из поставщиков идентификации).
3. Задание настроек связывания учетной записи внешнего поставщика идентификации и учетной записи Blitz Identity Provider. Эти настройки не имеют специфики, зависящей от типа поставщика, и описаны в п. 8.18.

8.1. Вход через Apple ID

Для конфигурирования входа через Apple ID необходимо перейти в «Apple Developer Account»[36] в раздел «Certificates, Identifiers & Profiles» (см. Рисунок 100)[37], в котором выполнить следующие операции:

1. В окне «Certificates, Identifiers & Profiles» выбрать в правом верхнем углу фильтр «App IDs». Кнопкой «+» рядом с «Identifiers» создать новый «App ID» (см. Рисунок 101):
   - выбрать тип App;
   - задать «Description». Он будет отображаться пользователю в окне подтверждения входа по Apple ID;
   - в «Bundle ID» задать идентификатор вида com.company.login на основе используемого в Blitz Identity Provider домена;
   - в «Capabilities» отметить «Sign In with Apple», нажать рядом кнопку Edit и проверить, что выбрано «Enable as a primary App ID»;
   - будет предложено завершить настройку – это все описано в последующих пунктах. Пока нужно нажать «Register».

[36] У компании должна быть действующая подписка Apple Developer ID.
[37] См.: https://developer.apple.com/account/resources/identifiers/list

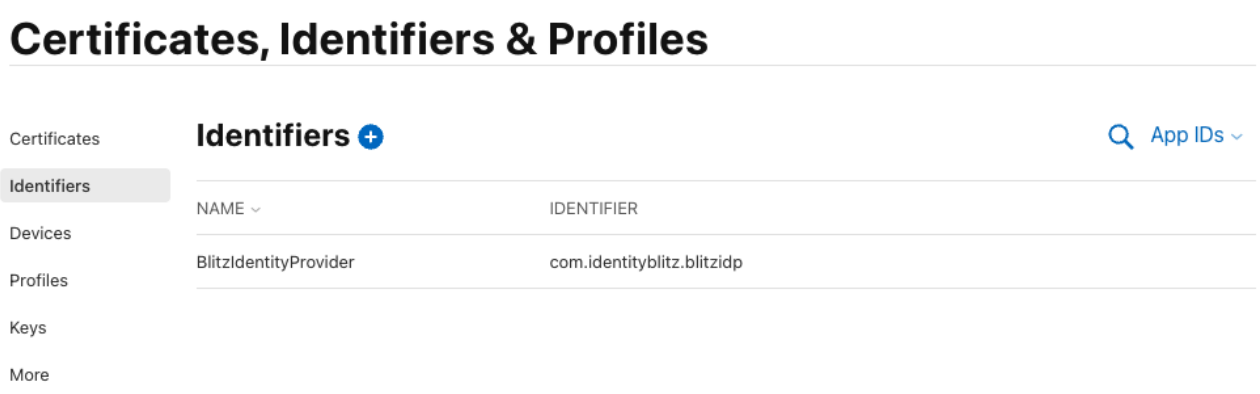

Рисунок 100 – Отображение списка App IDs

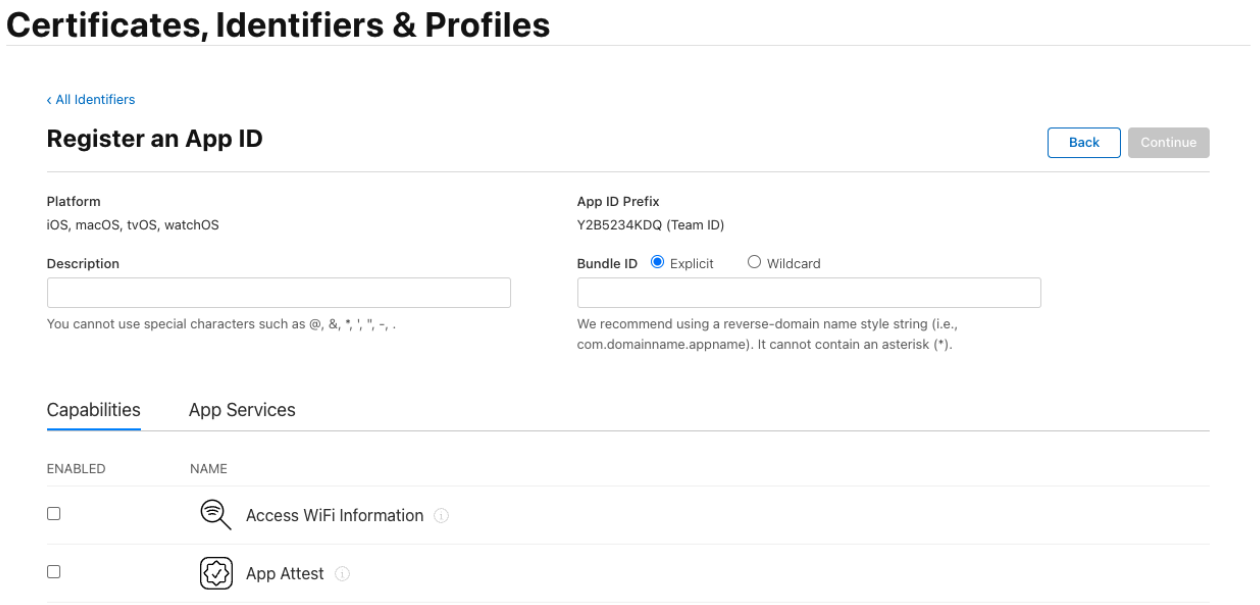

Рисунок 101 – Регистрация нового App ID

2. В окне «Certificates, Identifiers & Profiles» выбрать в правом верхнем углу фильтр «Services App IDs». Кнопкой «+» рядом с «Identifiers» создать новый «Services App ID» (см. Рисунок 102):
   - задать «Description». Он будет отображаться пользователю в окне подтверждения входа по Apple ID;
   - задать «Identifier». Рекомендуется задать в виде com.company.login на основе используемого в Blitz Identity Provider домена. Позже созданный Identifier нужно будет ввести в настройках Blitz Identity Provider в качестве client_id в настройку «Идентификатор клиента (Service ID)»;
   - нажать «Register»;
   - выбрать созданный «Service ID». В его настройках поставить чекбокс «ENABLED» и нажать «Configure» (см. Рисунок 103);
   - в открывшемся окне проверить, что в «Primary App ID» указывается созданный

ранее «App ID» (см. Рисунок 104);

- в «Domains and Subdomains» через запятую перечислить домены, используемые Blitz Identity Provider;
- в «Return URLs» перечислить URL возврата через запятую и с указанием https. Нужно указать URL, образцы которых Blitz Identity Provider показывает в настройках подключения Apple ID (см. Рисунок 106), например:

```
https://login.company.com/blitz/login/externalIdps/callback/apple/apple 1/false
https://login.company.com/blitz/profile/social/externalIdps/callbackPopup/apple/apple_1
```

- подтвердить задание настроек, нажав последовательно «Confirm», «Done», «Continue», «Save» в последующих экранах.

3. В меню «Keys» (см. Рисунок 105) создать ключ для «Sign In with Apple». Это можно сделать только однократно, так что рекомендуется созданный ключ где-то сохранить. В Blitz Identity Provider в настоящий момент этот ключ не используется и не понадобится, но он должен быть создан и сохранен на будущее.

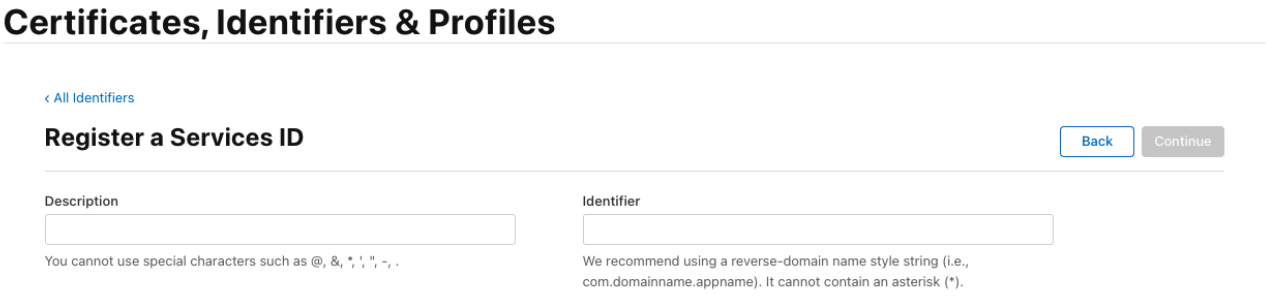

Рисунок 102 – Регистрация нового Services App ID

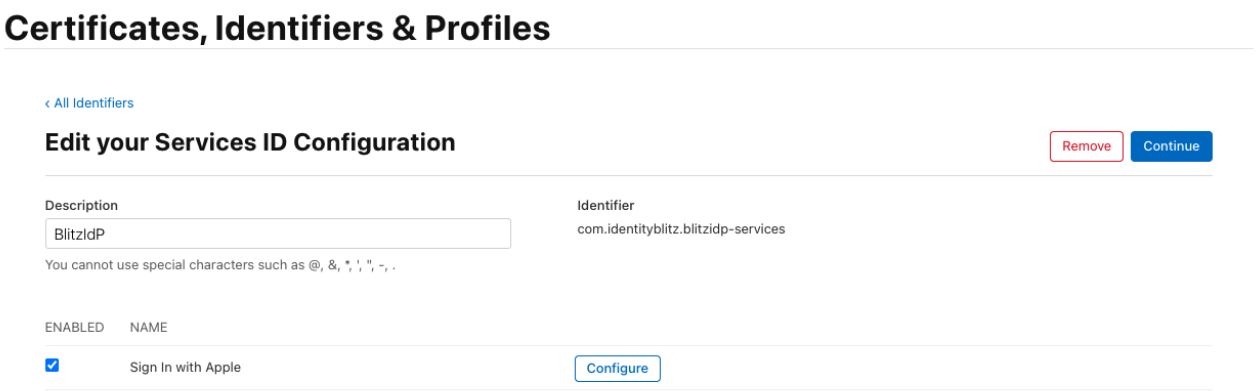

Рисунок 103 – Редактирование конфигурации Services App ID

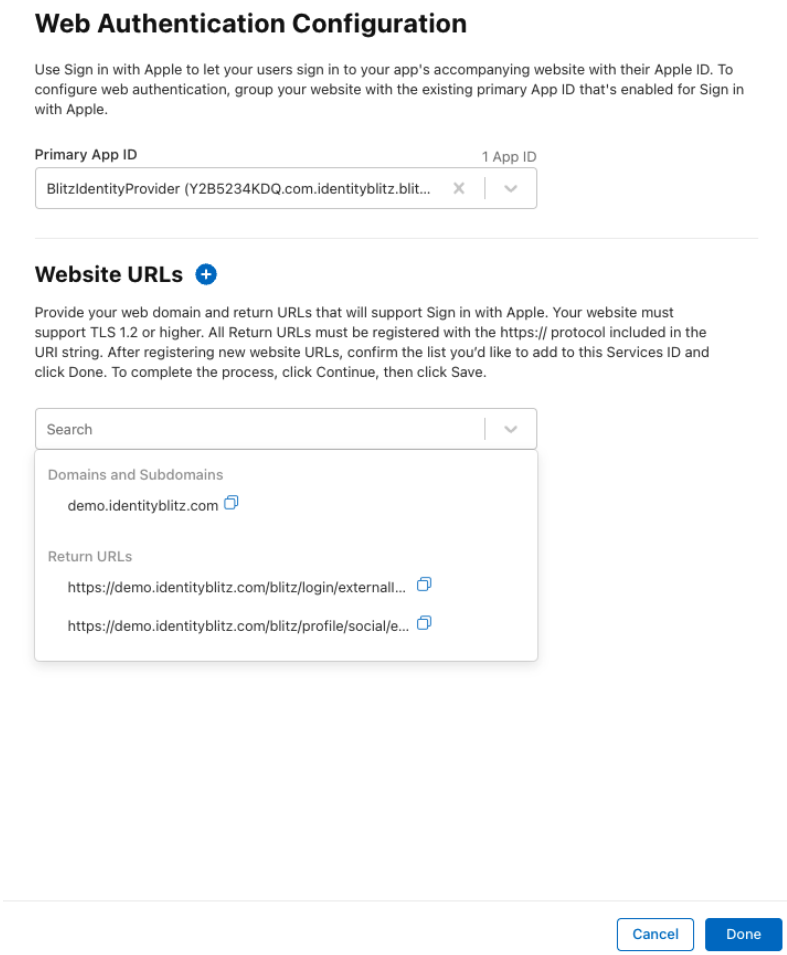

Рисунок 104 – Регистрация доменов и разрешенных адресов возврата для Services App ID

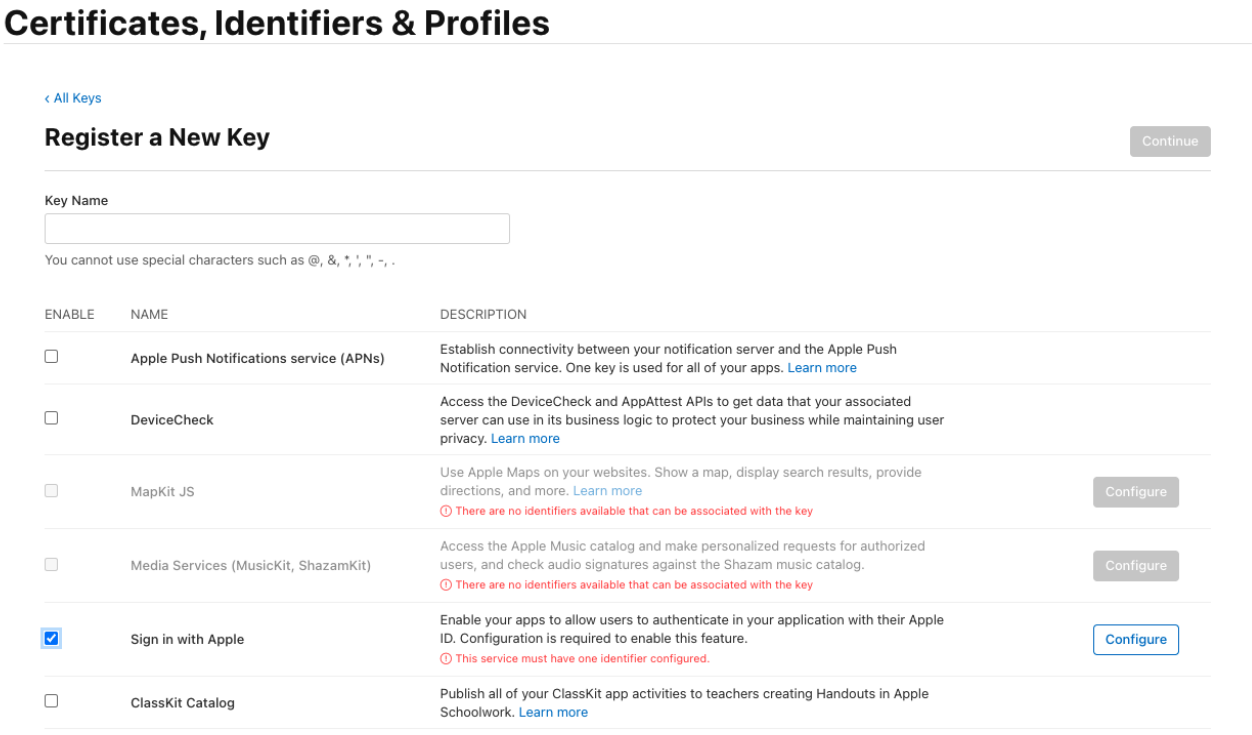

Рисунок 105 – Регистрация ключа

После окончания настроек в «Apple Developer Account» необходимо:

1. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип Apple
2. Заполнить настройки поставщика идентификации (см. Рисунок 106):
   - Идентификатор поставщика;
   - Название поставщика;
   - Идентификатор клиента (Service ID), полученный в Apple Developer Account;
3. Настроить правила связывания (см. п. 8.18).
4. В разделе «Аутентификация» консоли управления включить использование метода аутентификации с использованием поставщика идентификации Apple (см. п. 4.6).

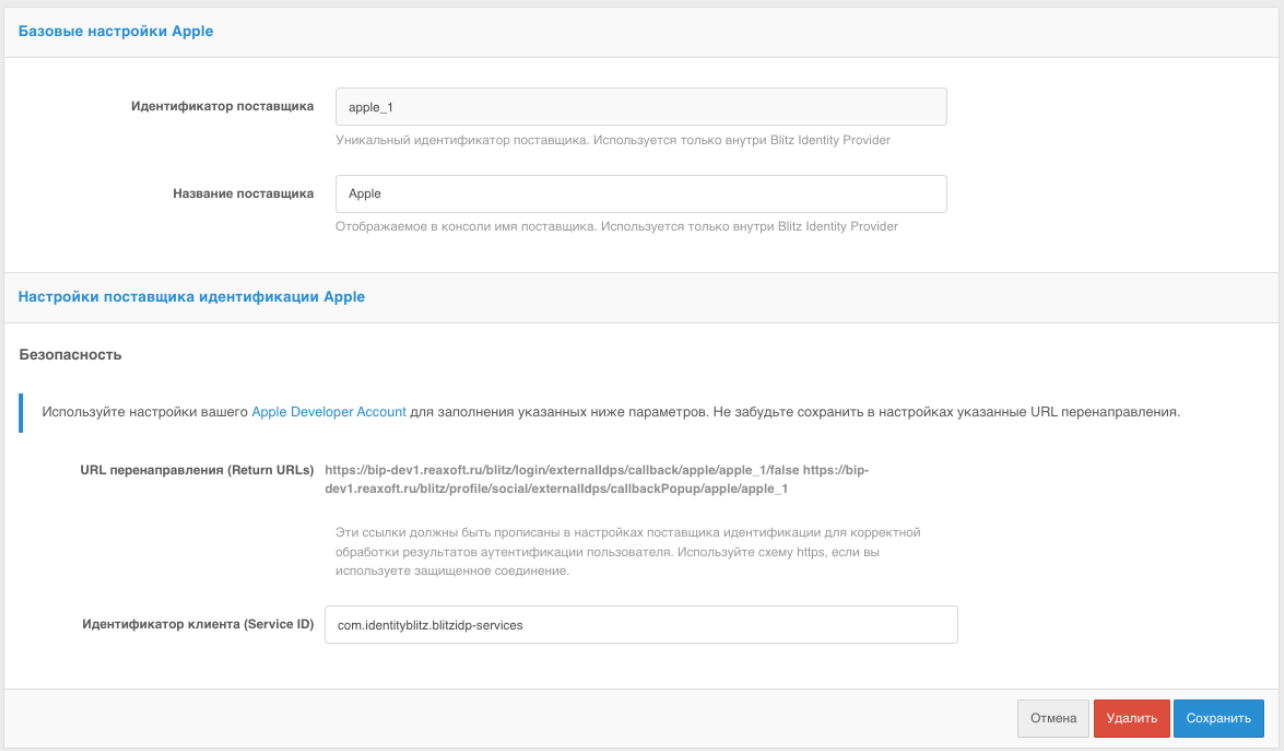

Рисунок 106 – Настройки поставщика идентификации Apple ID

8.2. Вход через Google

Для конфигурирования входа через учетную запись Google необходимо выполнить следующие настройки:

1. Перейти в «Диспетчер API Google» (см. Рисунок 107)[38], в котором выполнить следующие операции:
   - перейти в раздел «Учетные данные»;
   - создать проект и создать новые учетные данные типа «Идентификатор клиента

[38] См.: https://console.developers.google.com

OAuth»;

- выбрать тип нового идентификатора клиента (например, веб-приложение) и дать ему название;
- ограничения не задавать, они будут указаны позже;
- Google сгенерирует идентификатор и секрет клиента, они потребуются для последующего ввода в консоли управления Blitz Identity Provider.
- в «Разрешенные URI перенаправления» перечислить URL возврата через запятую и с указанием https. Нужно указать URL, образцы которых Blitz Identity Provider показывает в настройках подключения Google (см. Рисунок 108), например:

```
https://login.company.com/blitz/login/externalIdps/callback/google/google 1/false
https://login.company.com/blitz/profile/social/externalIdps/callbackPopup/google/google 1
```

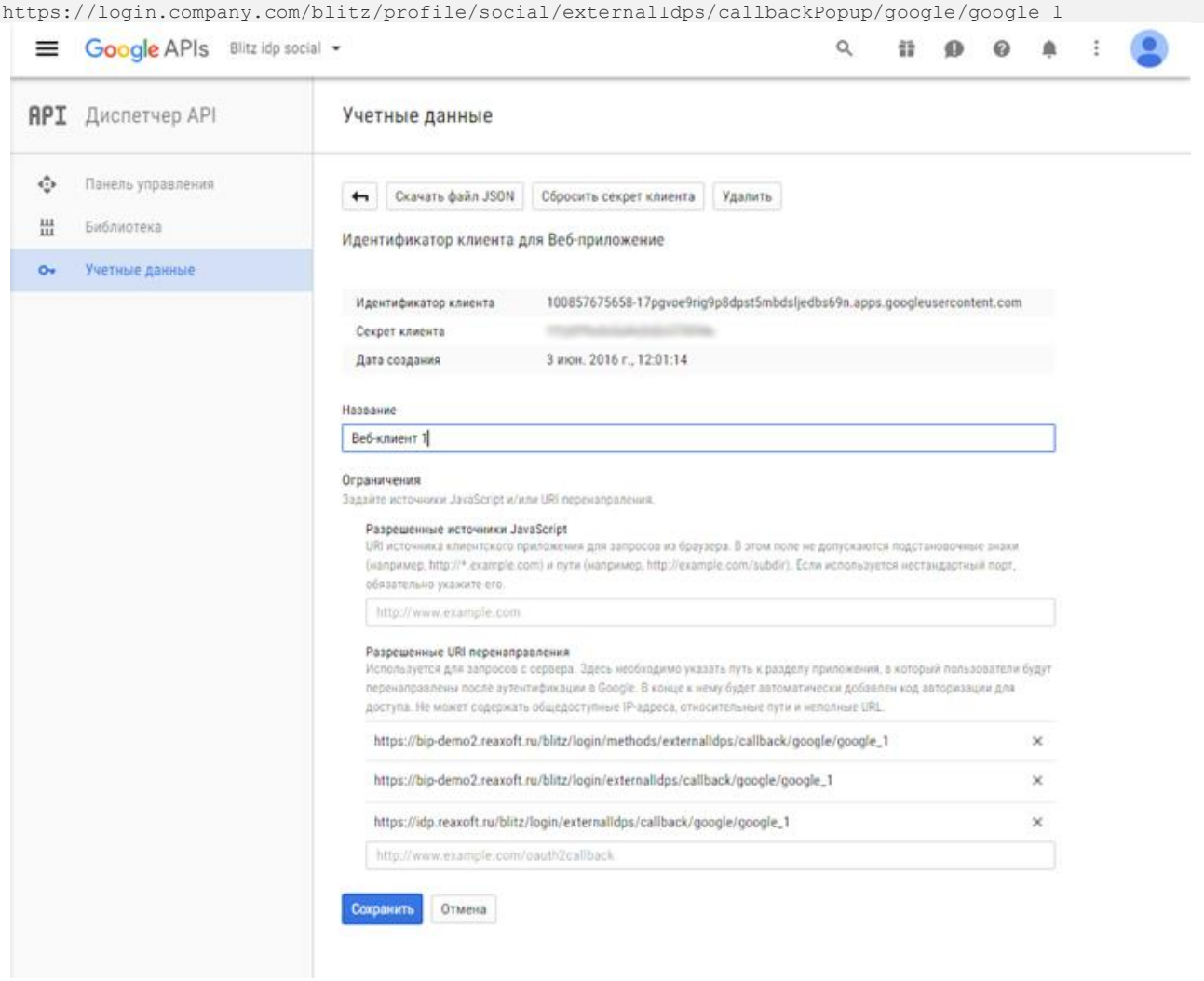

Рисунок 107 – Настройки в Диспетчере API Google

2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип Google.
3. Заполнить настройки поставщика идентификации (см. Рисунок 108):
   - Идентификатор поставщика;
   - Название поставщика;

- Идентификатор клиента (Client ID), полученный в Диспетчере API Google;
- Секрет клиента (Client secret), полученный в Диспетчере API Google;
- Запрашиваемые разрешения (scope), предусмотренные в Google[39].

4. Настроить правила связывания (см. п. 8.18).
5. В разделе «Аутентификация» консоли управления включить использование метода аутентификации с использованием поставщика идентификации Google (см. п. 4.6).

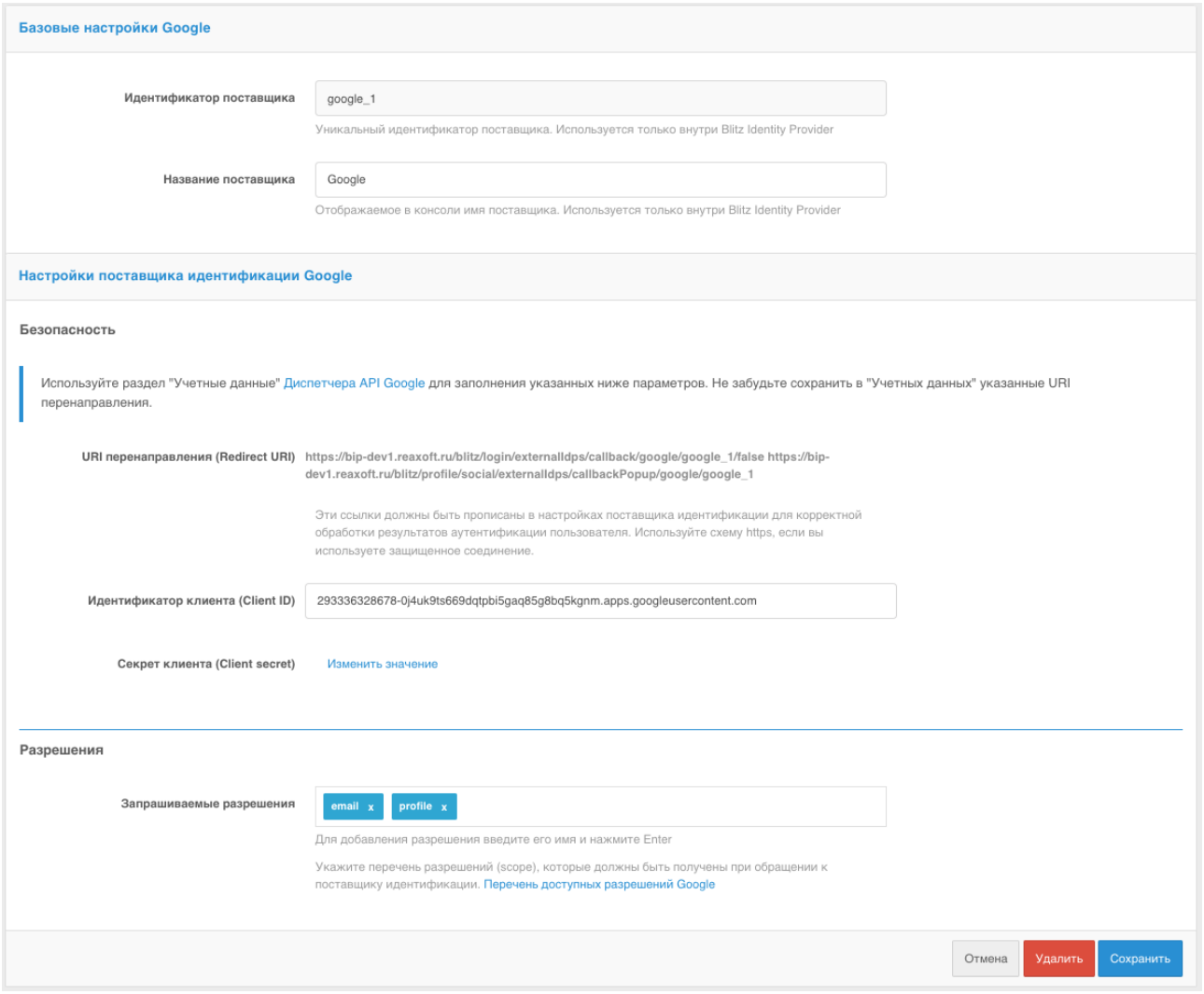

Рисунок 108 – Настройки поставщика идентификации Google

8.3. Вход через Яндекс

Для конфигурирования входа через учетную запись Яндекс необходимо выполнить следующие настройки:

1. Перейти в приложение «Яндекс.OAuth»[40] (см. Рисунок 109), в котором выполнить следующие операции:

[39] См.: https://developers.google.com/+/web/api/rest/oauth#authorization-scopes
[40] См.: https://oauth.yandex.ru/

- 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 (Рисунок 110) 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.

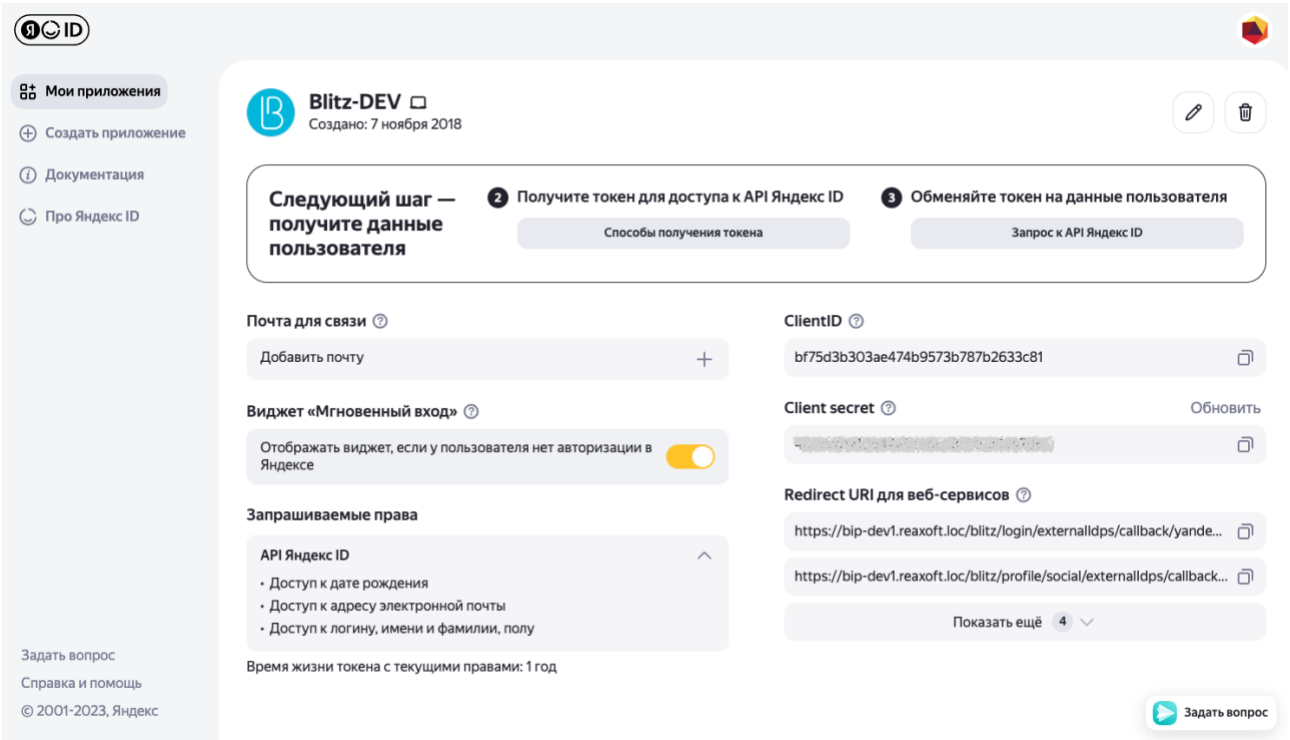

Рисунок 109 – Настройки в Яндекс.OAuth

2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип Яндекс.
3. Заполнить настройки поставщика идентификации (см. Рисунок 110):
   - Идентификатор поставщика;
   - Название поставщика;
   - Идентификатор клиента (ClientID), полученный в приложении Яндекс.OAuth;
   - Пароль приложения (Client secret), полученный в приложении Яндекс.OAuth;
   - Запрашиваемые разрешения (scope), предусмотренные в Яндекс.OAuth – для указанных ранее доступов следует указать login:email, login:info и login:birthday;

4. Настроить правила связывания (см. п. 8.18).
5. В разделе «Аутентификация» консоли управления включить использование метода аутентификации с использованием поставщика идентификации Яндекс (см. п. 4.6).

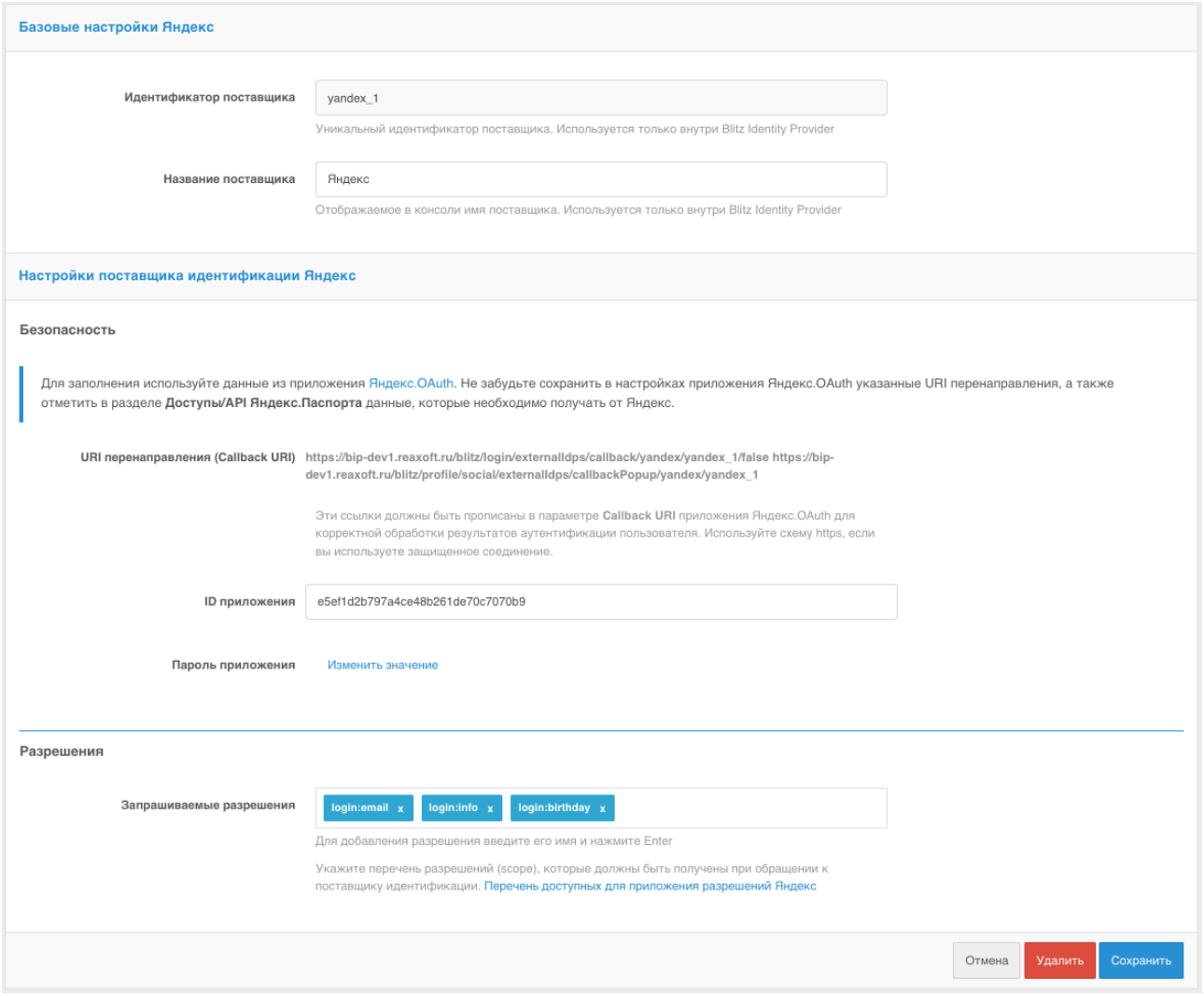

Рисунок 110 – Настройки поставщика идентификации Яндекс

8.4. Вход через Facebook

Для конфигурирования входа через учетную запись Facebook необходимо выполнить следующие настройки:

1. Перейти в панель «Facebook для разработчиков» (см. Рисунок 111)[41], в которой выполнить следующие операции:
   - добавить новое приложение, указав его название, адрес электронной почты для связи и категорию приложения;
   - создать идентификатор приложения;
   - перейти в настройки приложения, раздел «Основное». В этом разделе указать

[41] См.: https://developers.facebook.com/apps/

параметр «Домены приложения» (параметр должен соответствовать домену, на котором установлен Blitz Identity Provider) и добавить сайт с аналогичным URL.

- перейти в раздел «Проверка приложения» и активировать пункт «Сделать приложение «...» доступным для всех?»

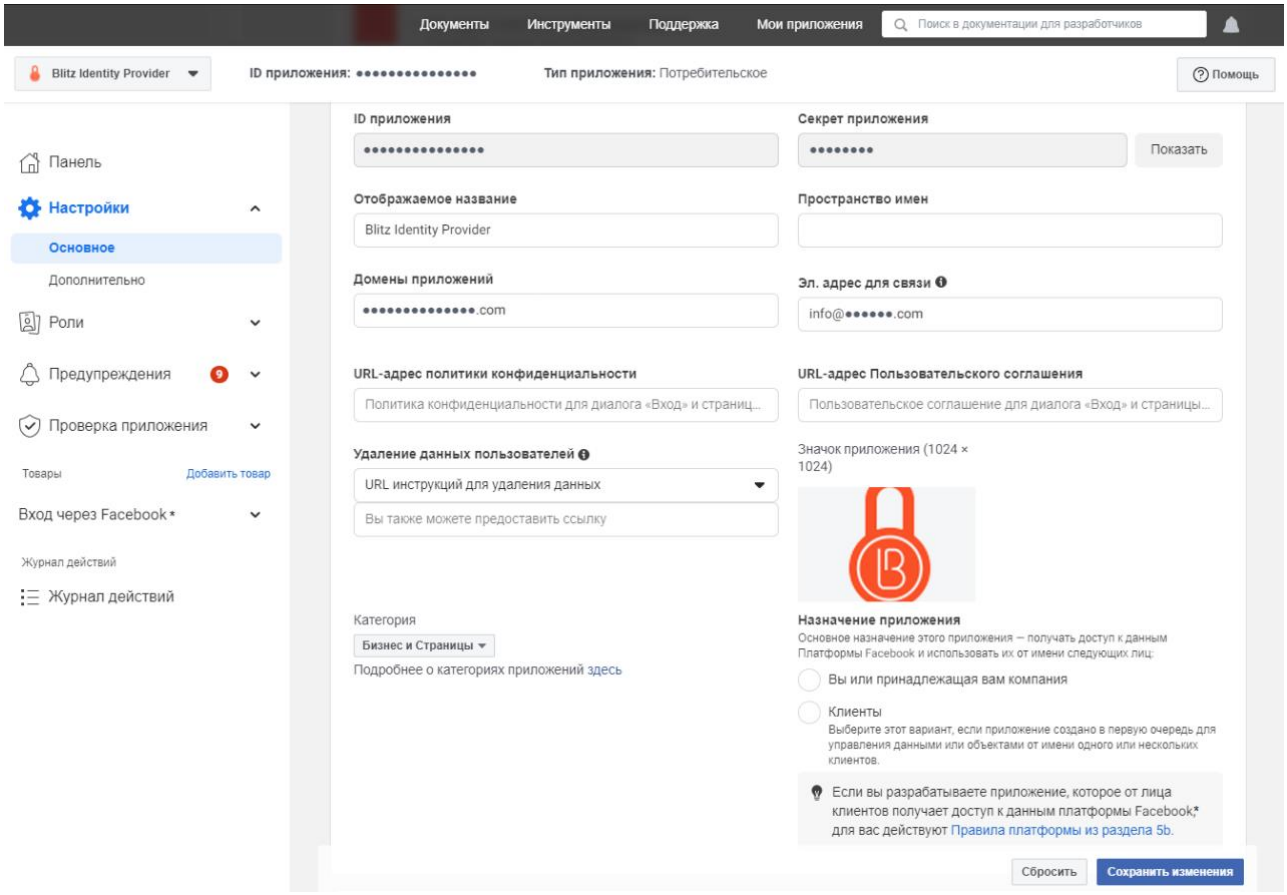

Рисунок 111 – Настройки в панели Facebook для разработчиков

2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип Facebook.
3. Заполнить настройки поставщика идентификации (см. Рисунок 112):
   - Идентификатор поставщика;
   - Название поставщика;
   - идентификатор приложения (App ID), полученный в панели Facebook для разработчиков;
   - секрет приложения (App Secret), полученный в панели Facebook для разработчиков;
   - запрашиваемые разрешения (scope), предусмотренные в Facebook[42];
   - запрашиваемые атрибуты, предусмотренные в Facebook; допустимо указывать только те атрибуты, которые предусмотрены выбранными разрешениями.
4. Настроить правила связывания (см. п. 8.18).

[42] См.: https://developers.facebook.com/docs/facebook-login/permissions/

5. В разделе «Аутентификация» консоли управления включить использование метода аутентификации с использованием поставщика идентификации Facebook (см. п. 4.6).

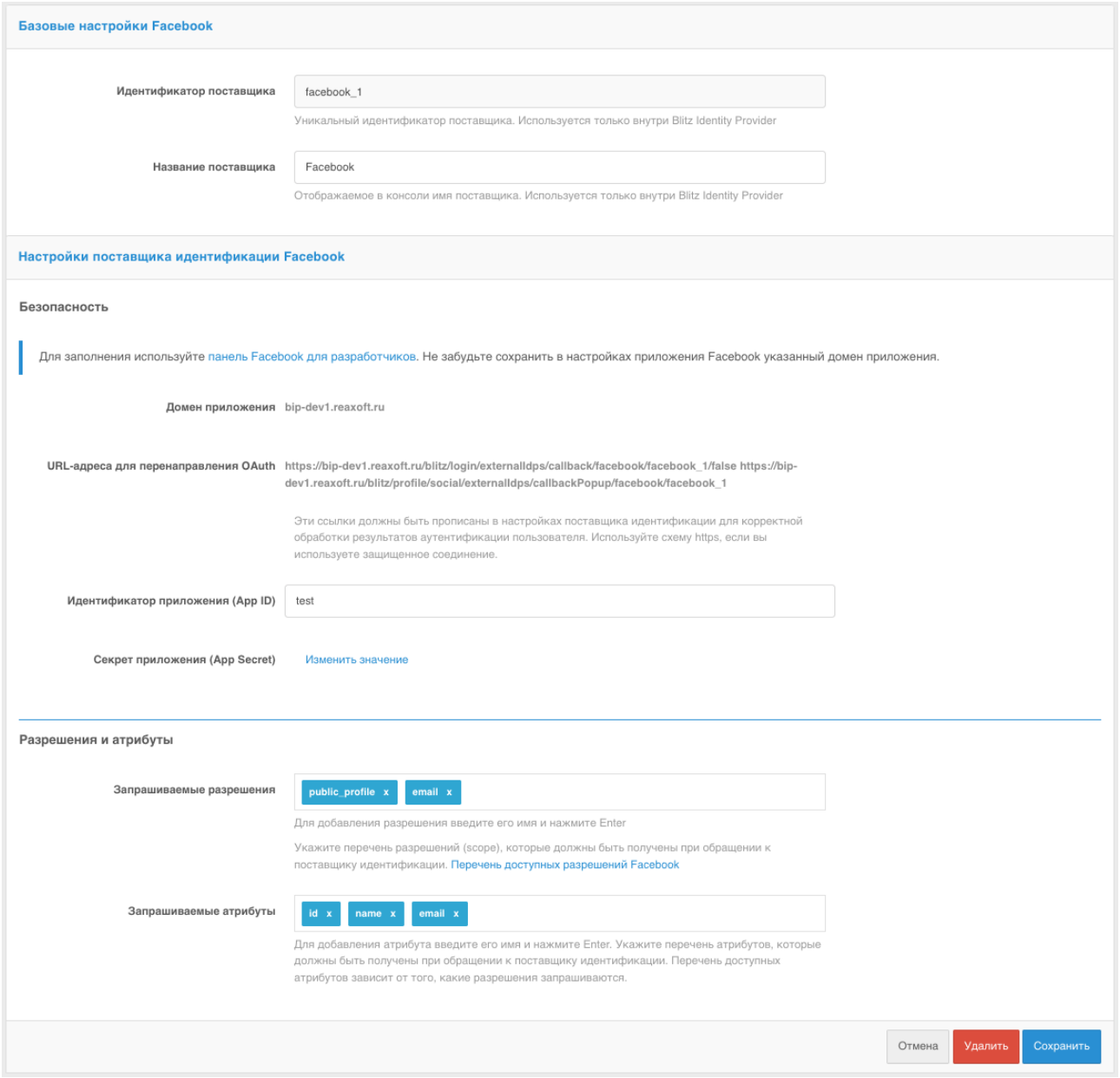

Рисунок 112 – Настройки поставщика идентификации Facebook

8.5. Вход через ВКонтакте

Для конфигурирования входа через учетную запись ВКонтакте необходимо выполнить следующие настройки:

1. Перейти в «Панель VK для разработчиков» (см. Рисунок 113)[43], в которой выполнить следующие операции:
   - перейти в раздел «Мои приложения»;
   - выбрать пункт «Создать приложение»;

[43] См.: https://new.vk.com/dev

- 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).

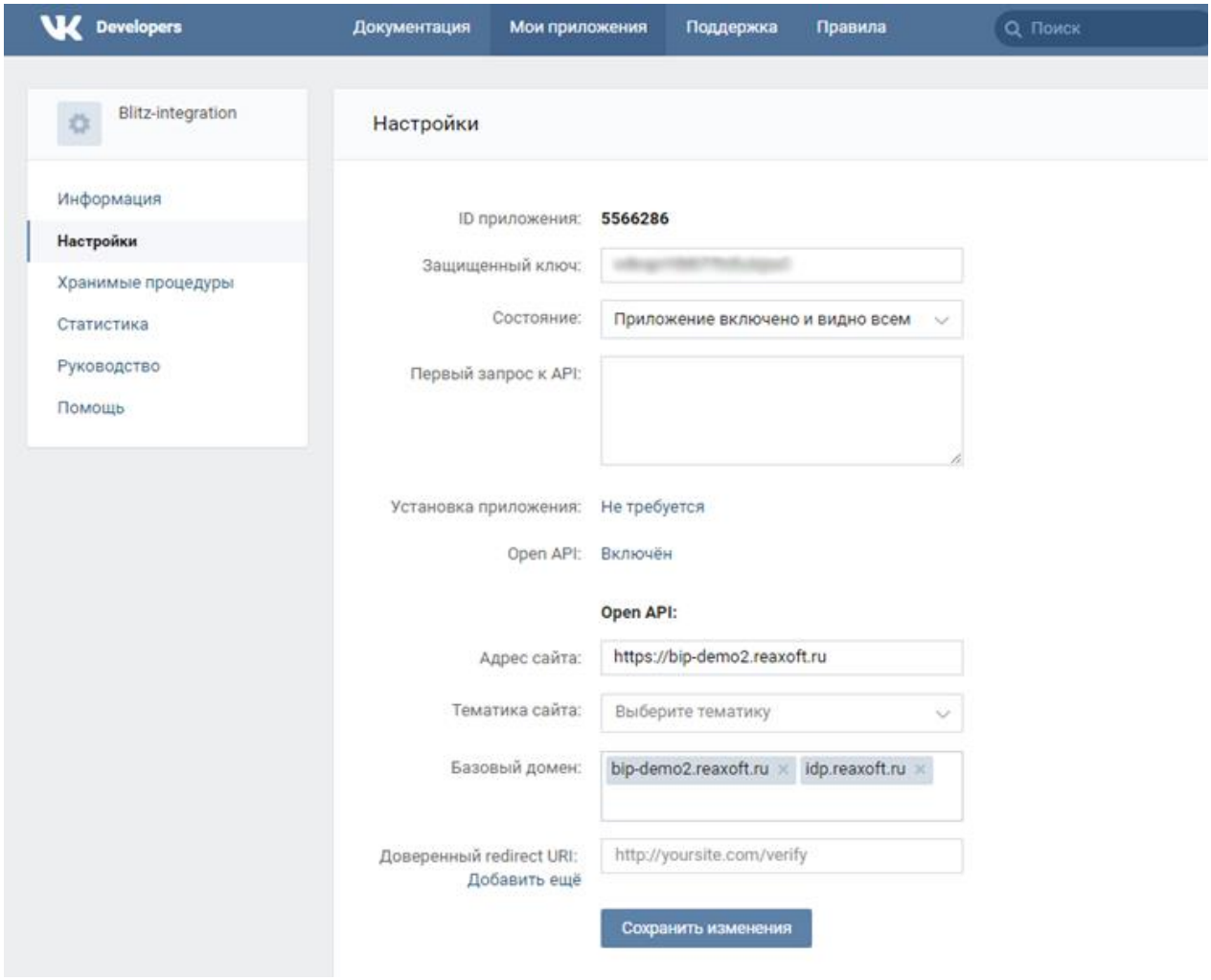

Рисунок 113 – Настройки в панели VK для разработчиков

2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип ВКонтакте.
3. Заполнить настройки поставщика идентификации (см. Рисунок 114):
   - 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;
   - Запрашиваемые разрешения, предусмотренные в ВКонтакте[44];

[44] См.: https://new.vk.com/dev/permissions

4. Настроить правила связывания (см. п. 8.18).
5. В разделе «Аутентификация» консоли управления включить использование метода аутентификации с использованием поставщика идентификации ВКонтакте (см. п 4.6).

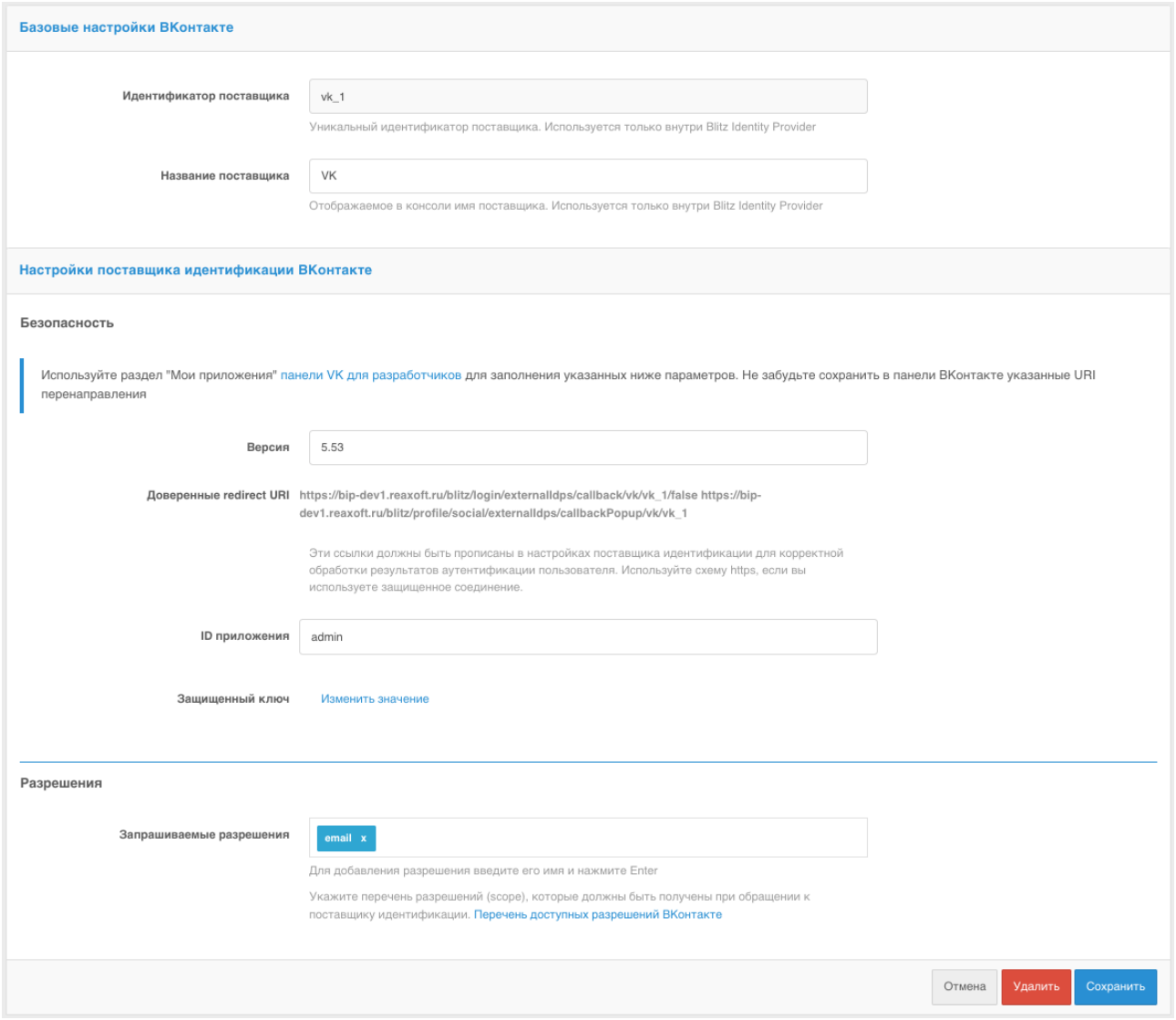

Рисунок 114 – Настройки поставщика идентификации ВКонтакте

8.6. Вход через Одноклассники

Для конфигурирования входа через учетную запись сети «Одноклассники» необходимо выполнить следующие настройки:

1. Перейти на страницу «OAuth авторизация»[45], 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;

[45] См.: https://apiok.ru/ext/oauth/

- 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 (см. Рисунок 115), например:

```
https://login.company.com/blitz/login/externalIdps/callback/ok/ok_1/false
https://login.company.com/blitz/profile/social/externalIdps/callbackPopup/ok/ok_1
```

2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип Одноклассники.
3. Заполнить настройки поставщика идентификации (см. Рисунок 115):
   - Идентификатор поставщика;
   - Название поставщика;
   - Название приложения (Application ID);
   - Секретный ключ приложения;
   - Публичный ключ приложения;
   - Запрашиваемые разрешения.

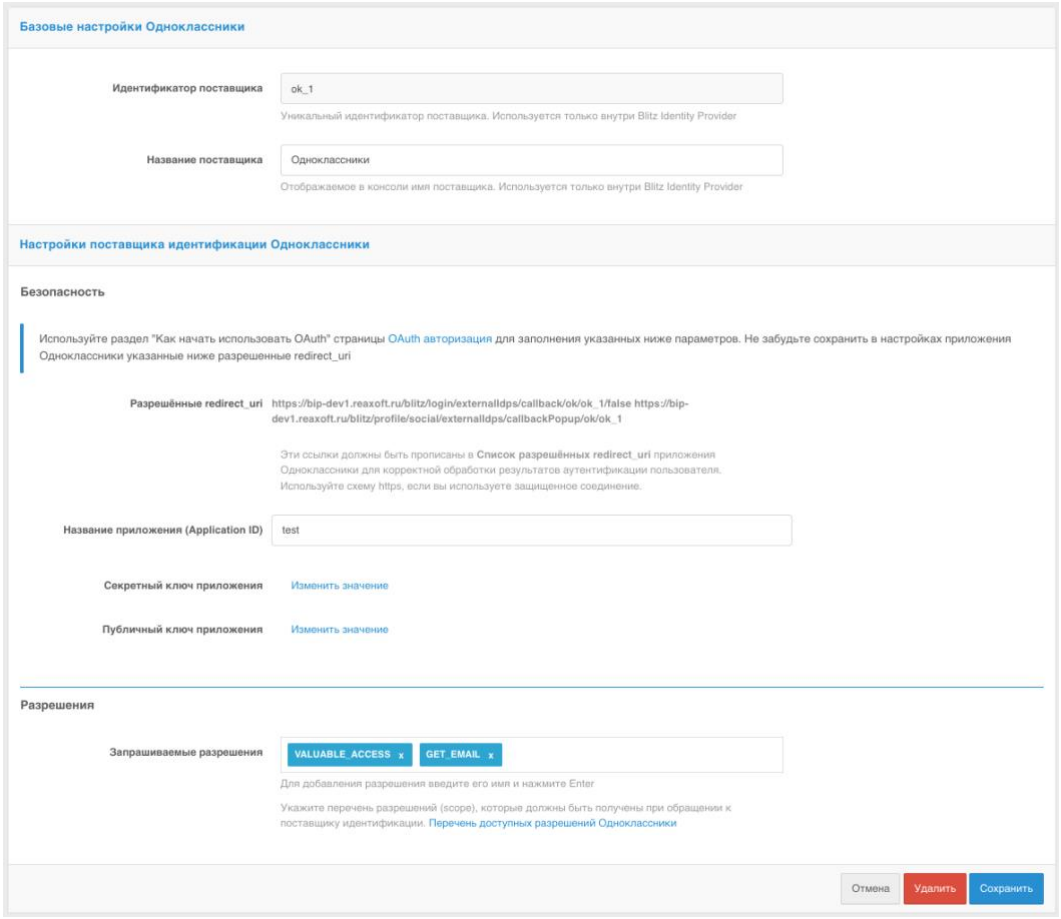

Рисунок 115 – Настройки поставщика идентификации сети Одноклассники

4. Настроить правила связывания (см. п. 8.18).
5. В разделе «Аутентификация» консоли управления включить использование метода аутентификации с использованием поставщика идентификации Одноклассники (см. п. 4.6).

8.7. Вход через Mail ID

Для конфигурирования входа через учетную запись Mail ID необходимо выполнить следующие настройки:

1. Перейти на страницу «Создание приложения» в OAuth@Mail.ru[46] (Рисунок 116), где выполнить следующие операции:
   - нажать на кнопку «Создать приложение»;
   - аутентифицировать под учетной записью Mail.ru;
   - ввести данные приложения, в том числе название приложения;
   - в поле «Все redirect_uri» указать перечень URI перенаправления, образцы которых Blitz Identity Provider показывает в настройках подключения Mail ID (Рисунок 117), 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 ID Приложения и его секрет, они потребуются для последующего ввода в Blitz Identity Provider.
2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип Mail ID.
3. Заполнить настройки поставщика идентификации (см. Рисунок 117):
   - Идентификатор поставщика;
   - Название поставщика;
   - Идентификатор клиента (ID приложения), полученный ранее;
   - Секрет приложения, полученный ранее;
   - Запрашиваемые разрешения (scope), например, userinfo;
4. Настроить правила связывания (см. п. 8.18).
5. В разделе «Аутентификация» консоли управления включить использование метода аутентификации с использованием поставщика идентификации Mail ID (см. п. 4.6).

[46] См.: https://help.mail.ru/developers/oauth/app

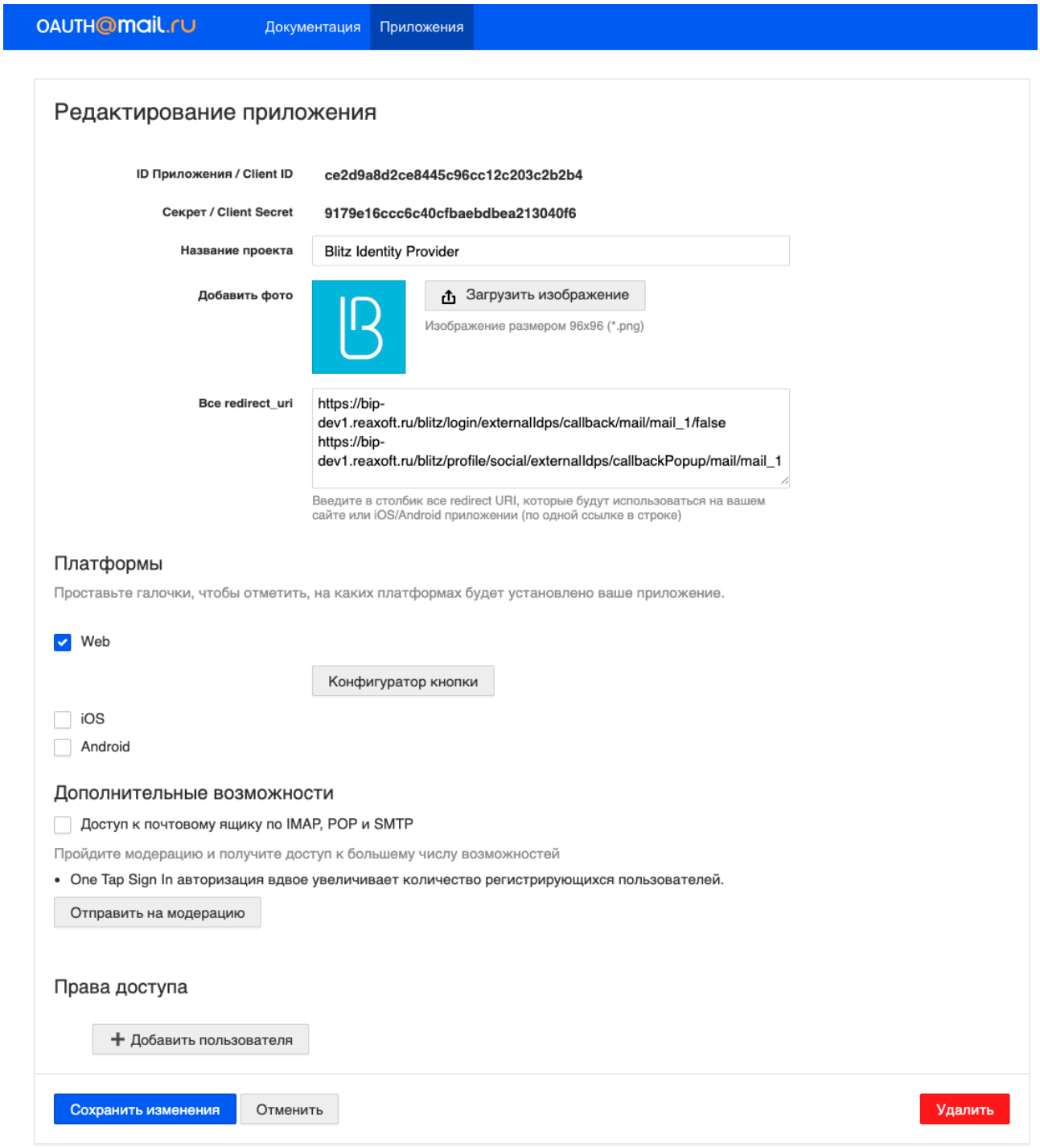

Рисунок 116 – Настройки в OAuth@Mail.ru

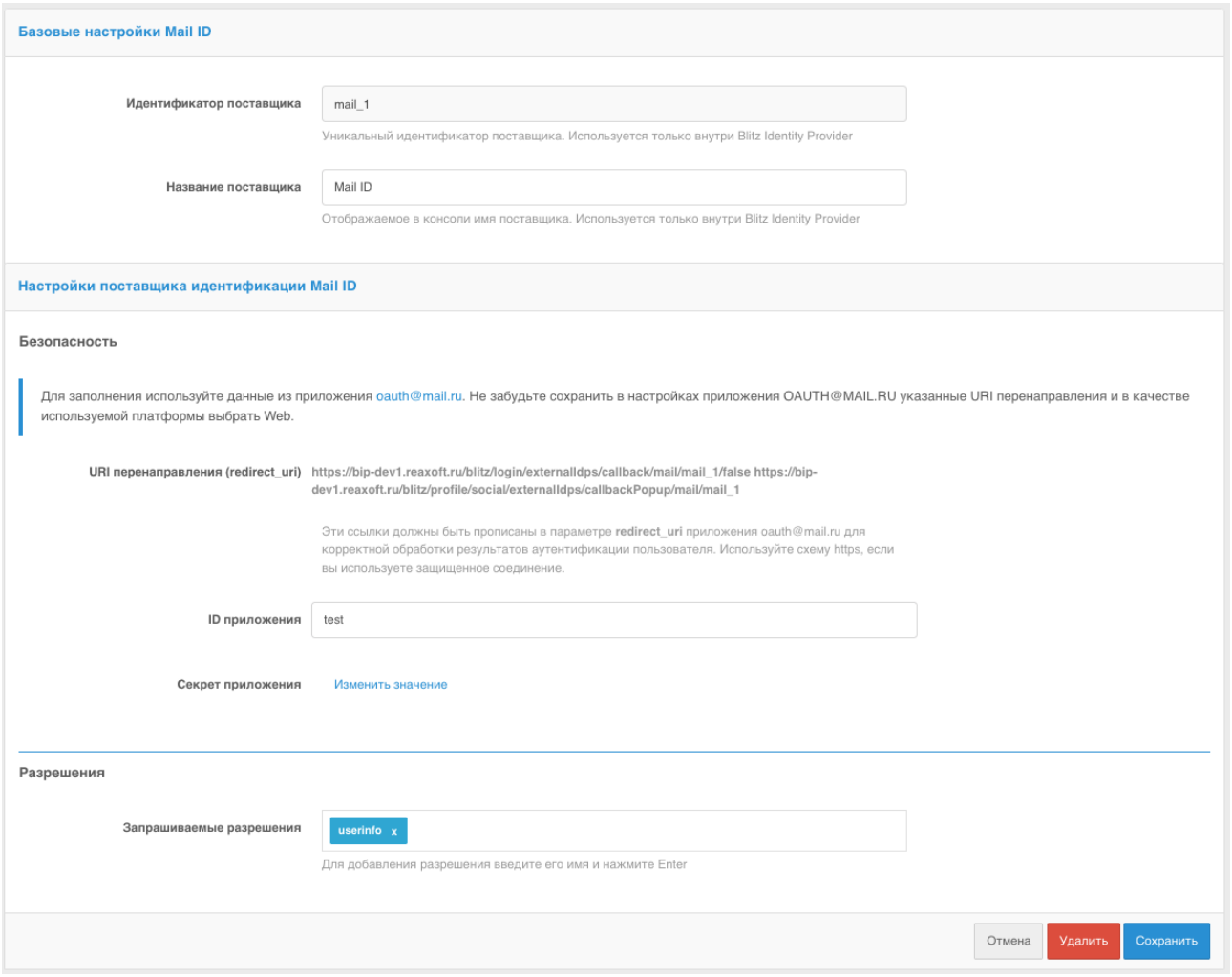

Рисунок 117 – Настройки поставщика идентификации Mail ID

8.8. Вход через 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. Задайте базовые настройки VK ID: 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.
2. Перейдите в онлайн-сервис VK ID [47] для разработчиков. Если у вас нет аккаунта, создайте его. В разделе **Приложения** аккаунта зарегистрируйте приложение Blitz Identity Provider. Для этого выполните следующие действия:

- Нажмите **Добавить приложение**.

[47] См. https://id.vk.com/about/business/go

- Шаг 1: введите название приложения, выберите платформу **Web**, задайте логотип.
- 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
```

- Нажмите **Готово**. В результате регистрации будут сгенерированы значения параметров **ID приложения** и **Сервисный ключ доступа**.

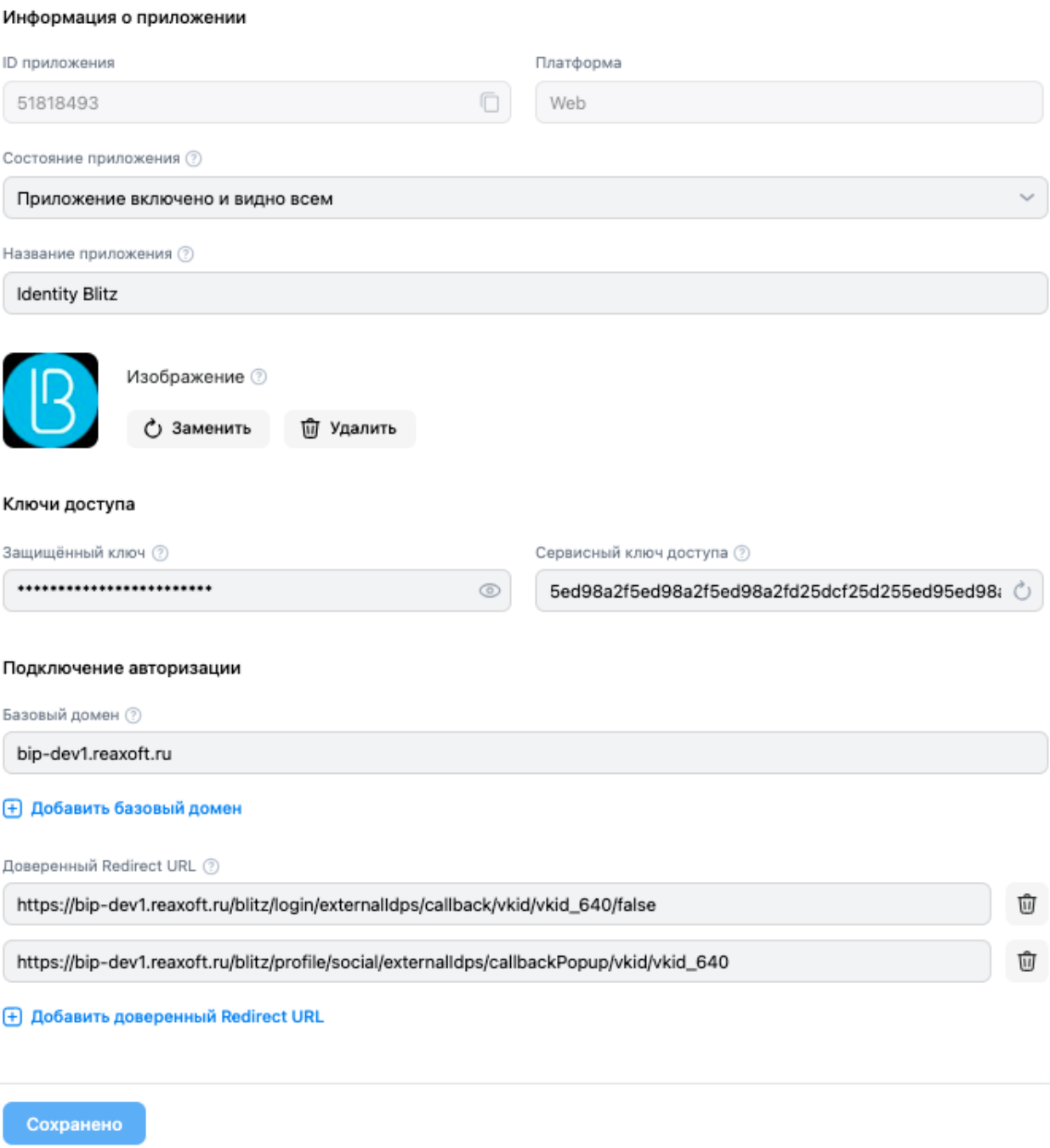

Рисунок 118 – Регистрация приложения в VK ID

3. 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**.

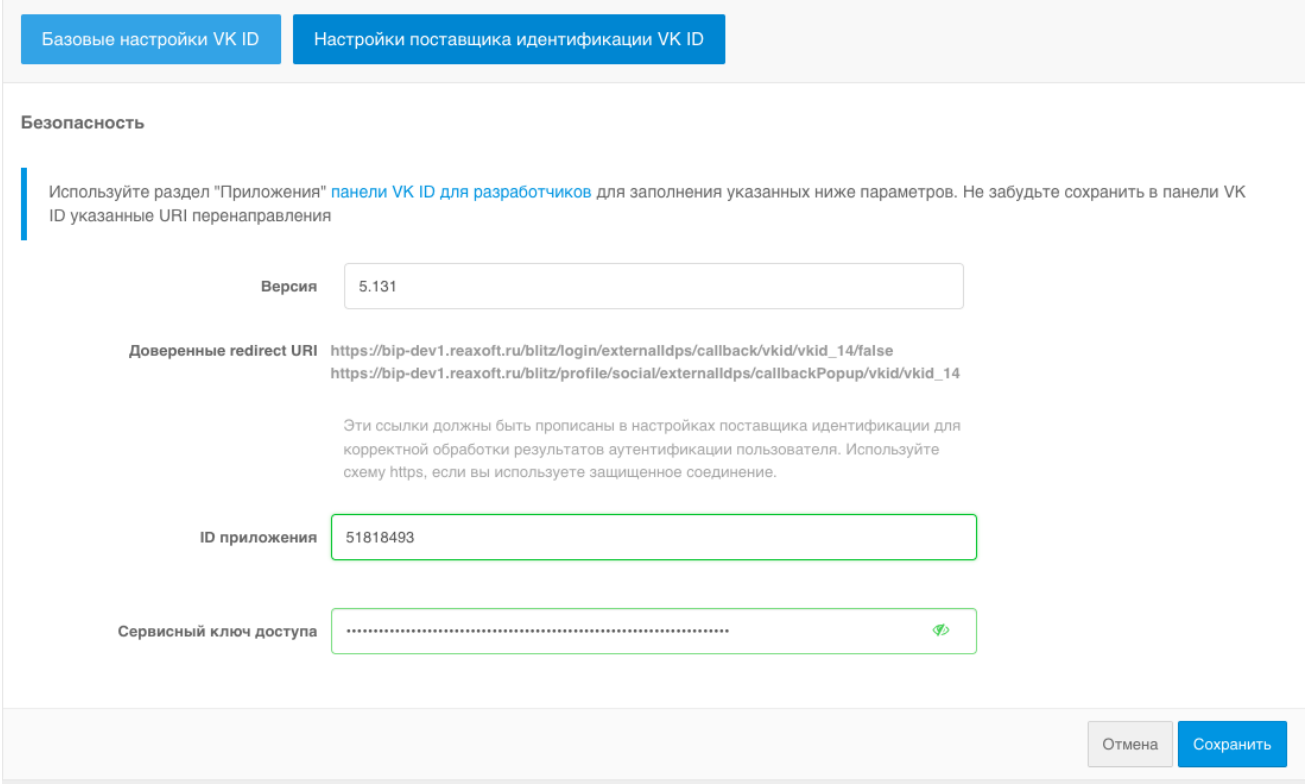

Рисунок 119 – Конфигурирование поставщика VK ID в Blitz Identity Provider

4. Настройте правила связывания (см. п. 8.18).
5. В разделе **Аутентификация** консоли управления включите использование метода аутентификации с использованием поставщика идентификации VK ID (см. п. 4.6).

8.9. Вход через Единую систему идентификации и аутентификации (ЕСИА)

Для конфигурирования входа через учетную запись ЕСИА необходимо выполнить следующие настройки:

1. Получить в удостоверяющем центре ключ электронной подписи для взаимодействия с ЕСИА и выгрузить сертификат открытого ключа. Произвести конвертацию ключа в формат, совместимый с Blitz Identity Provider (см. п. 16.1.17). Сертификат ключа необходимо будет зарегистрировать на Технологическом портале ЕСИА (см. следующий пункт).
2. Зарегистрировать систему организации на Технологическом портале ЕСИА[48]:
   - нажать на кнопку «Добавить систему»;

[48] См.: https://esia.gosuslugi.ru/console/tech/. До регистрации ИС в ЕСИА необходимо зарегистрировать учетную запись организации в ЕСИА и дать одному из сотрудников доступ к Технологическому порталу ЕСИА.

- указать название системы, отображаемое название, мнемонику системы, список URL системы (задать домен развернутой системы Blitz Identity Provider, с указанием протокола https), алгоритм формирования электронной подписи и выбрать ответственного сотрудника (см. Рисунок 120);
- сохранить данные и перейти к настройке сертификатов информационной системы;
- загрузить сертификат системы на Технологическом портале (см. Рисунок 121);

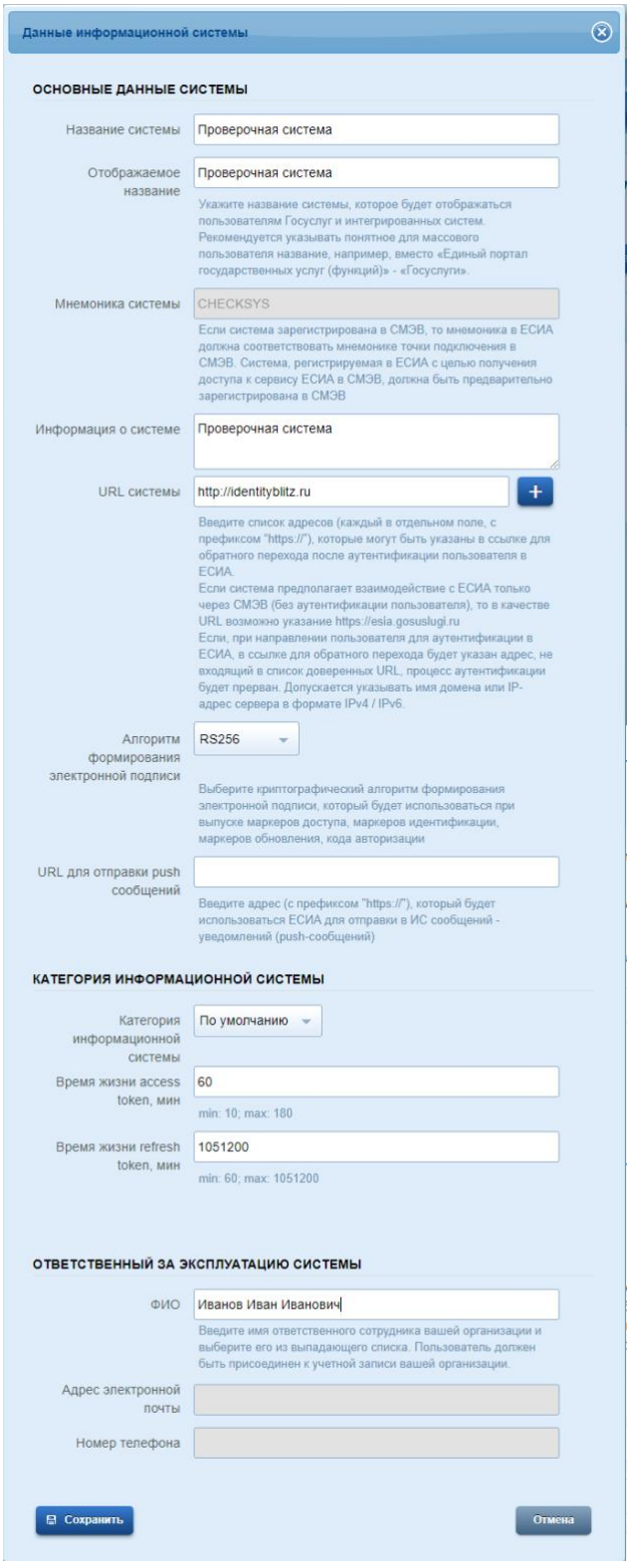

Рисунок 120 – Регистрация системы на Технологическом портале ЕСИА

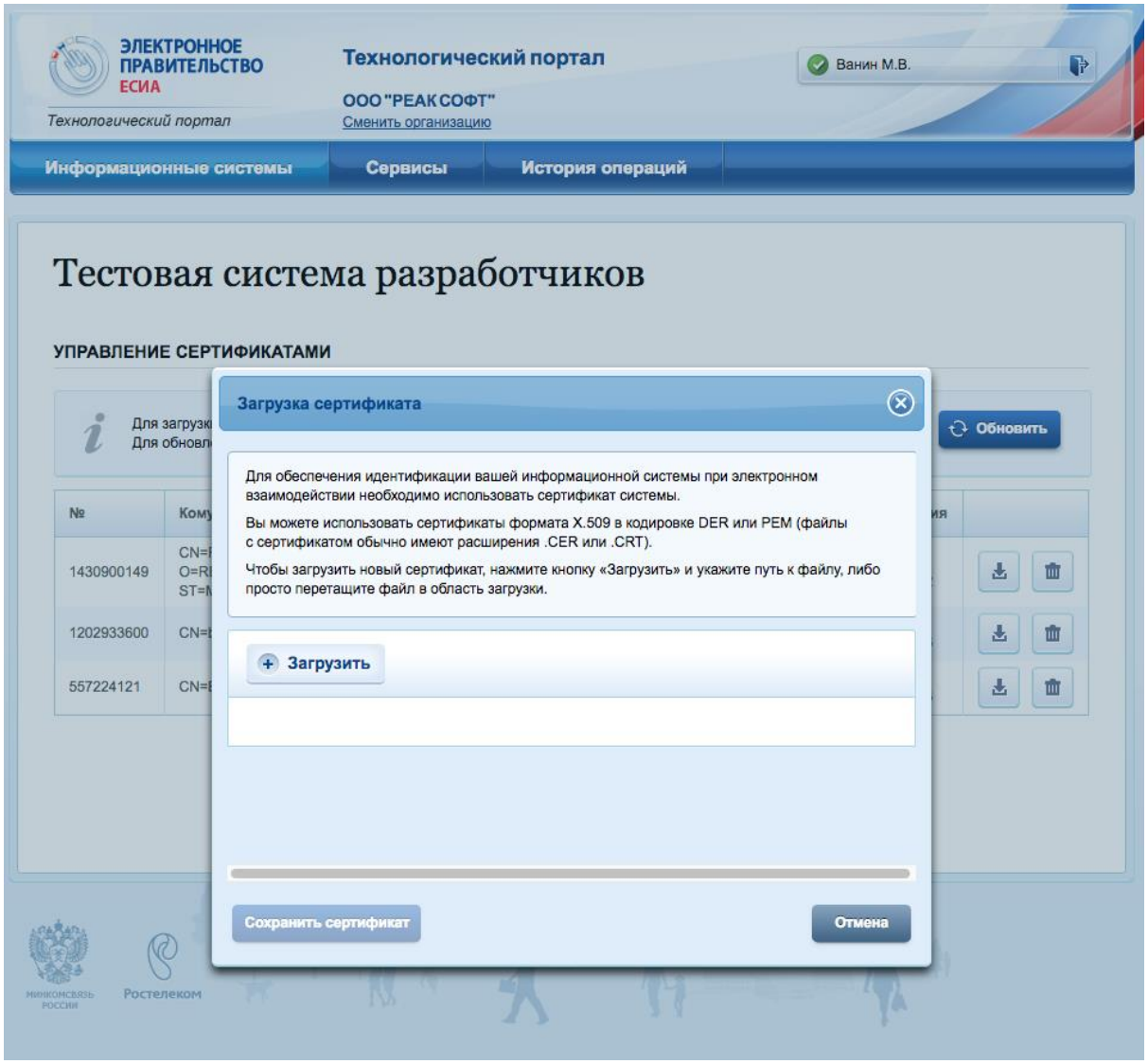

Рисунок 121 – Добавление сертификата системы на Технологическом портале ЕСИА

3. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип ЕСИА.
4. Заполнить настройки поставщика идентификации (см. Рисунок 122):
   - Идентификатор поставщика;
   - Название поставщика;
   - URL для авторизации – адрес обработчика ЕСИА, вызываемого из браузера, например, https://esia-portal1.test.gosuslugi.ru/aas/oauth2/ac (ТЕСТ ЕСИА) или https://esia.gosuslugi.ru/aas/oauth2/ac (ПРОД ЕСИА);
   - URL для получения и обновления маркера – адрес обработчика ЕСИА, вызываемого с сервера Blitz Identity Provider для получения маркера доступа, например, https://esia-portal1.test.gosuslugi.ru/aas/oauth2/te (ТЕСТ ЕСИА) или https://esia.gosuslugi.ru/aas/oauth2/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);
- Мнемоника системы (client_id), указанная ранее на Технологическом портале ЕСИА;
- Идентификатор ключа электронной подписи (alias) – идентификатор ключа электронной подписи, загруженный в хранилище Blitz Identity Provider[49]. Сертификат соответствующего ключа электронной подписи должен быть загружен на Технологическом портале ЕСИА;
- Запрашиваемые разрешения – перечень запрашиваемых разрешений из ЕСИА;
- Запрашиваемые данные пользователя – необходимо отметить те данные, которые следует получать из ЕСИА; эти данные должны быть доступны по запрашиваемым разрешениям.

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[50].

5. При необходимости настроить вход через ЕСИА в режиме выбора сотрудника организации (см. п. 16.1.18).
6. Настроить правила связывания (см. п. 8.18).
7. В разделе «Аутентификация» консоли управления включить использование метода аутентификации с использованием поставщика идентификации ЕСИА (см. п. 4.6).

[49] Хранилище, указанное в разделе keystore конфигурационного файла Blitz Identity Provider.
[50] См.: https://identityblitz.ru/services/esia-integration

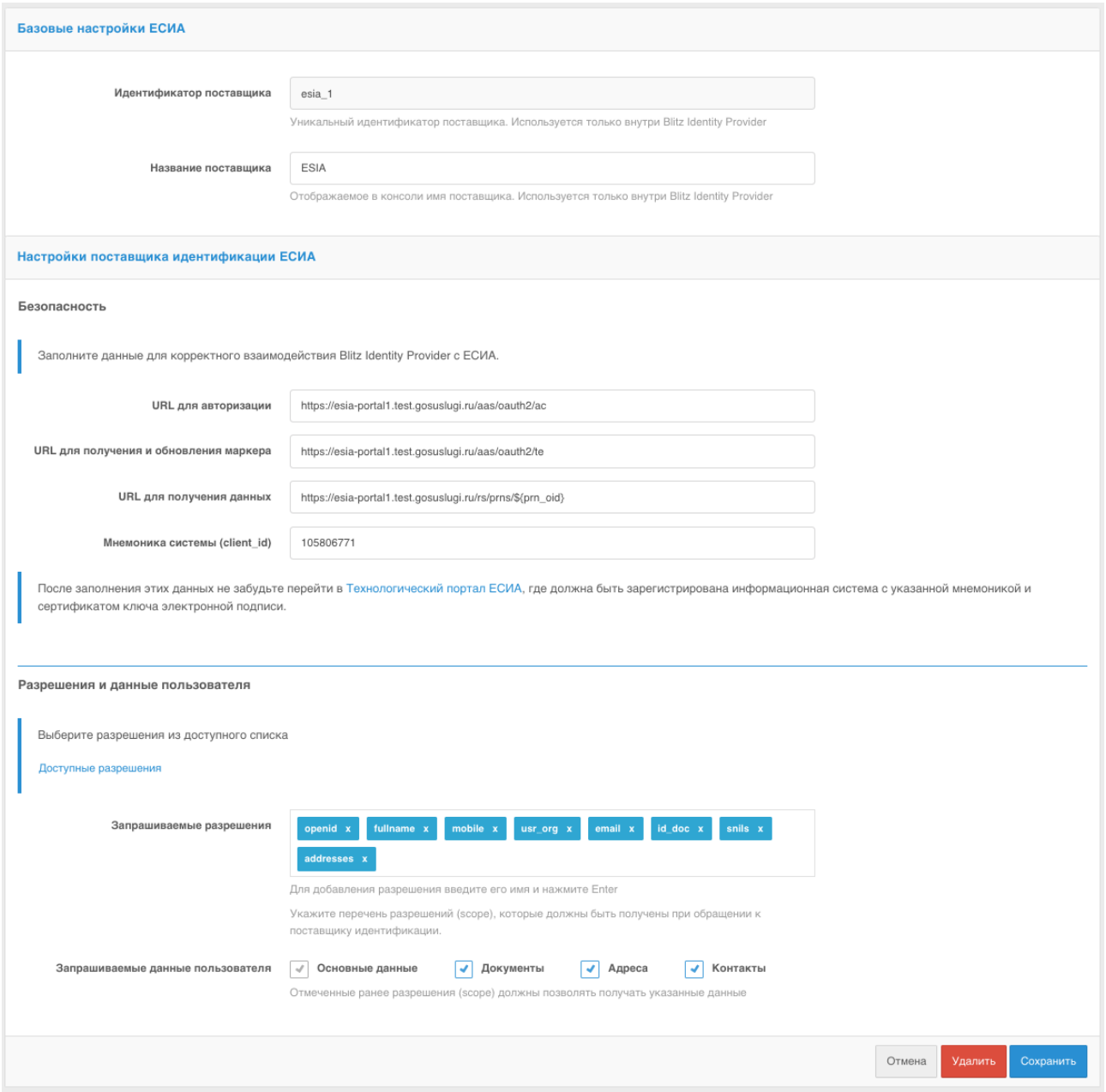

Рисунок 122 – Настройки поставщика идентификации ЕСИА

8.10. Вход через Цифровой профиль ЕСИА

Для конфигурирования входа через Цифровой профиль ЕСИА необходимо выполнить следующие настройки:

1. Получить в удостоверяющем центре ключ электронной подписи для взаимодействия с ЕСИА и выгрузить сертификат открытого ключа. Произвести конвертацию ключа в формат, совместимый с Blitz Identity Provider (см. п. 16.1.17). Сертификат ключа необходимо зарегистрировать на Технологическом портале ЕСИА (см. следующий пункт).

2. Зарегистрировать систему организации на Технологическом портале ЕСИА[51]:
   - нажать на кнопку «Добавить систему»;
   - указать название системы, отображаемое название, мнемонику системы, список URL системы (задать домен развернутой системы Blitz Identity Provider, с указанием протокола https), алгоритм формирования электронной подписи и выбрать ответственного сотрудника (см. Рисунок 123);
   - сохранить данные и перейти к настройке сертификатов информационной системы;
   - загрузить сертификат системы на Технологическом портале (см. Рисунок 124);

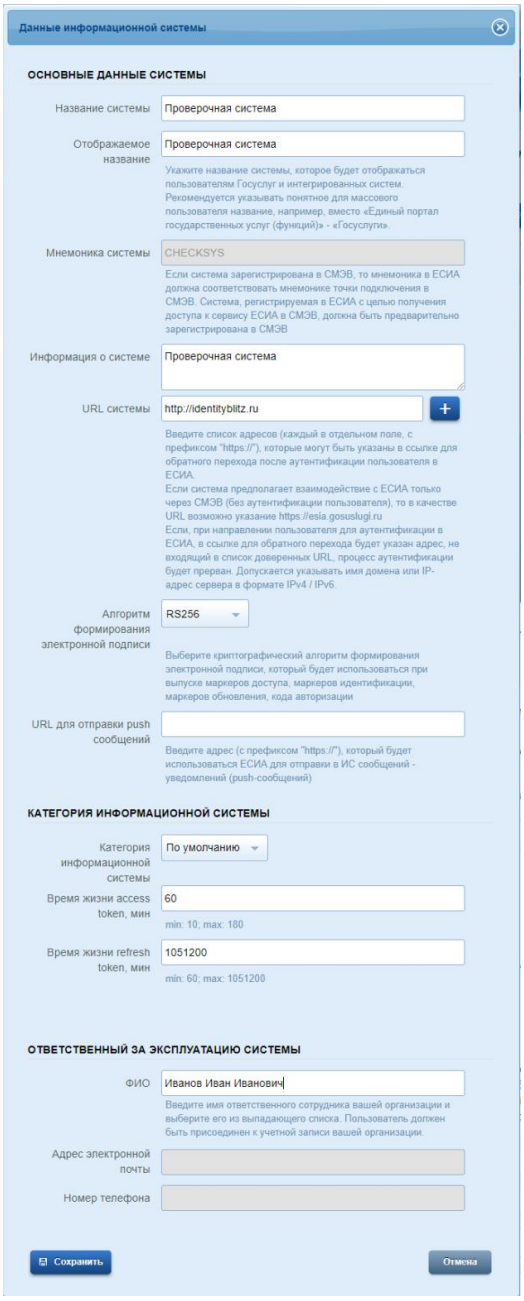

Рисунок 123 – Добавление системы на Технологическом портале ЕСИА

[51] См.: https://esia.gosuslugi.ru/console/tech/. До регистрации ИС в ЕСИА необходимо зарегистрировать учетную запись организации в ЕСИА и дать одному из сотрудников доступ к Технологическому порталу ЕСИА.

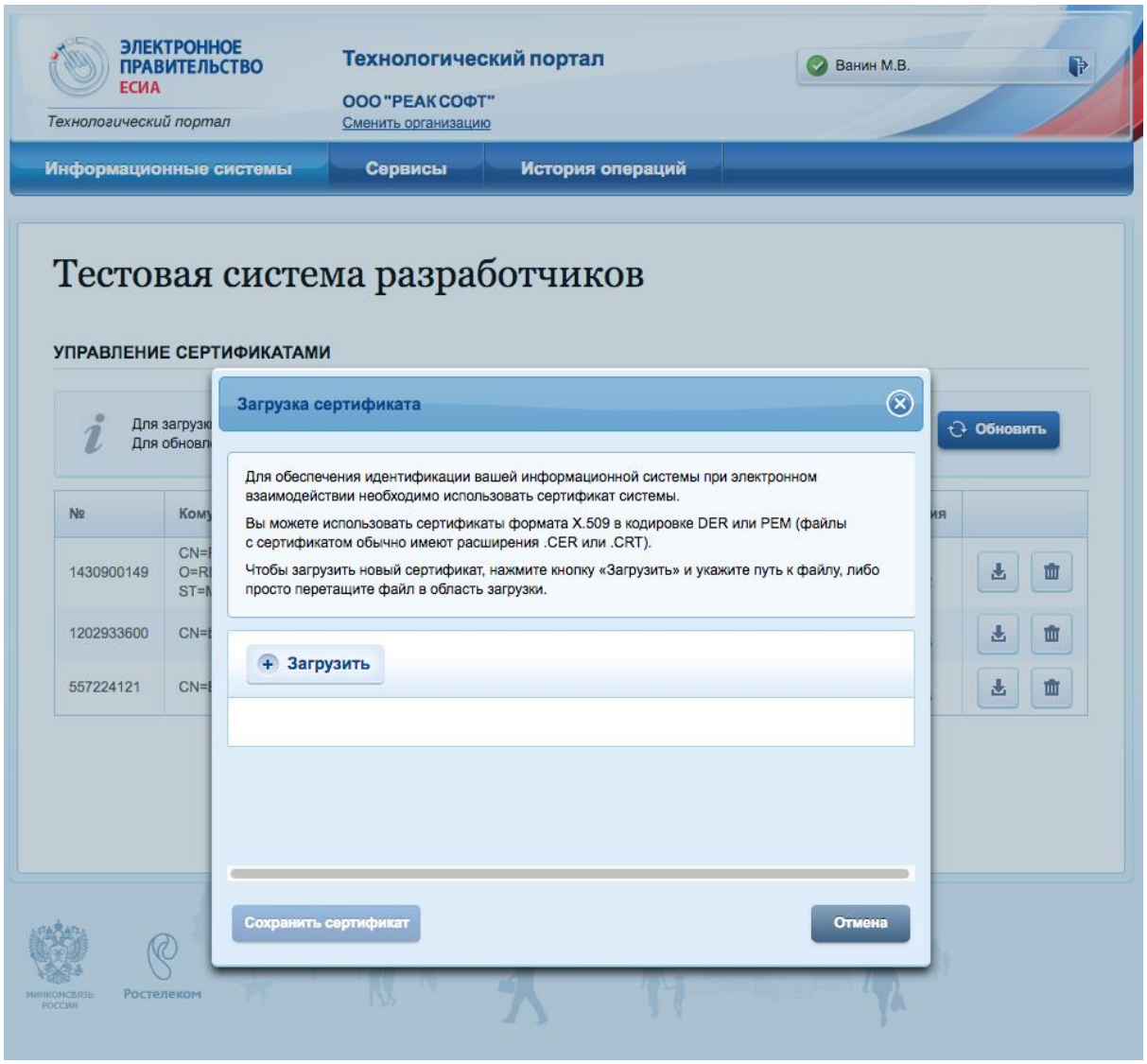

Рисунок 124 – Добавление сертификата системы на Технологическом портале ЕСИА

3. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип ЦП ЕСИА.
4. Заполнить настройки поставщика идентификации (см. Рисунок 125):
   - Идентификатор поставщика;
   - Название поставщика;
   - URL для авторизации – адрес обработчика ЕСИА, вызываемого из браузера, например, https://esia-portal1.test.gosuslugi.ru/aas/oauth2/ac (ТЕСТ ЕСИА) или https://esia.gosuslugi.ru/aas/oauth2/ac (ПРОД ЕСИА);
   - URL для получения и обновления маркера – адрес обработчика ЕСИА, вызываемого с сервера Blitz Identity Provider для получения маркера доступа, например, https://esia-portal1.test.gosuslugi.ru/aas/oauth2/te (ТЕСТ ЕСИА) или https://esia.gosuslugi.ru/aas/oauth2/te (ПРОД ЕСИА);
   - URL для получения данных – адрес обработчика ЕСИА, вызываемого с сервера

Blitz Identity Provider для получения данных цифрового профиля, например, https://esia-portal1.test.gosuslugi.ru/digital/api/public/v1 (ТЕСТ ЕСИА) или https://esia.gosuslugi.ru/digital/api/public/v1 (ПРОД ЕСИА);

- Мнемоника системы (client_id), указанная ранее на Технологическом портале ЕСИА;
- Идентификатор ключа электронной подписи (alias) – идентификатор ключа электронной подписи, загруженный в хранилище Blitz Identity Provider[52]. Именно сертификат ключа этой электронной подписи должен быть загружен в Технологический портал ЕСИА;
- Запрашиваемые разрешения – перечень запрашиваемых разрешений из ЕСИА;
- Тип согласия – запрашиваемый в цифровом профиле тип согласия;
- Срок действия согласия – количество минут, на которое запрашивается согласие;
- Ответственное лицо – сотрудник организации, ответственный за обработку данных, полученных из цифрового профиля;
- Запрашиваемые данные пользователя – необходимо отметить те данные, которые следует получать из цифрового профиля ЕСИА; эти данные должны быть доступны по запрашиваемым разрешениям.

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[53] с доступом к цифровому профилю.

5. Настроить правила связывания (см. п. 8.18).
6. В разделе «Аутентификация» консоли управления включить использование метода аутентификации с использованием поставщика идентификации ЦП ЕСИА (см. п. 4.6).

[52] Хранилище, указанное в разделе keystore конфигурационного файла Blitz Identity Provider.
[53] См.: https://identityblitz.ru/services/esia-integration

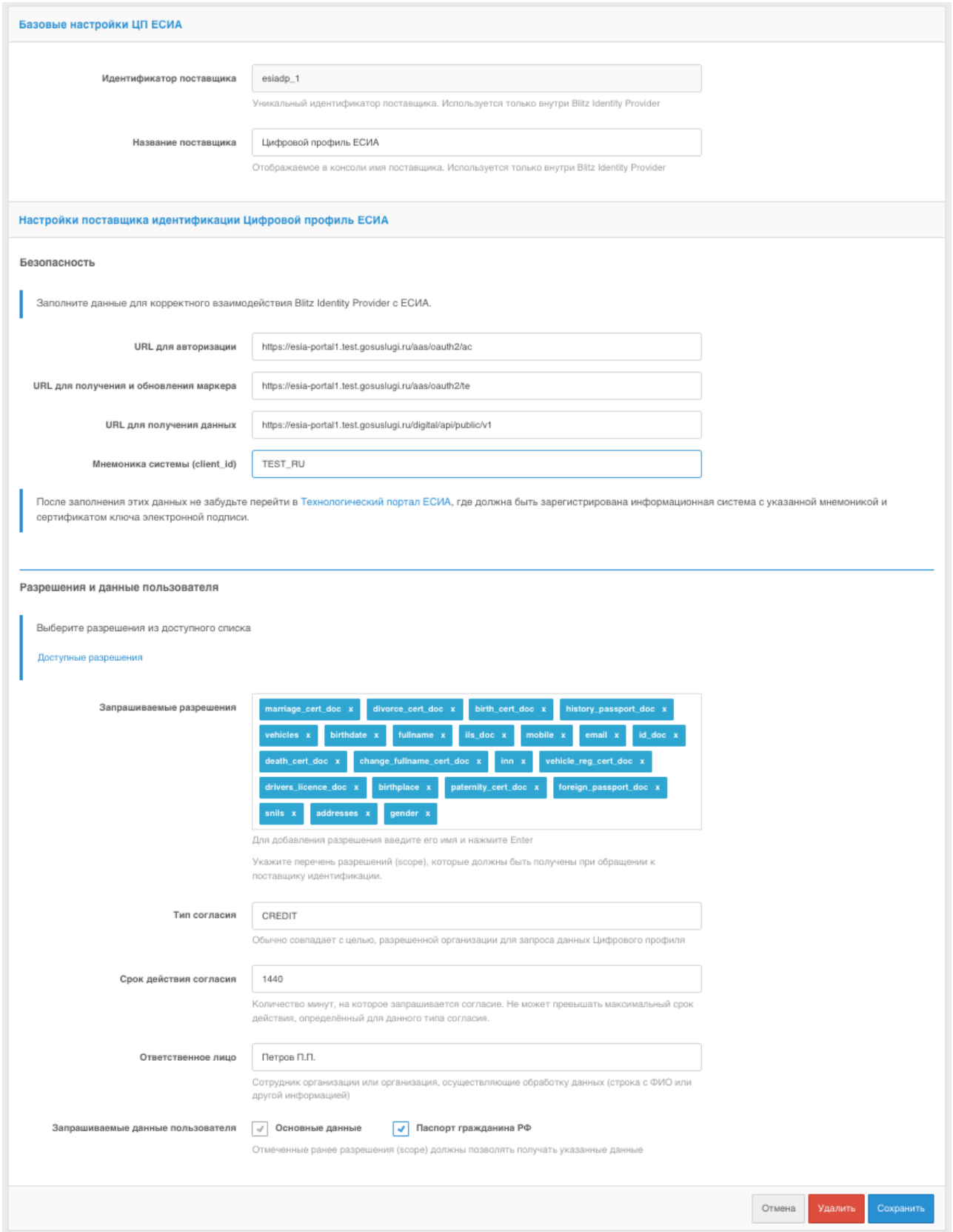

Рисунок 125 – Настройки поставщика идентификации ЦП ЕСИА

8.11. Вход через Сбер ID

Для конфигурирования входа через Сбер ID необходимо выполнить следующие настройки:

1. Зарегистрировать приложение в системе Сбер ID. Для этого воспользоваться инструкцией, размещенной на официальном сайте этого поставщика идентификации[54]. По результатам регистрации необходимо получить:
   - идентификатор клиента (Client ID);
   - секрет клиента (Client Secret);
   - сертификат системы, подключаемой 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 (см. Рисунок 126):
   - 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;
   - URL для получения данных – адрес, по которому происходит получение данных пользователя. Должен быть указан внутренний адрес сети, обращение через который обеспечит работу по защищенному каналу связи между организацией и банком;
   - Идентификатор клиента (Client ID);
   - Секрет клиента (Client Secret);
   - Запрашиваемые группы данных – какие группы данных запрашивать из Сбер ID.
5. Настроить правила связывания (см. п. 8.18).
6. В разделе «Аутентификация» консоли управления включить использование метода аутентификации с использованием поставщика идентификации Сбер ID (см. п. 4.6).

[54] См.: https://developer.sberbank.ru/doc/v1/sberbank-id/enrollsteps

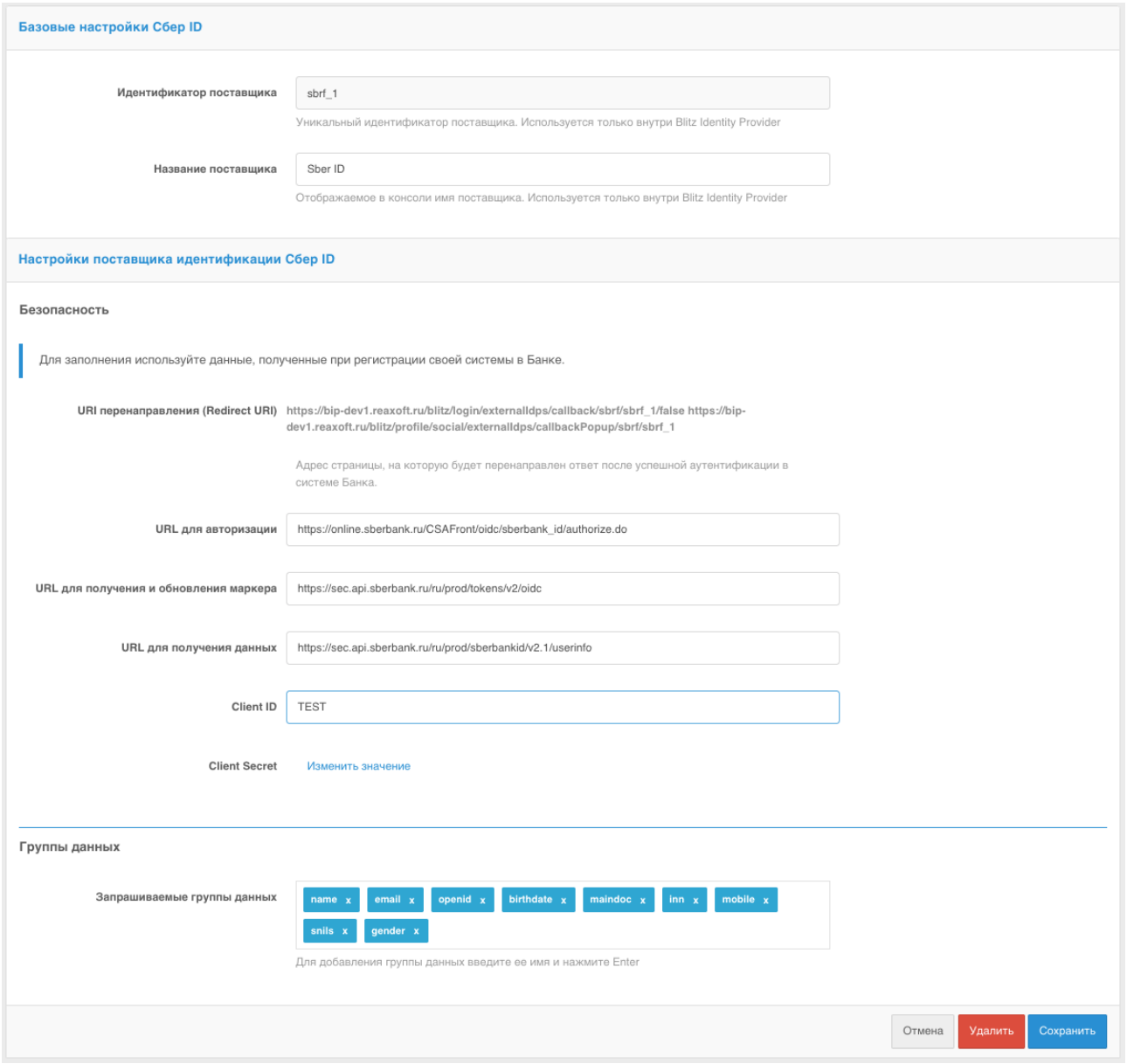

Рисунок 126 – Настройки поставщика идентификации Сбер ID

8.12. Вход через Tinkoff ID

Для конфигурирования входа через Тинькофф ID необходимо выполнить следующие настройки:

1. Зарегистрировать приложение в системе Tinkoff ID. Для этого подать заявку на официальном сайте этого поставщика идентификации[55]. По результатам регистрации необходимо получить:
   - идентификатор клиента (Client ID);
   - секрет клиента (Client Secret).

[55] См.: https://www.tinkoff.ru/corporate/business-solutions/open-api/tinkoff-id/integration/

2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип Tinkoff ID.
3. Заполнить настройки поставщика идентификации (см. Рисунок 127):
   - Идентификатор поставщика;
   - Название поставщика;
   - URL для авторизации – адрес, по которому должна инициироваться аутентификация, например: https://id.tinkoff.ru/auth/authorize;
   - URL для получения и обновления маркера – адрес, по которому происходит получение и обновление маркера доступа, например: https://id.tinkoff.ru/auth/token;
   - URL для получения данных – адрес, по которому происходит получение данных пользователя, например: https://id.tinkoff.ru/userinfo/userinfo;
   - URL для получения паспортных данных – адрес, по которому происходит получение паспортных данных пользователя, например: https://business.tinkoff.ru/openapi/api/v1/individual/documents/passport;
   - URL для получения данных о СНИЛС – адрес, по которому происходит получение СНИЛС пользователя, например: https://business.tinkoff.ru/openapi/api/v1/individual/documents/snils;
   - Идентификатор клиента (Client ID);
   - Секрет клиента (Client Secret);
   - Запрашиваемые группы данных – перечень запрашиваемых групп данных из Tinkoff ID.
4. Настроить правила связывания (см. п. 8.18).
5. В разделе «Аутентификация» консоли управления включить использование метода аутентификации с использованием поставщика идентификации Tinkoff ID (см. п. 4.6).

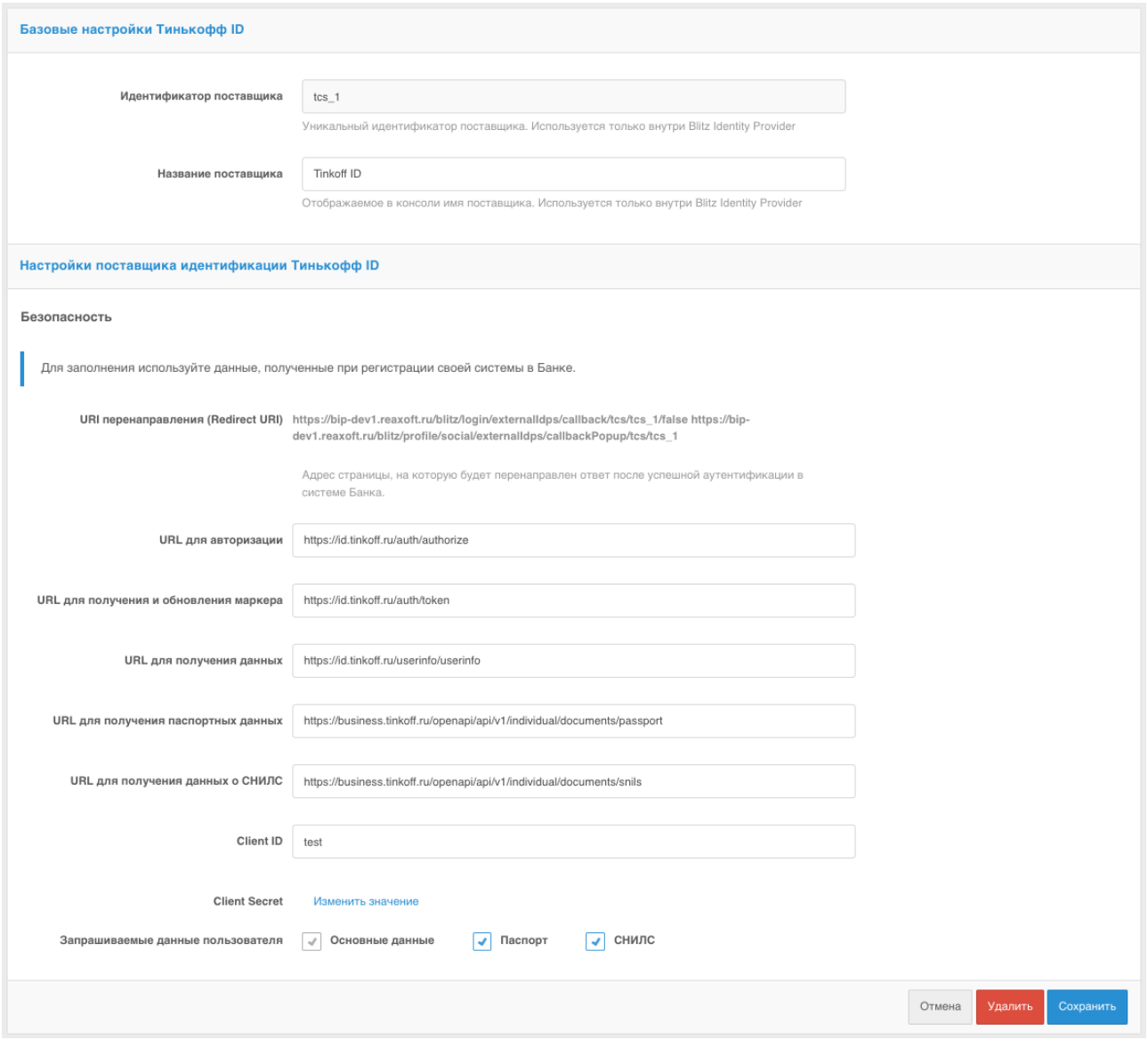

Рисунок 127 – Настройки поставщика идентификации Tinkoff ID

8.13. Вход через систему идентификации ВТБ ID

Для конфигурирования входа через ВТБ ID необходимо выполнить следующие настройки:

1. Зарегистрировать приложение в системе ВТБ ID. Для этого подать заявку на официальном сайте этого поставщика идентификации[56]. 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.

[56] См.: https://developer.vtb.ru/connection

3. Заполнить настройки поставщика идентификации (см. Рисунок 128):
   - Идентификатор поставщика;
   - Название поставщика;
   - URL для авторизации – адрес, по которому должна инициироваться аутентификация, например: https://id.vtb.ru;
   - URL для получения и обновления маркера – адрес, по которому происходит получение и обновление маркера доступа, например: https://id.vtb.ru/oauth2/token;
   - 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 (см. п. 8.18).
5. В разделе «Аутентификация» консоли управления включить использование метода аутентификации с использованием поставщика идентификации ВТБ ID (см. п. 4.6).

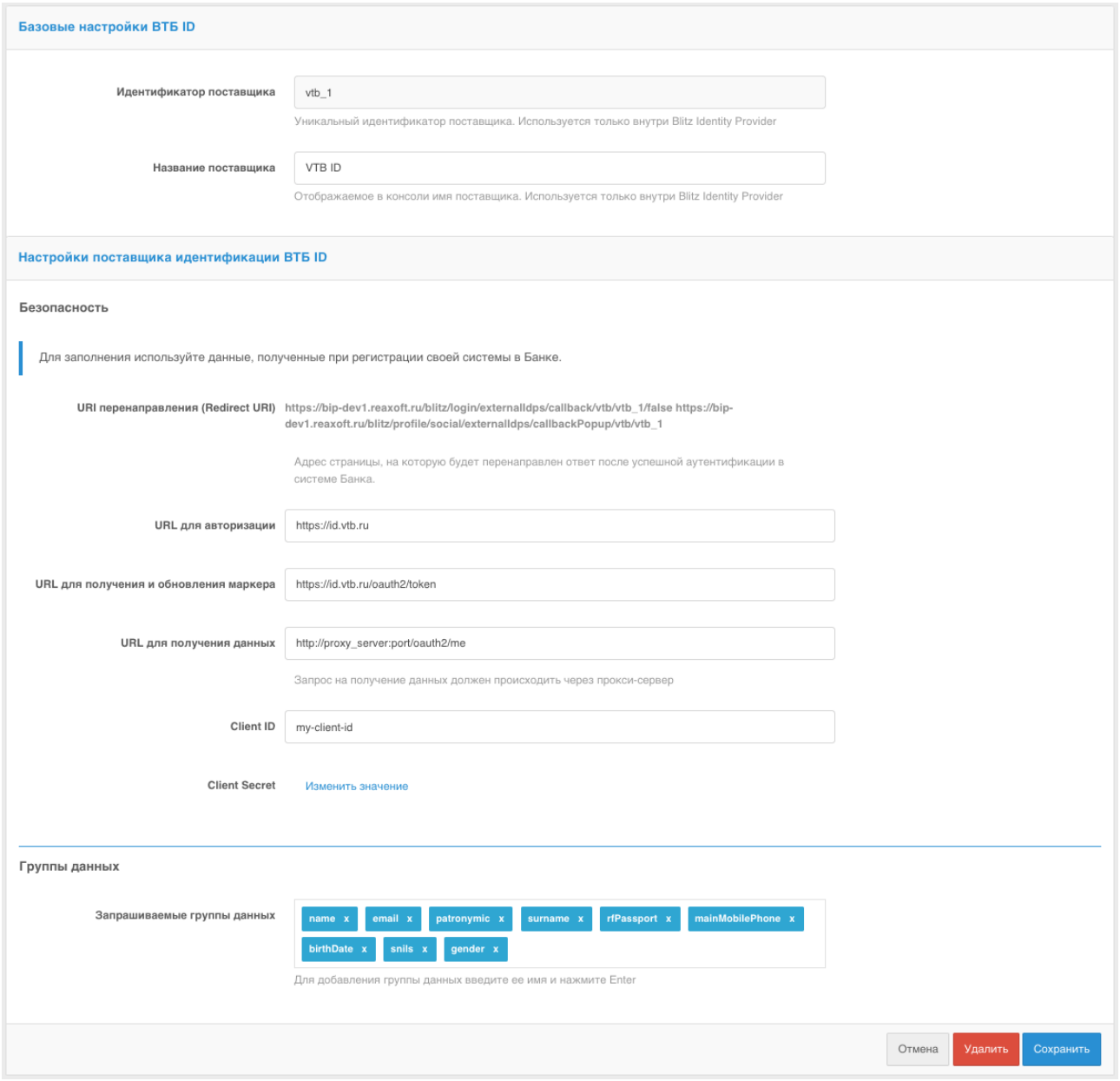

Рисунок 128 – Настройки поставщика идентификации ВТБ ID

8.14. Вход через систему идентификации СберБизнес ID

Для конфигурирования входа через СберБизнес ID необходимо выполнить следующие настройки:

1. Зарегистрировать приложение в системе СберБизнес ID. Для этого подать заявку на официальном сайте этого поставщика идентификации[57]. По результатам регистрации необходимо получить:
   - идентификатор клиента (Client ID);
   - секрет клиента (Client Secret).

[57] См.: https://api.developer.sber.ru/how-to-use/sberbusiness_id

2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип СберБизнес ID.
3. Перейти в Blitz Identity Provider и заполнить настройки поставщика идентификации (см. Рисунок 129), которые включают в себя:
   - 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;
   - 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 (см. п. 8.18).
5. В разделе «Аутентификация» консоли управления включить использование метода аутентификации с использованием поставщика идентификации СберБизнес ID (см. п. 4.6).

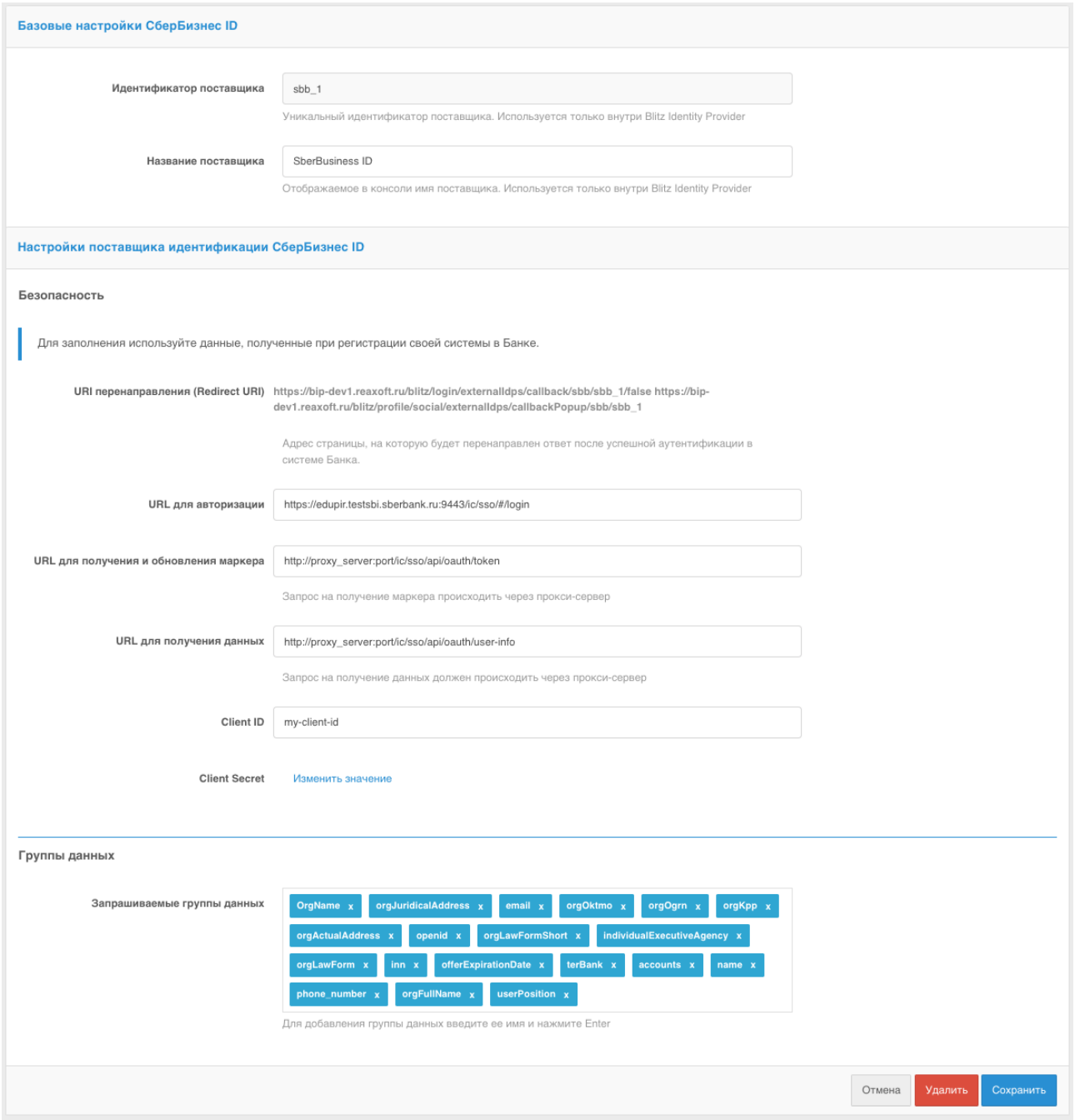

Рисунок 129 – Настройки поставщика идентификации СберБизнес ID

8.15. Вход через систему идентификации Альфа ID

Для конфигурирования входа через Альфа ID необходимо выполнить следующие настройки:

1. Зарегистрировать приложение в системе Альфа ID. Для этого подать заявку на официальном сайте этого поставщика идентификации[58]. По результатам регистрации необходимо получить:
   - идентификатор клиента (Client ID);

[58] См.: https://alfabank.ru/sme/alfaid/

- секрет клиента (Client Secret).

2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип Альфа ID.
3. Заполнить настройки поставщика идентификации (см. Рисунок 130):
   - 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;
   - 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 (см. п. 8.18).
5. В разделе «Аутентификация» консоли управления включить использование метода аутентификации с использованием поставщика идентификации Альфа ID (см. п. 4.6).

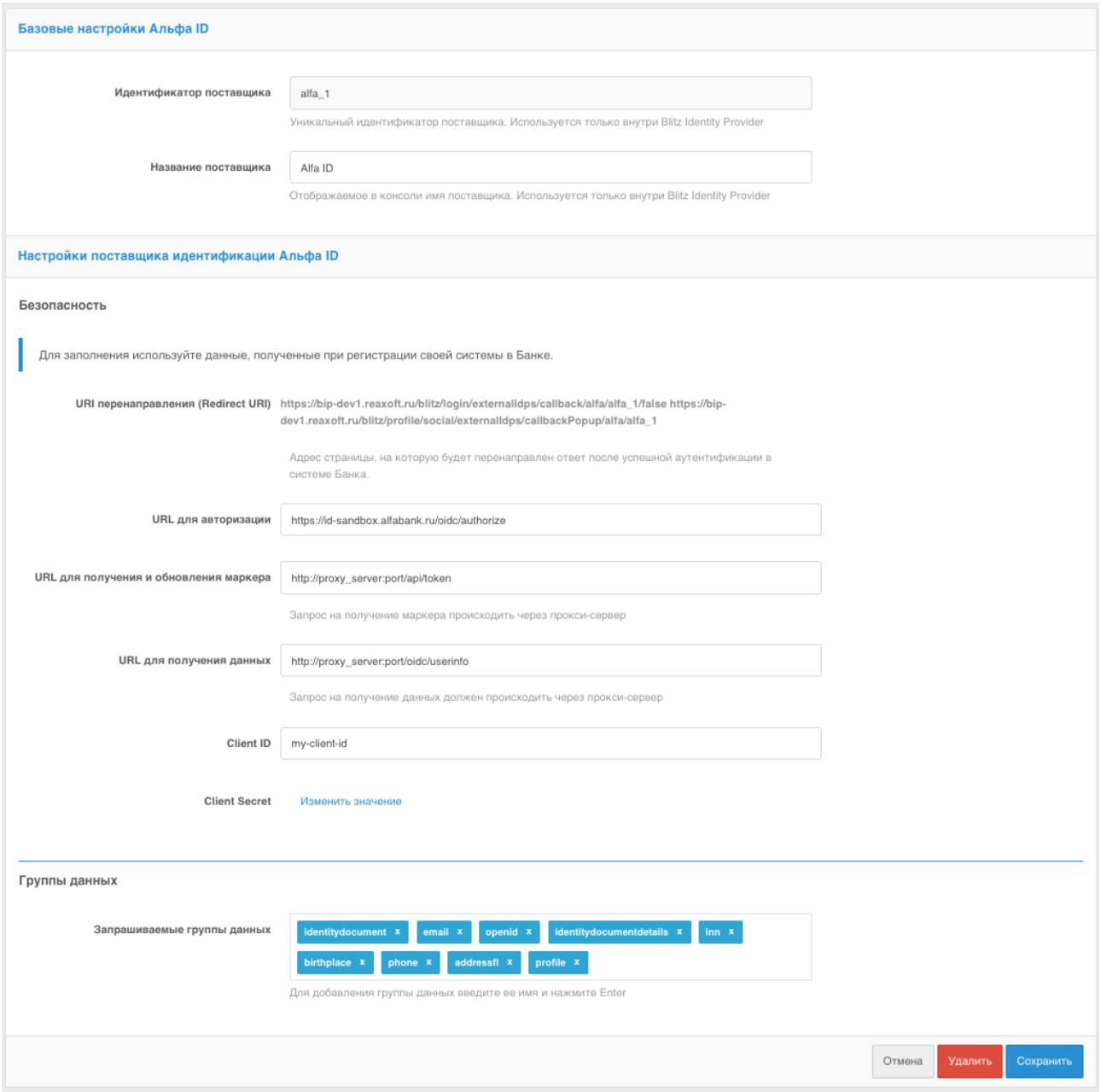

Рисунок 130 – Настройки поставщика идентификации Альфа ID

8.16. Вход через Mos ID (СУДИР)

Для конфигурирования входа через Mos ID (СУДИР) необходимо выполнить следующие настройки:

1. Зарегистрировать приложение в системе СУДИР. Для этого подать заявку согласно инструкции, размещенной на официальном сайте этого поставщика идентификации[59]. По результатам регистрации необходимо получить:
   - идентификатор (client_id);
   - секрет (client_secret).

[59] См.: https://login.mos.ru/support (внешний СУДИР) и https://sudir.mos.ru/support (внутренний СУДИР)

2. Перейти в консоль управления Blitz Identity Provider и добавить поставщика, имеющего тип СУДИР.
3. Заполнить настройки поставщика идентификации (см. Рисунок 131):
   - 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;
   - 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 (client_secret);
   - Запрашиваемые разрешения – перечень запрашиваемых разрешений, например, openid и profile;
   - Идентификатор – укажите имя атрибута в СУДИР, который должен использоваться в качестве уникального идентификатора учетной записи. Для внешнего СУДИР это guid, для внутреннего СУДИР это uid.
4. Настроить правила связывания (см. п. 8.18).
5. В разделе «Аутентификация» консоли управления включить использование метода аутентификации с использованием поставщика идентификации СУДИР (см. п. 4.6).

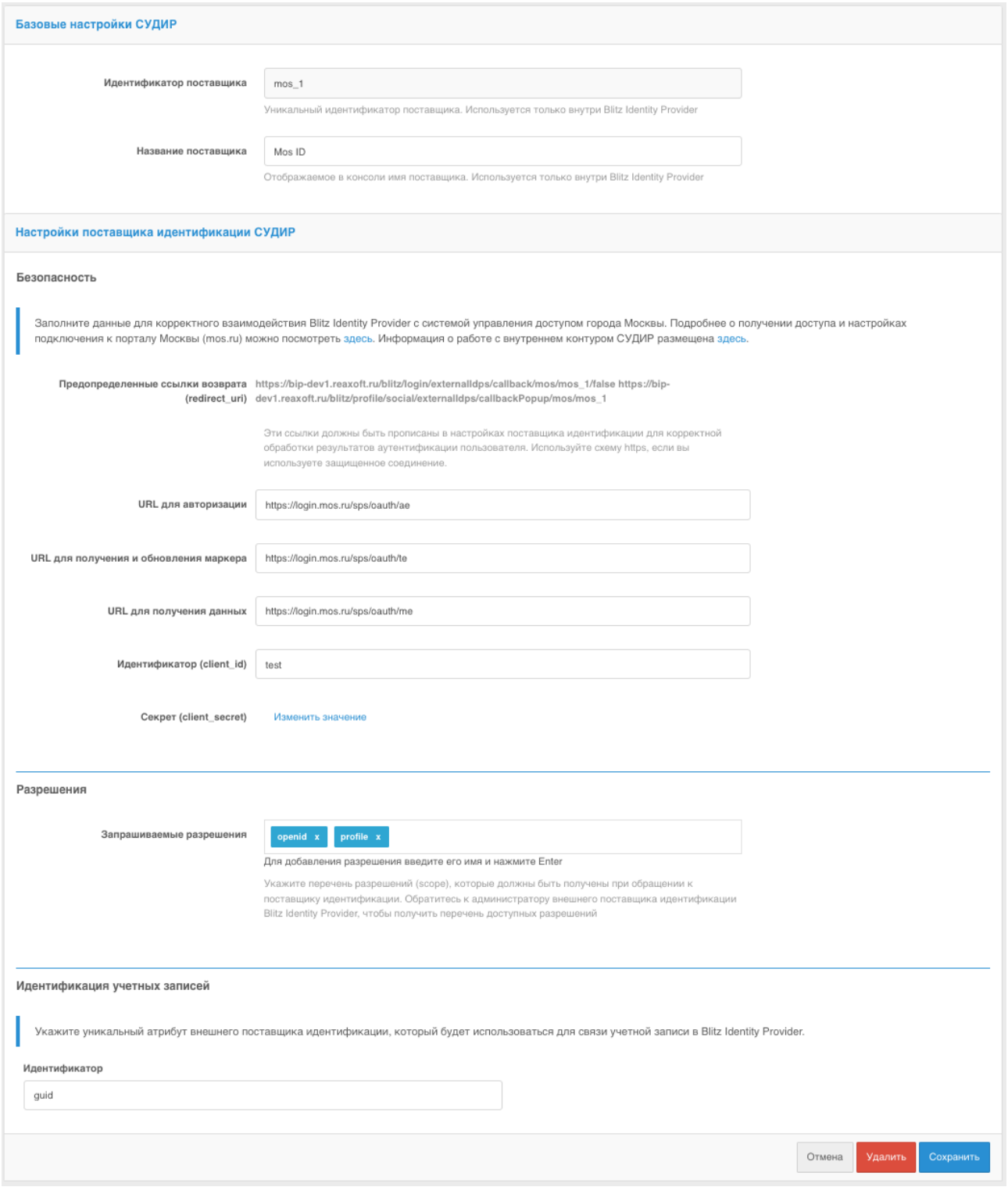

Рисунок 131 – Настройки поставщика идентификации СУДИР

8.17. 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 настройки:

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 основного 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 дополнительные настройки поставщика идентификации (см. Рисунок 132):
   - Идентификатор поставщика;
   - Название поставщика;
   - URL для авторизации – адрес доверенного Blitz Identity Provider, по которому можно получить код авторизации;
   - URL для получения и обновления маркера – адрес доверенного Blitz Identity Provider, по которому можно получать маркеры доступа;
   - URL для получения данных – адрес доверенного Blitz Identity Provider, по которому можно получать данные пользователя;
   - 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 (см. п. 8.18).
5. В разделе «Аутентификация» консоли управления включить использование метода аутентификации с использованием поставщика идентификации Blitz Identity Provider (см. п 4.6).

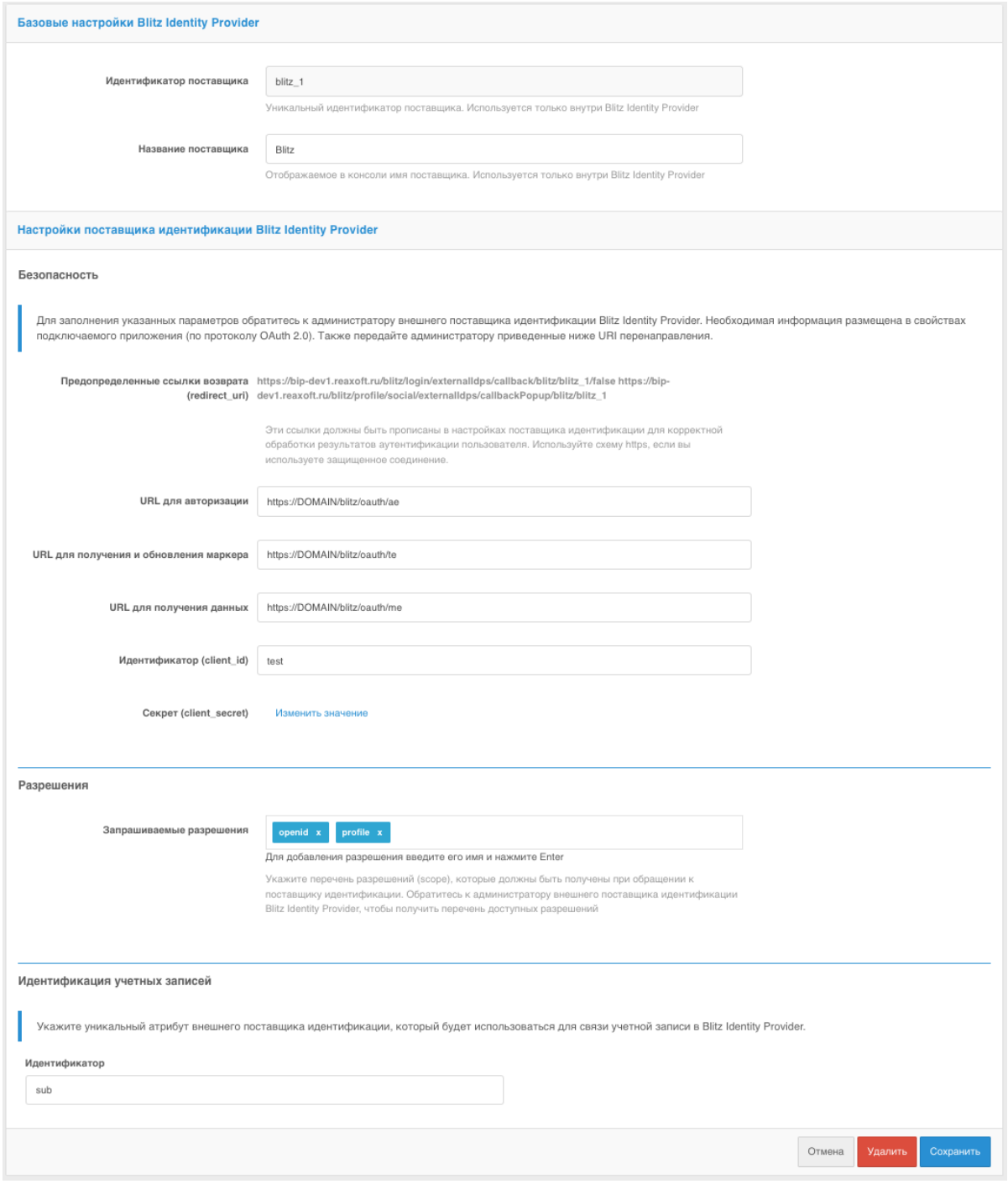

Рисунок 132 – Настройки подключения к внешнему поставщику идентификации Blitz Identity Provider

8.18. 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 настройка (см. Рисунок 133) – настройка выполняется с помощью конструктора правил. Данный режим подходит для типовых сценариев связывания учетных записей и сопоставления атрибутов.
- Расширенная настройка (см. Рисунок 134) – 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 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.

В режиме базовой настройки предусмотрены следующие настройки:

- Опция «Разрешить привязывать одну учетную запись поставщика идентификации ко многим аккаунтам Blitz Identity Provider»:
  - 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.
- Опция «Предлагать пользователю ввести логин и пароль для привязки, если учетная запись не была идентифицирована»:
  - опция выбрана – пользователю, будет предложено пройти идентификацию и аутентификацию альтернативным способом, чтобы привязать внешнюю учетную

запись, если по настроенным правилам не удалось найти учетную запись в Blitz Identity Provider.

- опция не выбрана – Blitz Identity Provider не разрешит вход пользователя, для которого не удалось сопоставить учетные записи. Если настроен процесс регистрация для внешних учетных записей, то будет автоматически запущен процесс регистрации.

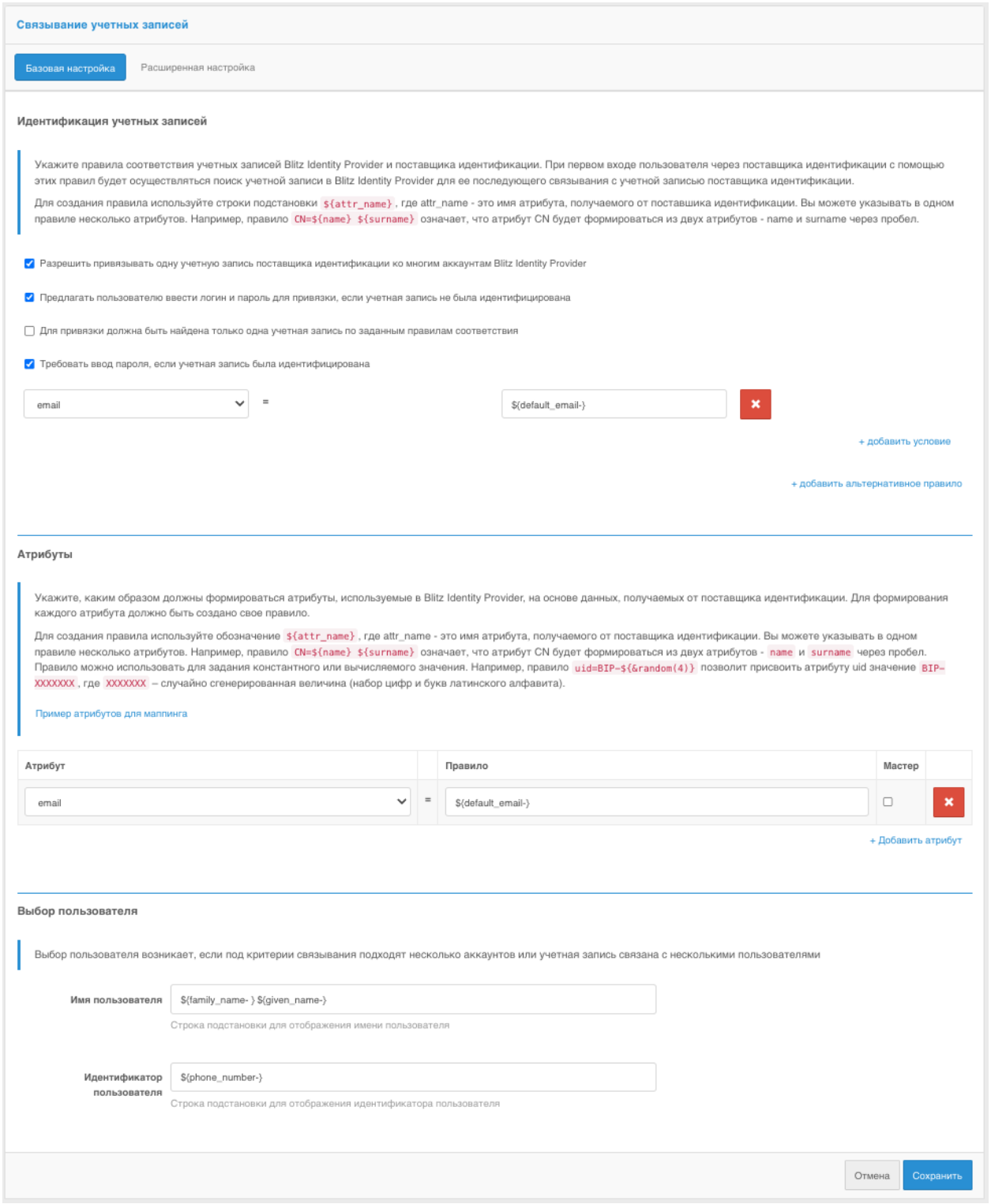

Рисунок 133 – Базовая настройка правил связывания

- Опция «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.
- Опция «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»).
- Блок «Атрибуты» с правилами сохранения атрибутов. Например, правило email=${default_email} означает, что атрибут с именем email в Blitz Identity Provider будет заполняться значением из атрибута default_email внешней учетной записи (для пользователей, воспользовавшихся этим поставщиком идентификации). Если у атрибута отмечен чекбокс «Мастер», то заполнение или обновление атрибута будет происходить при каждом входе через внешний поставщик идентификации. Если чекбокс «Мастер» не отмечен, то заполнение произойдет только при первом входе, в результате которого возникло связывание учетных записей.
- Блок «Выбор пользователя» определяет правила отображения пользователю найденной по настроенным правилам соответствия учетной записи в Blitz Identity Provider. Настройка «Имя пользователя» определяет информацию, отображаемую в верхней строке карточки пользователя (строке, предназначенной для отображения имени учетной записи). Например, ${family_name- } ${given_name-} определяет, что

пользователю в верхней строке будут показаны фамилия и имя (если они заполнены). Настройка «Идентификатор пользователя» определяет информацию, отображаемую в нижней строке карточки пользователя (строке, предназначенной для отображения идентификатора учетной записи). При настройке можно использовать маскирование значений. Например, правило `${phone_number&maskInMiddle(3,3)}` будет отображать средние числа номера телефона в виде `*`.

Рисунок 134 – Расширенная настройка правил связывания

9. Управление учетными записями пользователей

В разделе «Пользователи» консоли управления доступны следующие операции:

- поиск учетных записей пользователей;
- добавление учетной записи пользователя;
- просмотр и редактирование атрибутов учетной записи пользователя;
- сброс сессий пользователя;
- изменение пароля учетной записи пользователя;
- просмотр и отвязка учетных записей внешних поставщиков идентификации;
- привязка устройств для проведения двухфакторной аутентификации;
- просмотр групп, в которые включен пользователь, управление членством пользователя в группах;
- просмотр и удаление устройств пользователя;
- просмотр, привязка, удаление ключей безопасности пользователя;
- просмотр прав учетной записи пользователя, назначение и отзыв прав;
- просмотр разрешений, выданных пользователем приложениям;
- удаление учетной записи пользователя.

Общий вид страницы управления данными пользователей представлен на рисунке 135.

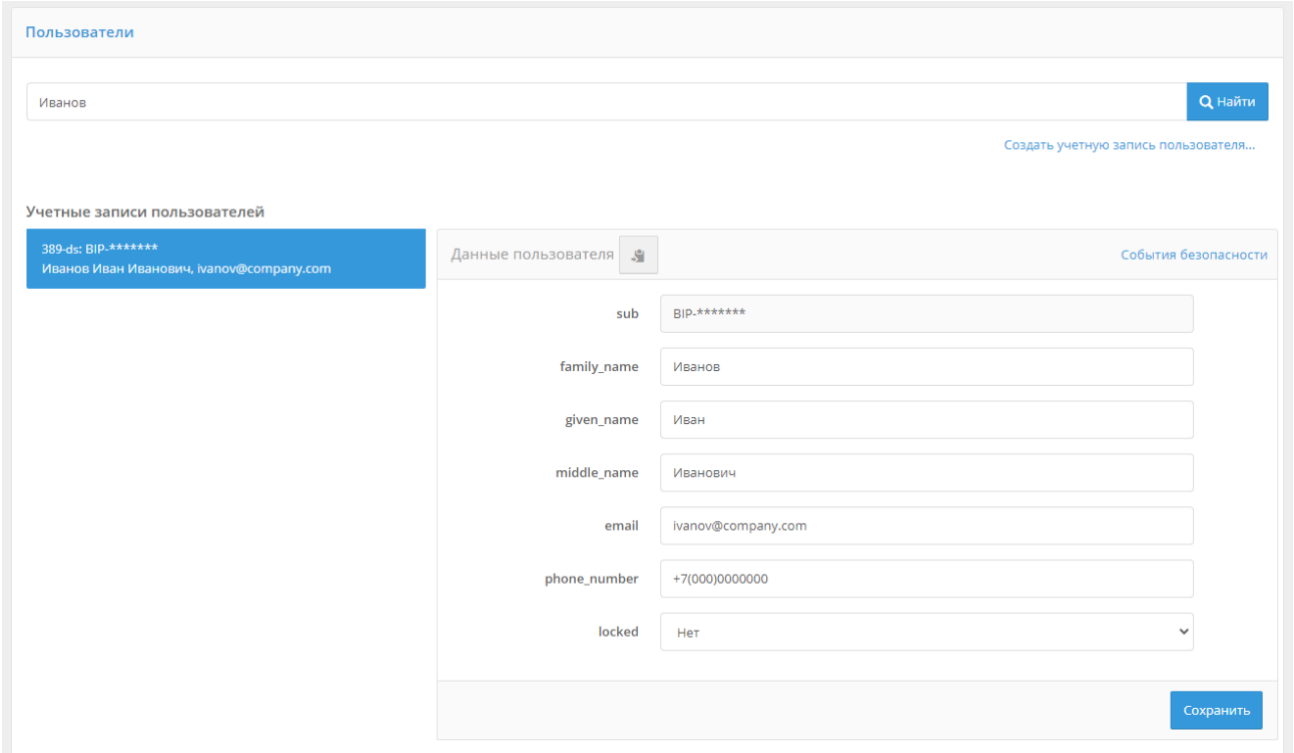

Рисунок 135 – Вид страницы управления пользователями

9.1. Поиск учетных записей 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.

9.2. Добавление учетной записи пользователя

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

- указать хранилище, в которым следует сохранить данные пользователя;
- задать все необходимые атрибуты;
- нажать на кнопку «Создать».

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 «звезда» (*)).

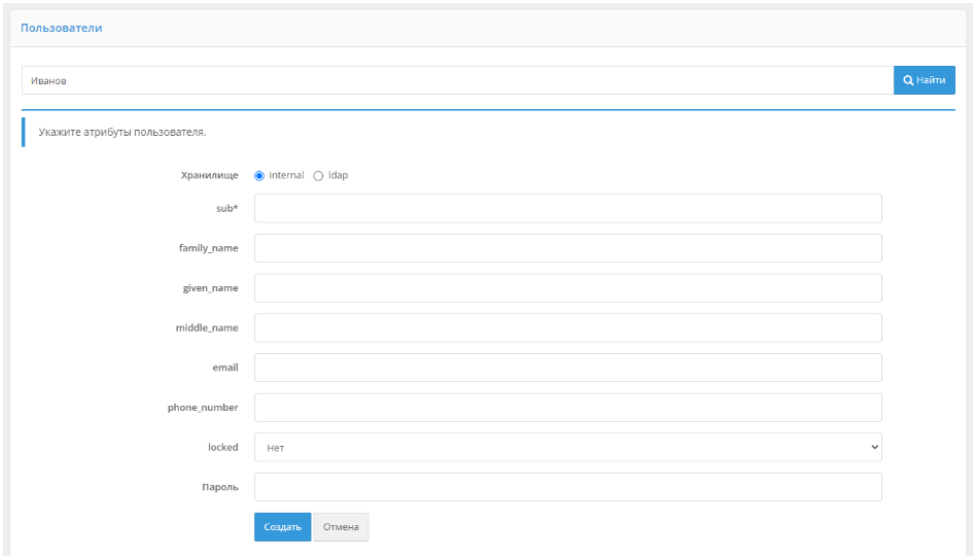

Рисунок 136 – Создание учетной записи пользователя

9.3. 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.

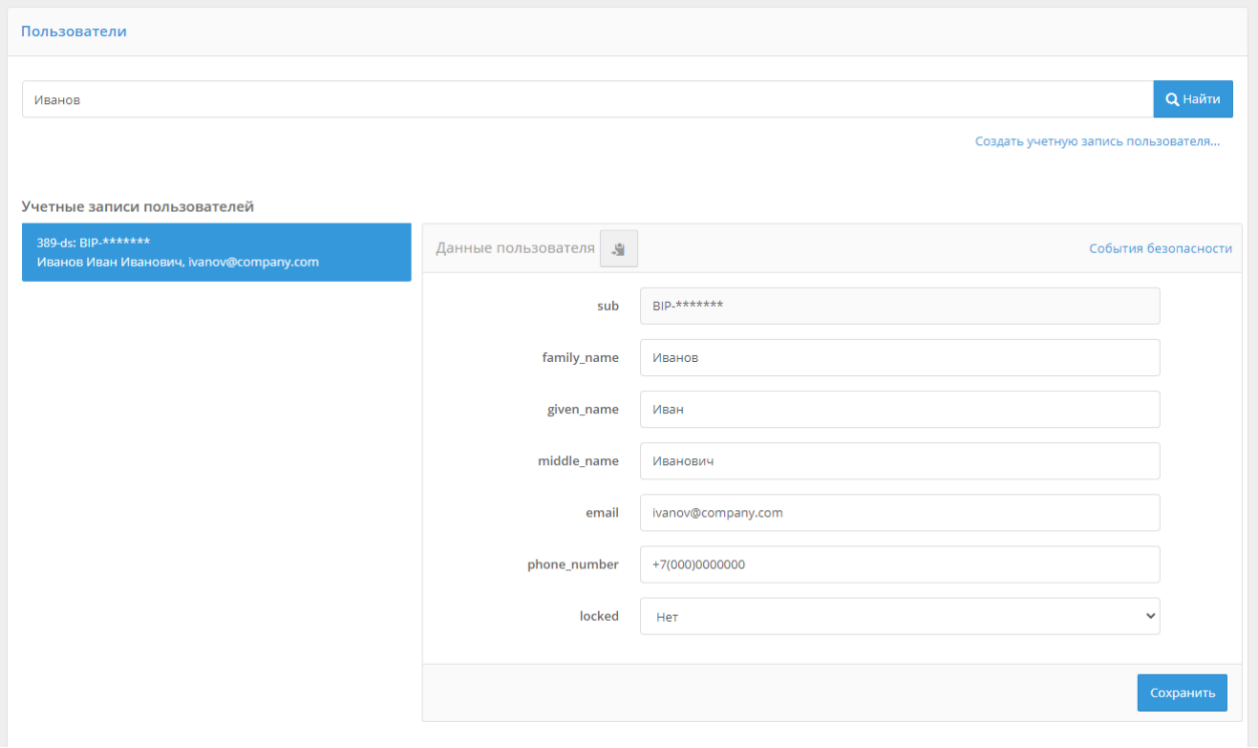

Рисунок 137 – Просмотр информации о пользователе (фрагмент)

На карточке пользователя можно совершать следующие операции:

- редактировать атрибуты пользователя;
- сбросить сессии пользователя;

- изменять пароль;
- просматривать перечень привязанных учетных записей внешних поставщиков идентификации, отвязывать внешние учетные записи;
- изменять требуемый уровень аутентификации для пользователя;
- привязывать или удалять устройства для проведения аутентификации: генераторы разовых паролей (см. п. 9.3.5) и мобильные приложения для получения push-уведомлений (см. п. 9.3.6);
- просматривать группы, в которые включен пользователь (см. п. 9.3.7);
- просматривать права пользователя и права, которые имеются в отношении данного пользователя (см. п. 9.3.8);
- просматривать и удалять запомненные устройства и браузеры пользователя;
- просматривать, добавлять и удалять ключи безопасности пользователя;
- просматривать и удалять выданные приложениям разрешения.

9.3.1. Редактирование атрибутов пользователя

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.

9.3.2. Сброс сессий пользователя

Для сброса сессий пользователя используется кнопка «Сбросить сессии» в блоке «Сброс сессий пользователя».

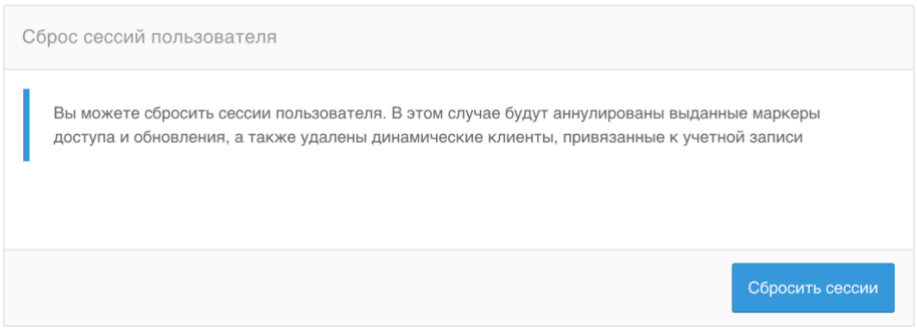

Рисунок 138 – Сброс сессий пользователя

При сбросе сессий пользователя выполняются следующие действия:

- выданные на пользователя приложениям маркеры безопасности (маркеры доступа, маркеры обновления, маркеры идентификации) становятся недействительными – при вызове в Blitz Identity Provider сервиса интроспекции с такими маркерами сервис вернет, что маркер недействителен;
- в запомненных для пользователя устройствах убираются флаги доверенных устройств и запоминания на них длительных сессий;
- привязанные к учетной записи пользователя выпущенные для мобильных приложений пары динамических client_id/client_secret аннулируются;
- запомненные в браузере пользователя SSO-сессии становятся недействительными, так что при очередном запросе со стороны приложений идентификации в Blitz Identity Provider будет запрошена новая идентификация и аутентификация.

9.3.3. Смена пароля пользователя

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.

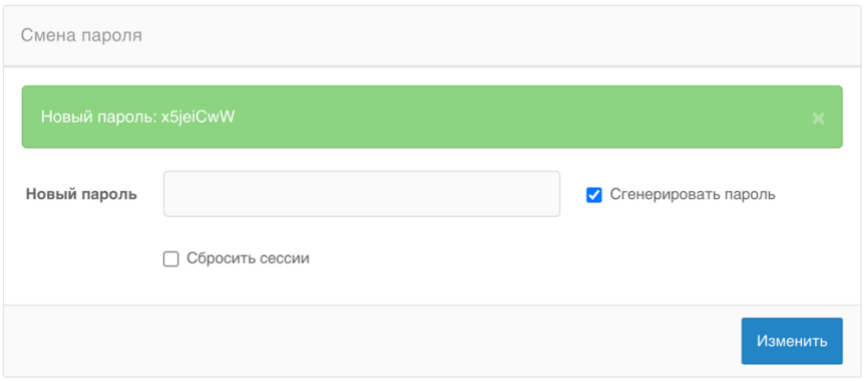

Рисунок 139 – Смена пароля

9.3.4. Просмотр и отвязка привязанных учетных записей внешних поставщиков идентификации

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.

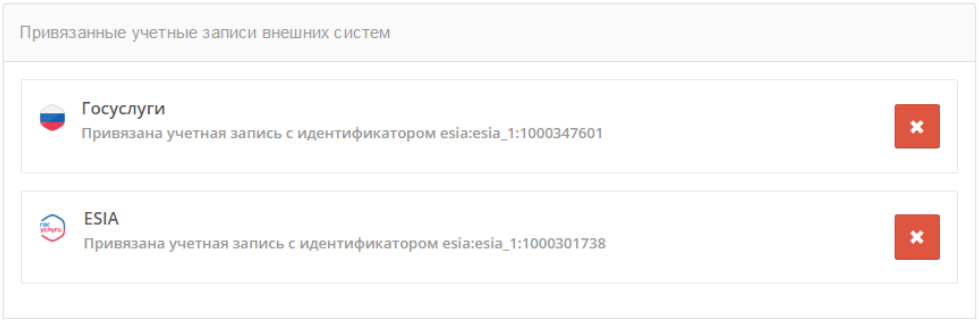

Рисунок 140 – Просмотр информации о пользователе:
привязанные учетные записи внешних поставщиков

9.3.5. Привязка устройств для проведения двухфакторной аутентификации 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 (см. Рисунок 141) либо привязать к учетной записи по QR-коду мобильное приложение, осуществляющее выработку TOTP-кодов (см. Рисунок 142).

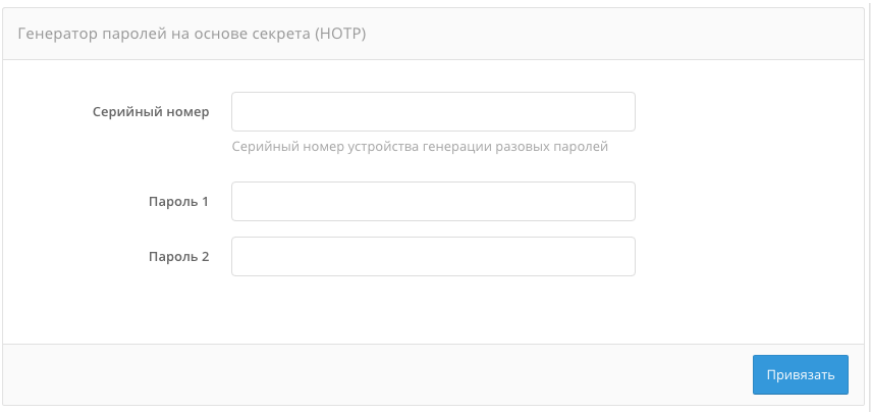

Рисунок 141 – Привязка HOTP-устройства по серийному номеру администратором

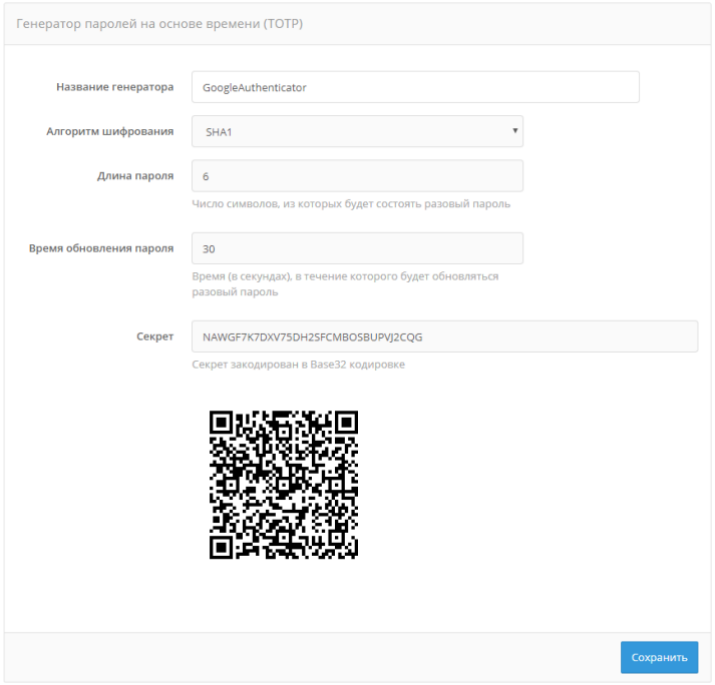

Рисунок 142 – Привязка TOTP-приложения по QR-коду администратором

9.3.6. Привязка мобильного приложения 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 отсканировать отображенный QR-код мобильным приложением Duo Mobile.

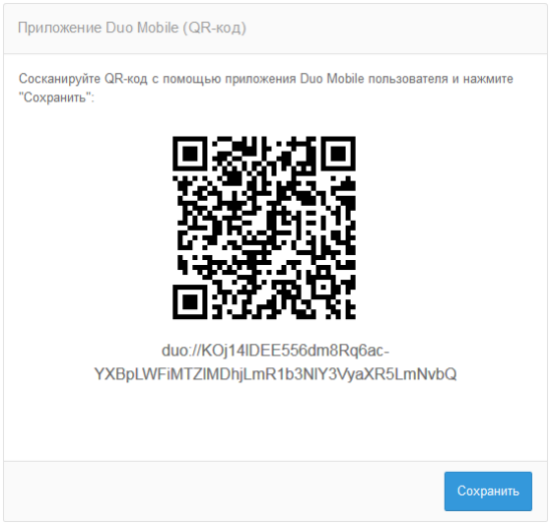

Рисунок 143 – Привязка мобильного приложения Duo Mobile

9.3.7. Просмотр групп, в которые включен пользователь, управление членством пользователя в группах

Если пользователь включен в группы, то эта информация будет отображена в блоке «Членство в группах» (см. Рисунок 144). По каждой группе будет отображены следующие данные:

- идентификатор группы;
- значения атрибутов группы.

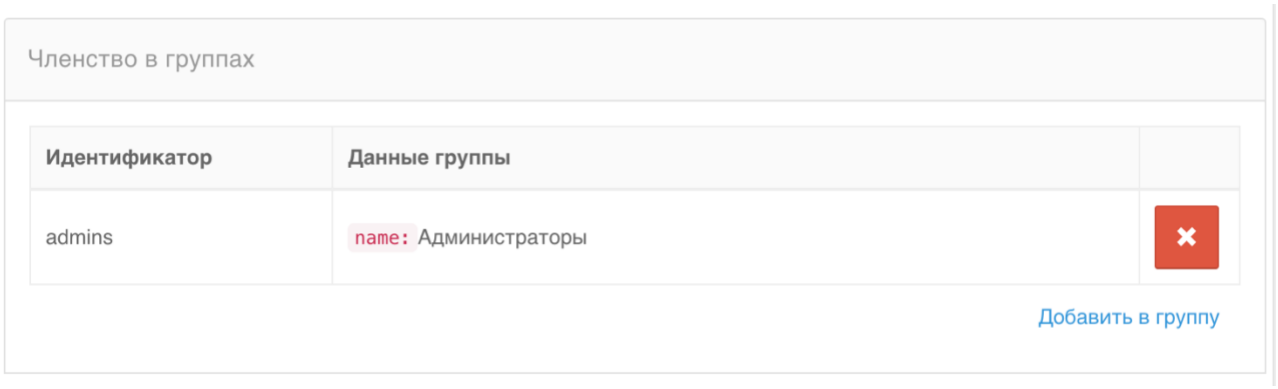

Рисунок 144 – Просмотр групп пользователя

Можно исключить пользователя из группы с помощью кнопки удаления или добавить пользователя в другую группу с помощью ссылки «Добавить в группу» (см. Рисунок 145). 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».

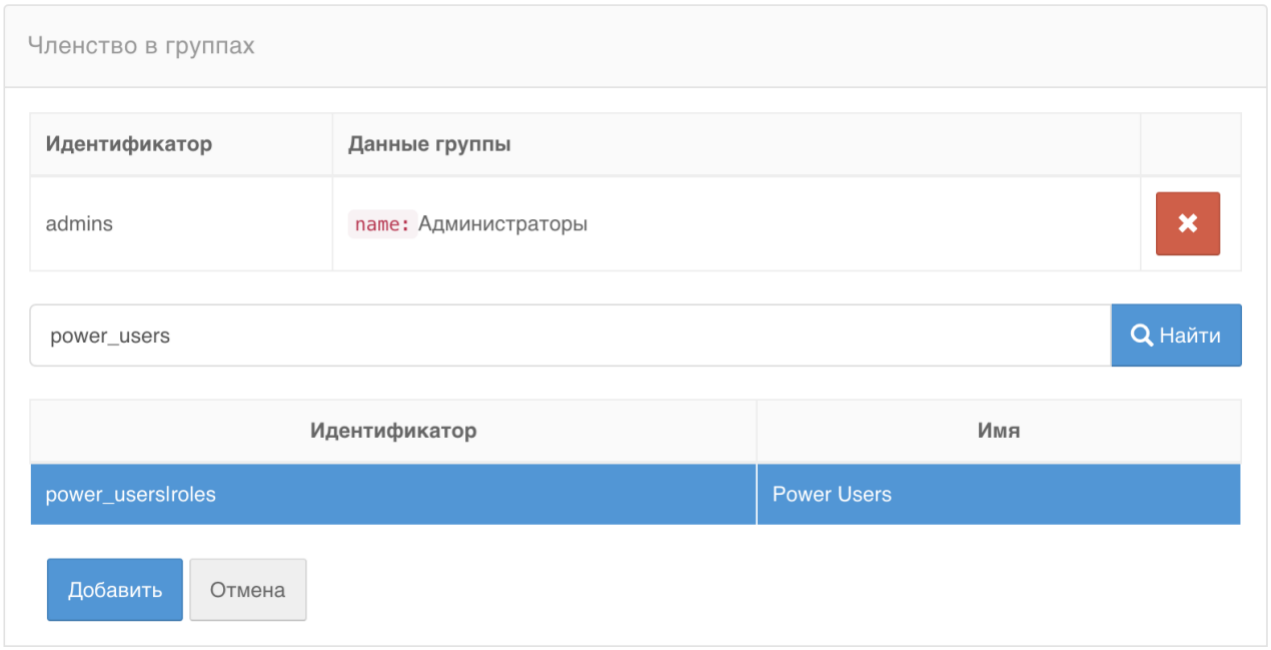

Рисунок 145 – Добавление пользователя в группу

9.3.8. Просмотр прав, 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» (см. Рисунок 146). Если пользователь имеет права в отношении объектов, например, других учетных записей, то это будет отображено в блоке «Права пользователя в отношении объектов» (см. Рисунок 147).

Каждое право характеризуется следующими параметрами:

- идентификатор объекта;
- имя;
- право.

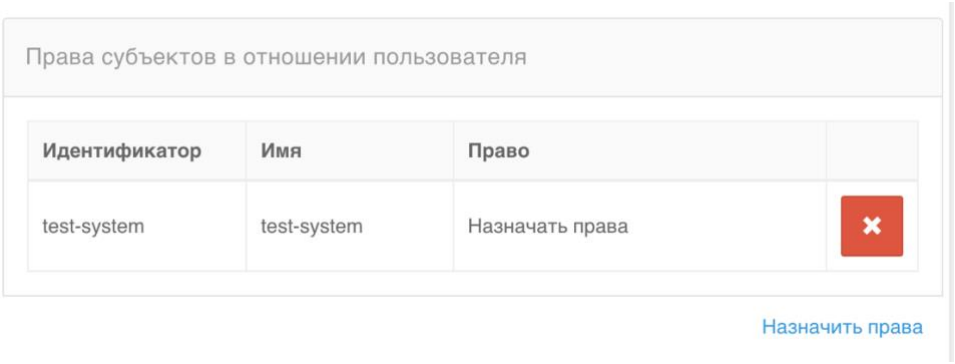

Рисунок 146 – Просмотр прав других субъектов в отношении пользователя

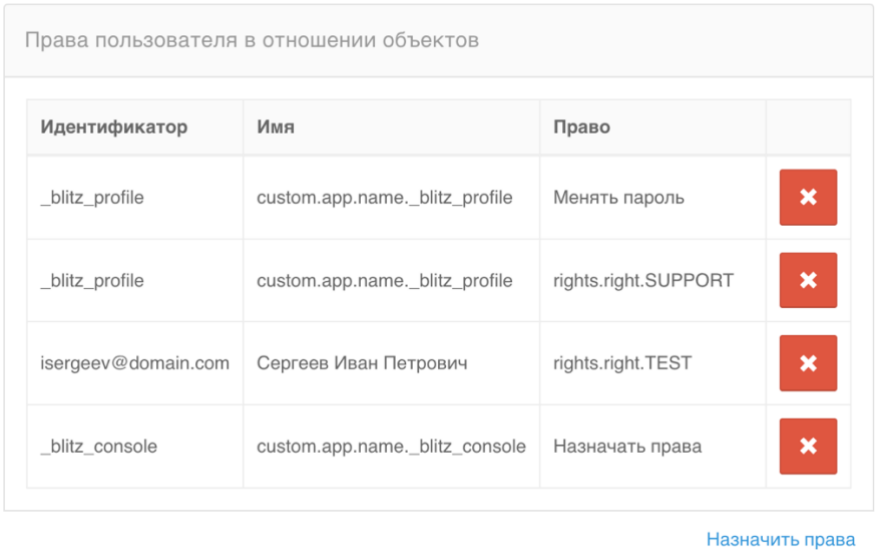

Рисунок 147 – Просмотр прав пользователя в отношении объектов

Отозвать право доступа можно с помощью кнопки удаления рядом с правом доступа (см. Рисунок 148 и Рисунок 149). 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.

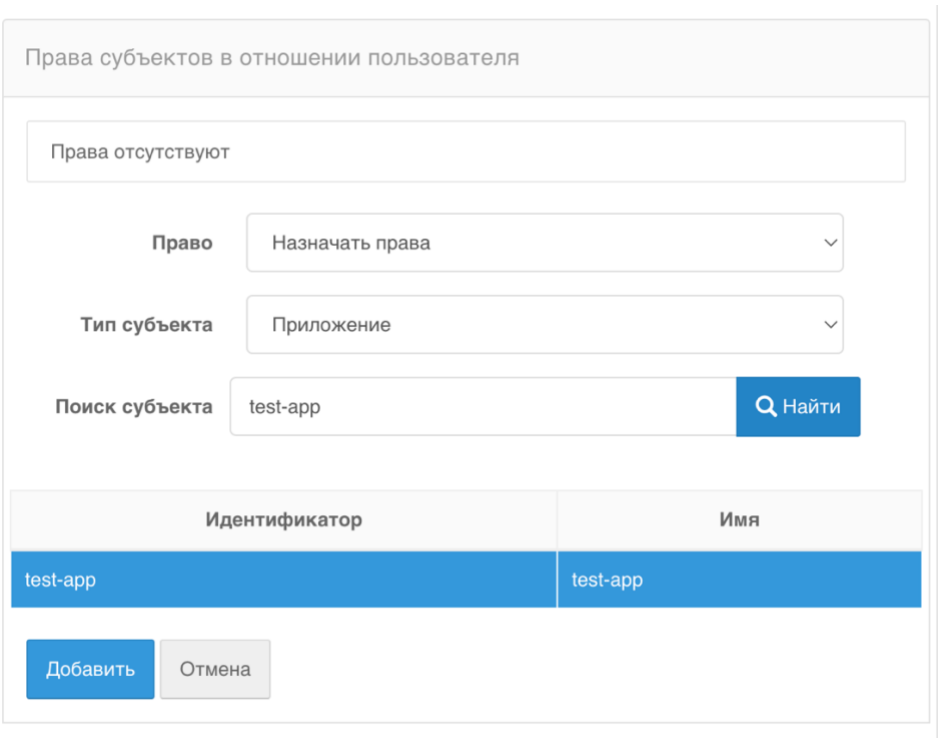

Рисунок 148 – Назначение другим субъектам прав к учетной записи пользователя

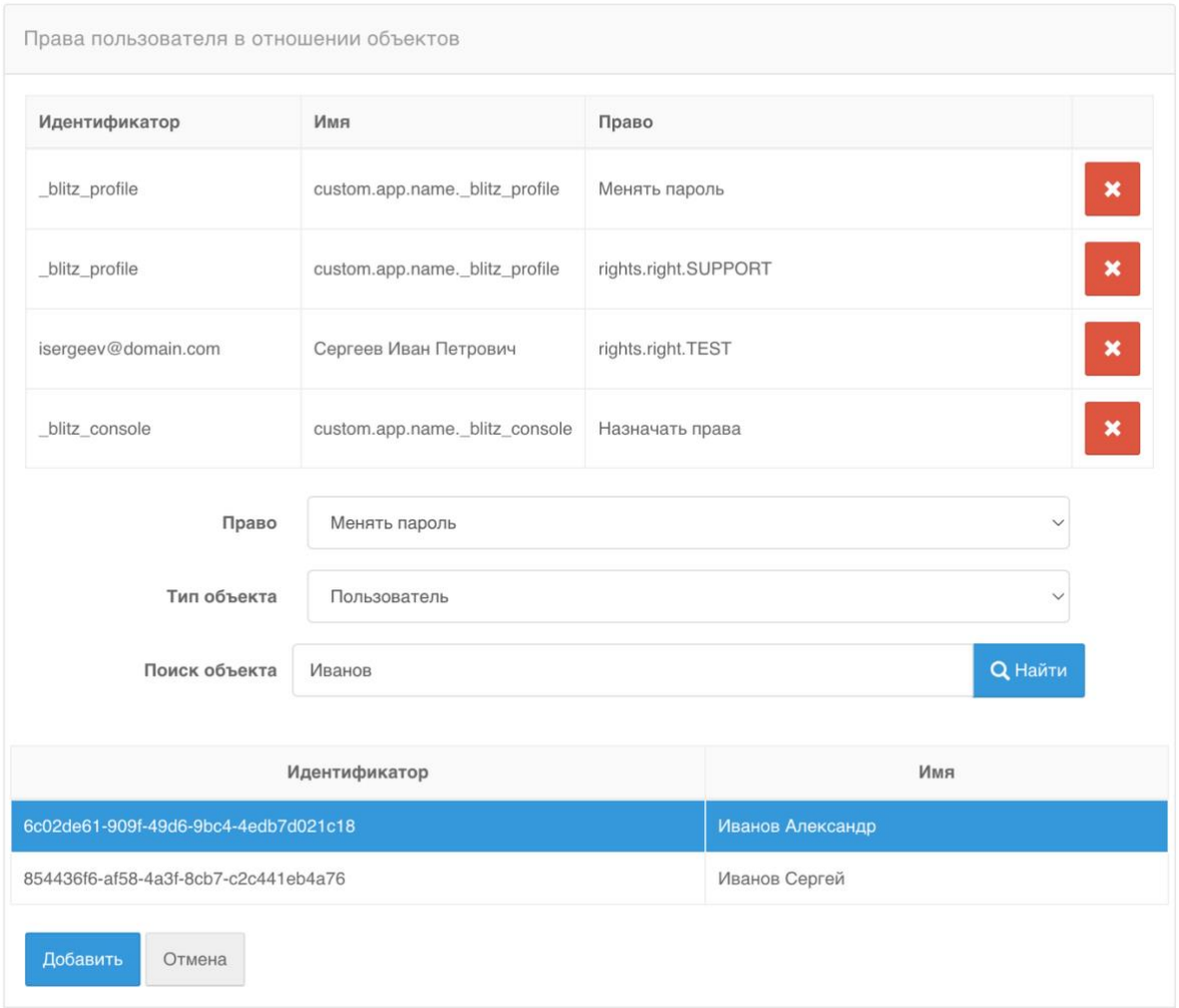

Рисунок 149 – Назначение пользователю прав к объектам

9.3.9. Просмотр и удаление запомненных устройств и браузеров

Администратор имеет возможность просмотреть устройства и браузеры, с которых пользователь осуществлял вход с использованием своей учетной записи (см. Рисунок 150). Описание устройств включает:

- признак того, запомнена ли на устройстве сессия входа и является ли устройство доверенным. Признак кодируется с помощью цвета:
  - серый – на устройстве не запомнена сессия входа и устройство не является доверенным;
  - желтый – на устройстве на запомнена сессия входа, но устройство является доверенным;
  - синий – на устройстве запомнена сессия входа, но устройство не является доверенным;
  - зеленый – на устройстве запомнена сессия входа и устройств является доверенным.
- имя и версия операционной системы устройства, определенные на основе UserAgent;
- имя и версия браузера, определенные на основе UserAgent;
- дата и время последнего входа с данного устройства и браузера;
- IP-адрес пользователя, который был определен при последнем входе с данного устройства и браузера.

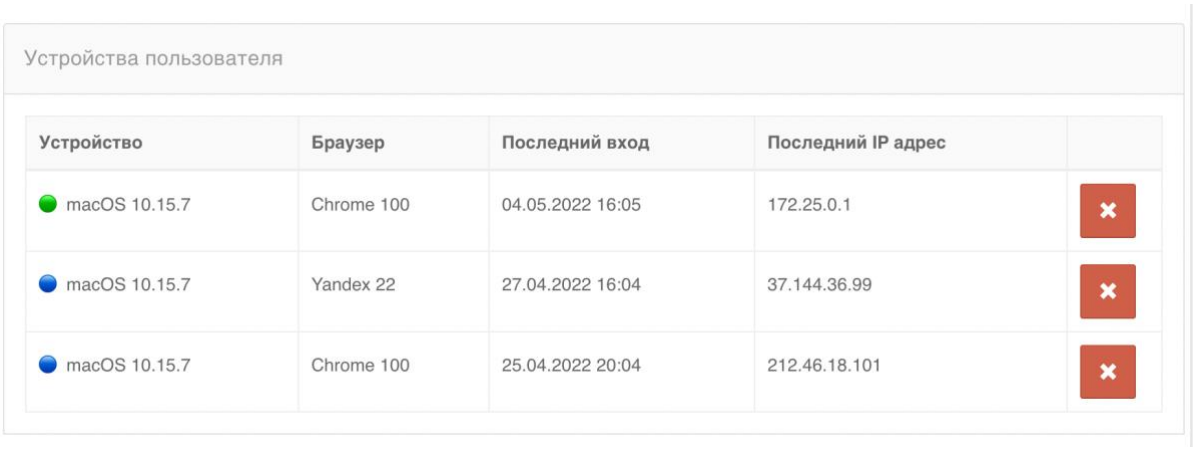

Рисунок 150 – Просмотр и удаление запомненных устройств и браузеров пользователя

9.3.10. Управление ключами безопасности

Администратор имеет возможность просмотреть перечень ключей безопасности (Passkey, WebAuthn, FIDO2, U2F), зарегистрированных для учетной записи пользователя (см. Рисунок 151). 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.

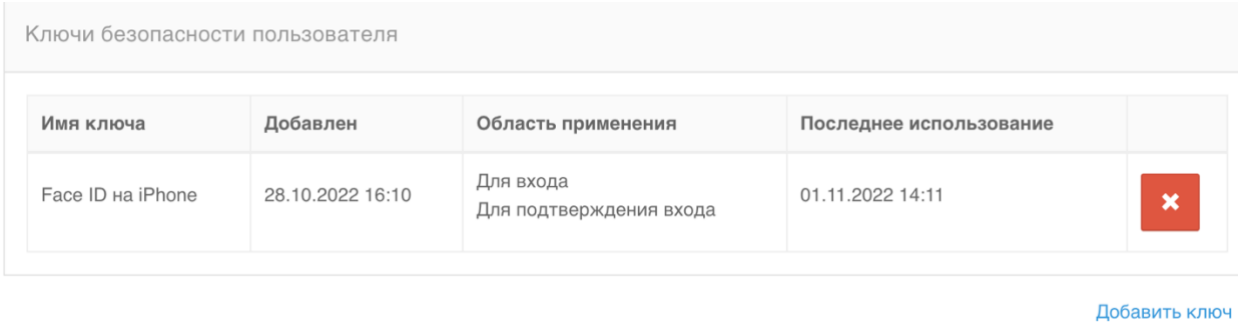

Рисунок 151 – Просмотр ключей безопасности пользователя

Администратор может зарегистрировать новый ключ безопасности с помощью ссылки «Добавить ключ» (см. Рисунок 152). В обычном сценарии использования ключи безопасности себе добавляет сам пользователь в момент входа (онбординг) или через личный кабинет.

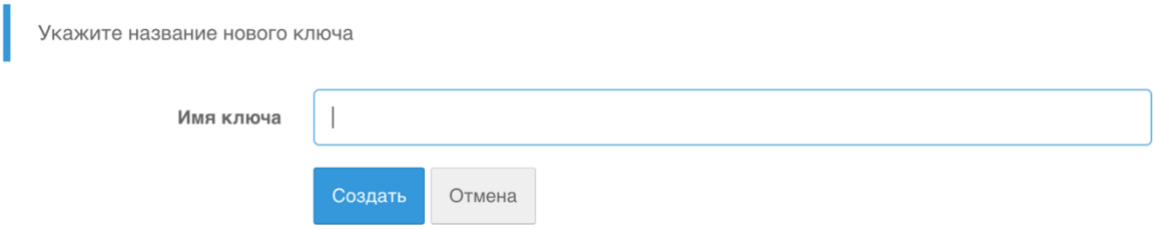

Рисунок 152 – Добавление ключа безопасности пользователя

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.

9.3.11. Просмотр и удаление выданных приложениям разрешений

Администратор имеет возможность просмотреть перечень разрешений, выданных пользователем приложениям (см. Рисунок 153). Каждое разрешение описывается:

- идентификатор приложения;
- перечень разрешений (scope);
- дата выдачи разрешений.

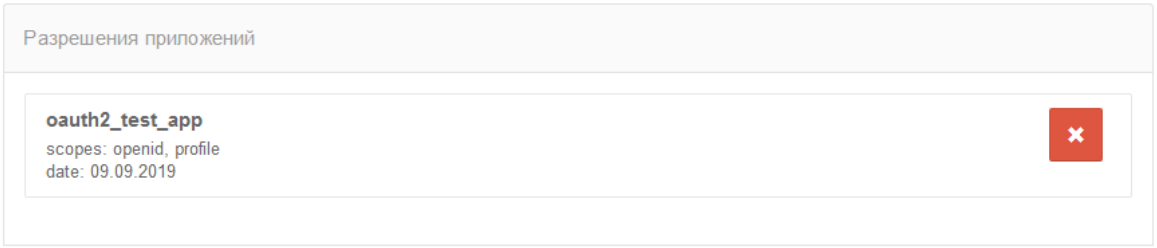

Рисунок 153 – Просмотр выданных разрешений

10. Управление группами пользователей

Если в Blitz Identity Provider настроена возможность работы с группами пользователей (см. п. 16.1.16), то в консоли управления появится раздел «Группы». В данном разделе 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 (см. Рисунок 154). 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»).

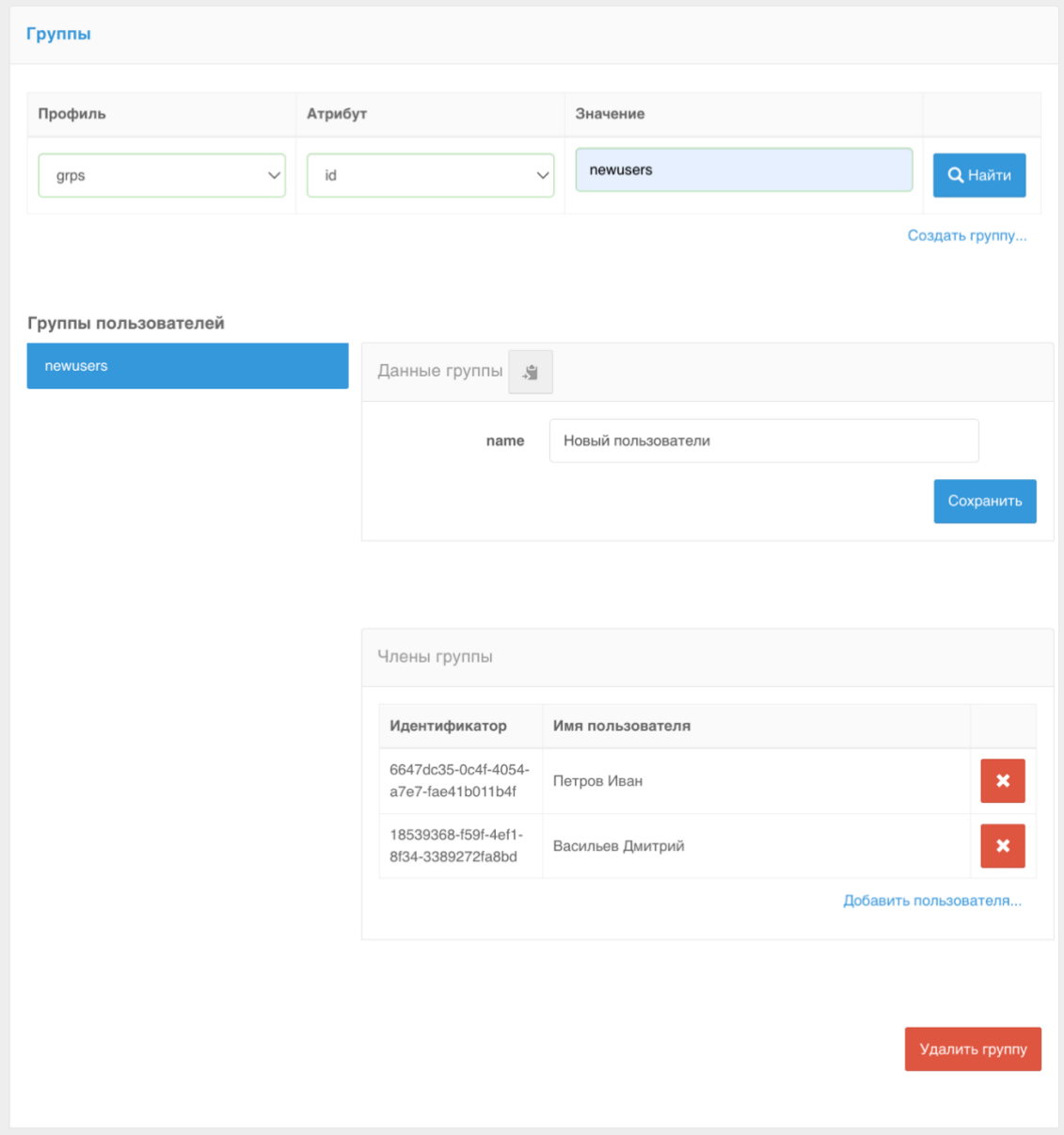

Рисунок 154 – Просмотр групп пользователей

Доступны возможности по редактированию атрибутов группы, удалению группы, включению пользователей в группу с помощью ссылки «Добавить пользователя…» (см. Рисунок 155), исключению пользователя из группы, созданию новых групп пользователей с помощью ссылки «Создать группу…» (см. Рисунок 156).

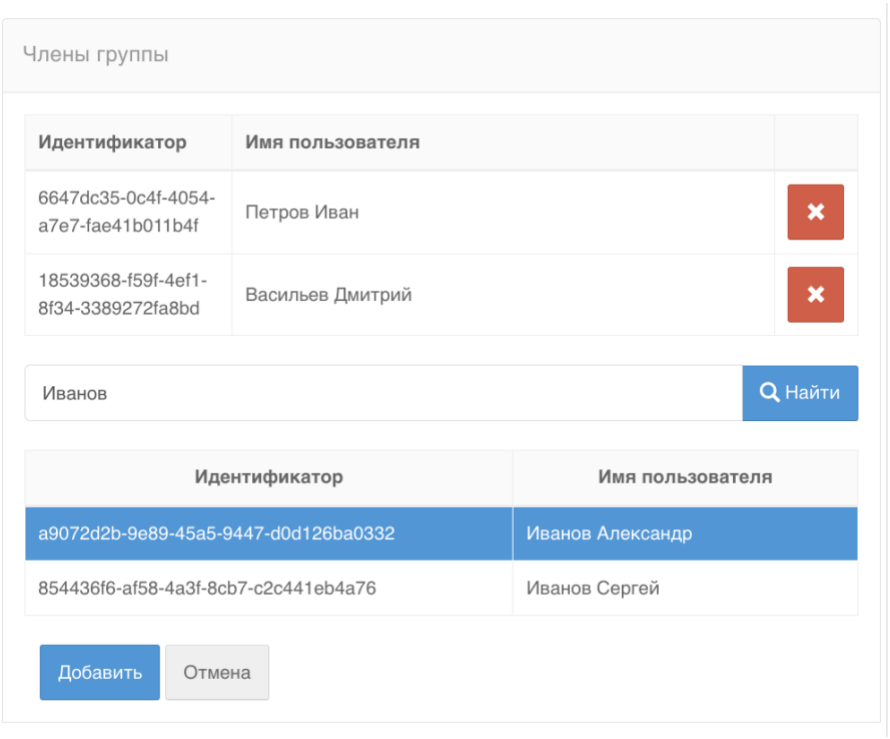

Рисунок 155 – Включение пользователя в группу

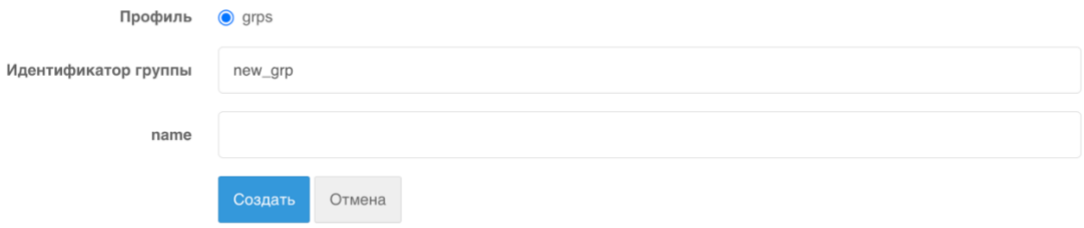

Рисунок 156 – Создание новой группы пользователей

11. Управление правами доступа

Для ведения справочника прав доступа в Blitz Identity Provider используется раздел «Права доступа» консоли управления (см. Рисунок 157). 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.

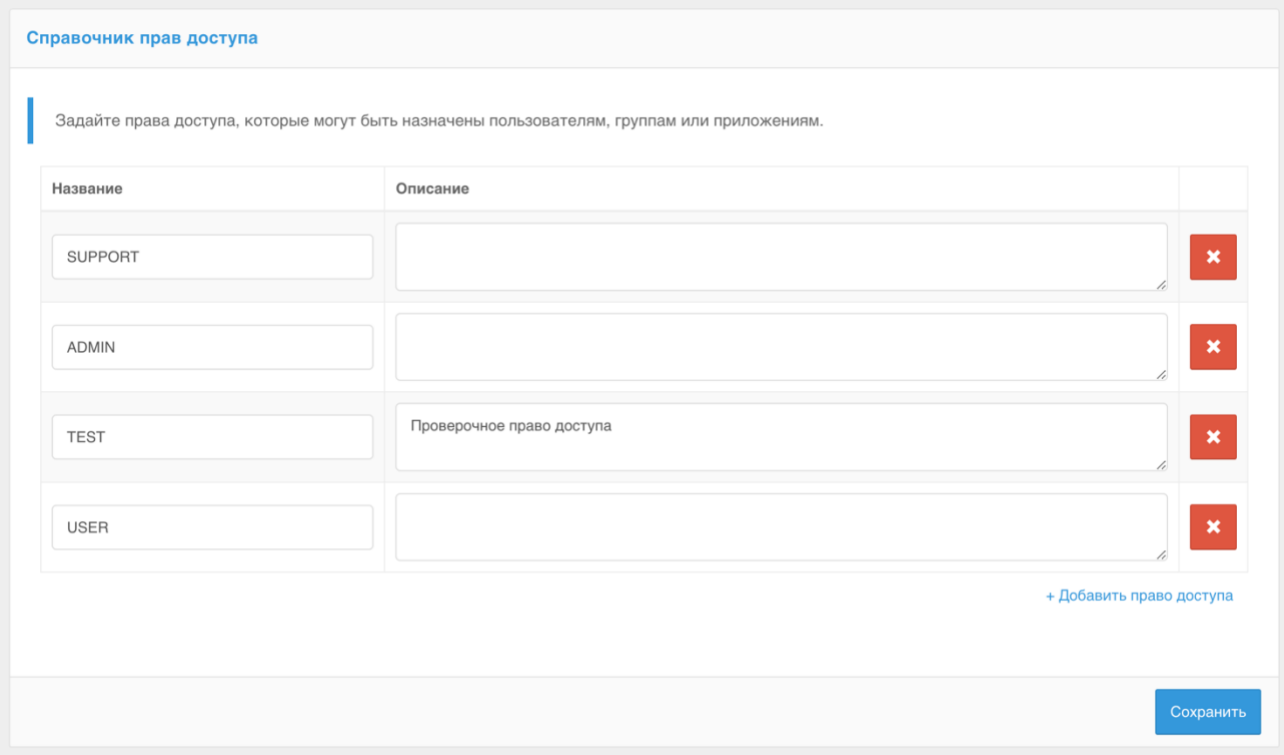

Рисунок 157 – Ведение справочника прав доступа

12. 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.

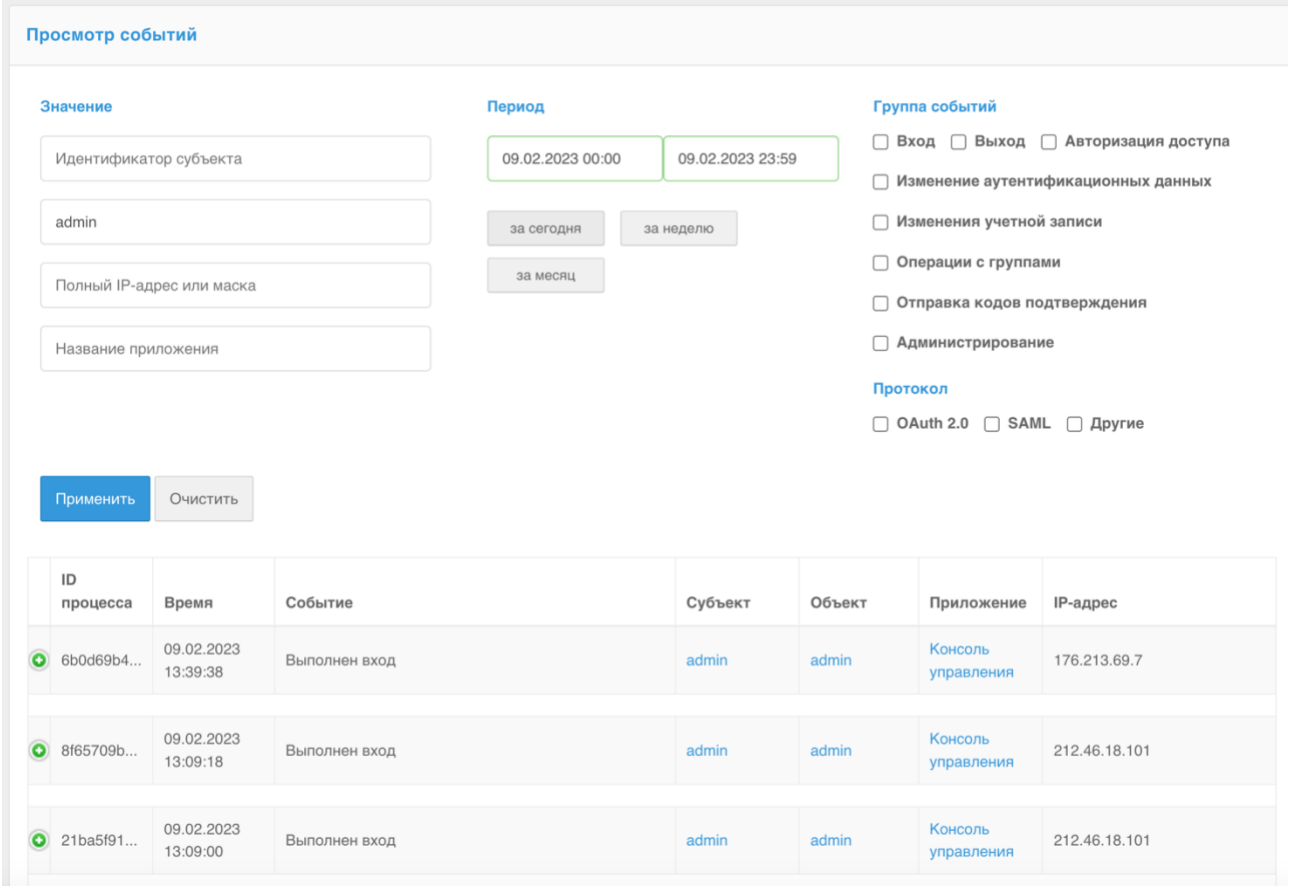

Рисунок 158 – Просмотр событий безопасности

13. Настройка уведомлений и отправки сообщений

Для задания настроек уведомлений и подключения к системам отправки сообщений используется раздел «Сообщения» консоли управления Blitz Identity Provider (см. Рисунок 159). 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.

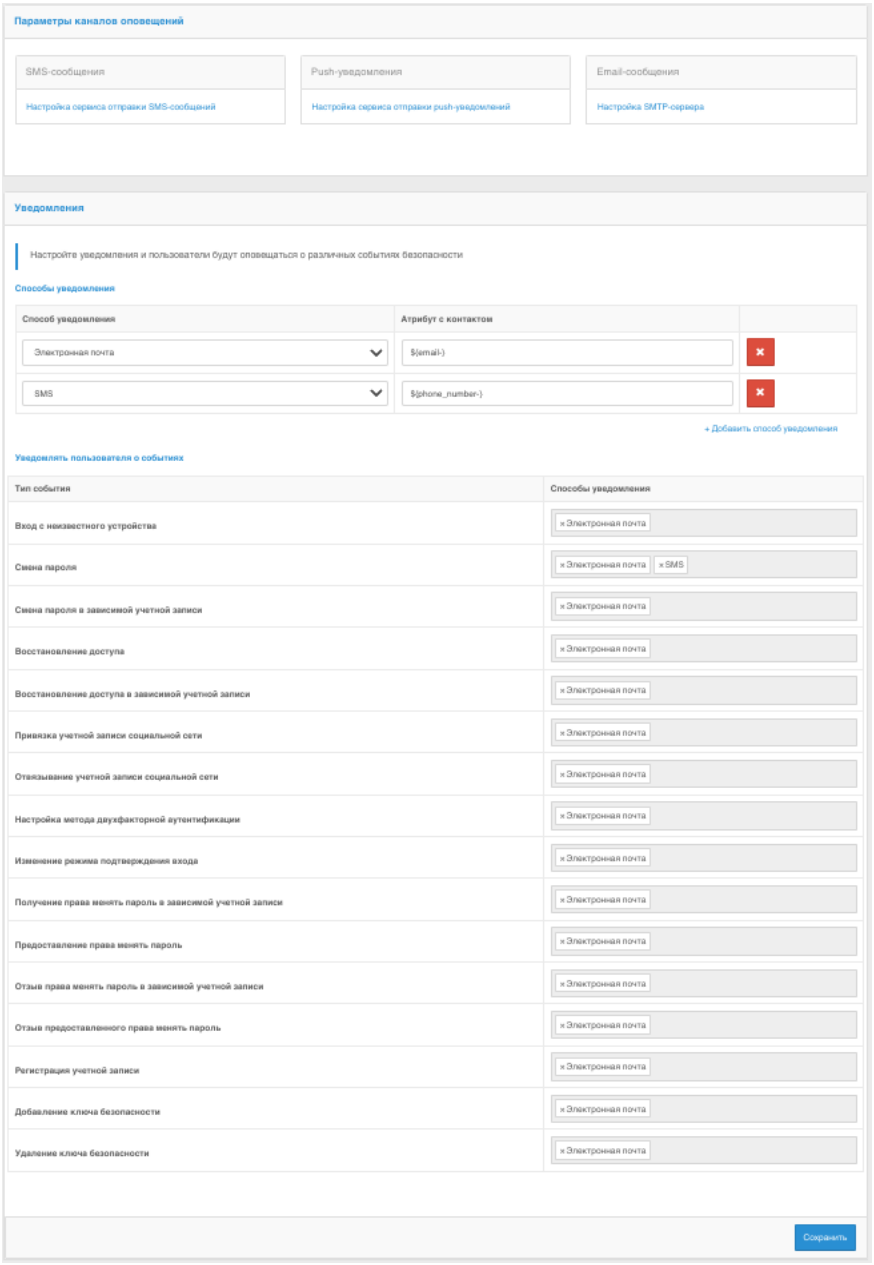

Рисунок 159 – Настройка уведомлений и подключения к системам отправки сообщений

13.1. Настройка подключения 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» (см. Рисунок 160).

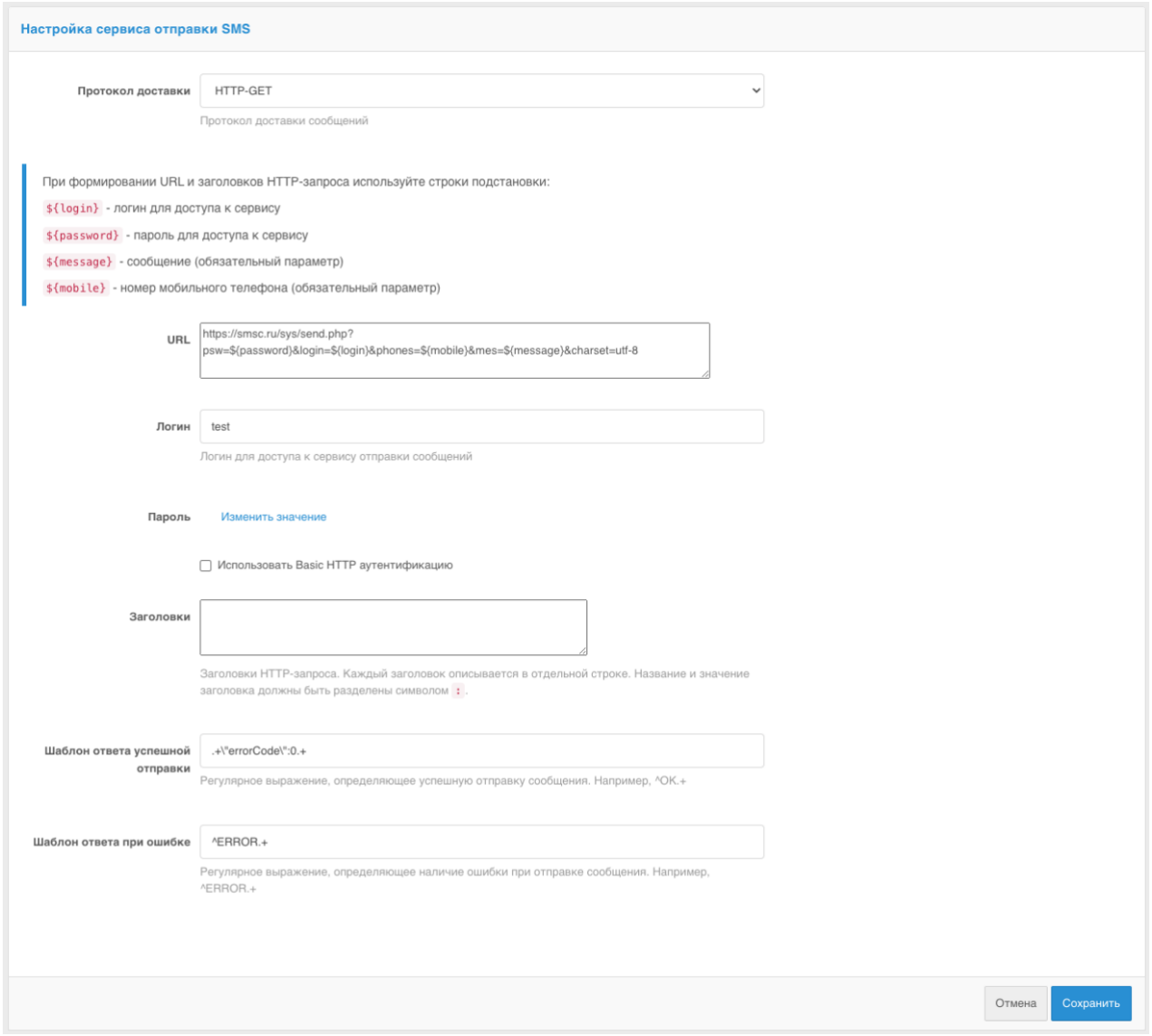

Рисунок 160 – Настройка подключения к SMS-шлюзу

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 авторизации HTTP Basic Authorization);
- 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.

13.2. Настройка подключения 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 авторизации HTTP Basic Authorization);
- 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 ниже).

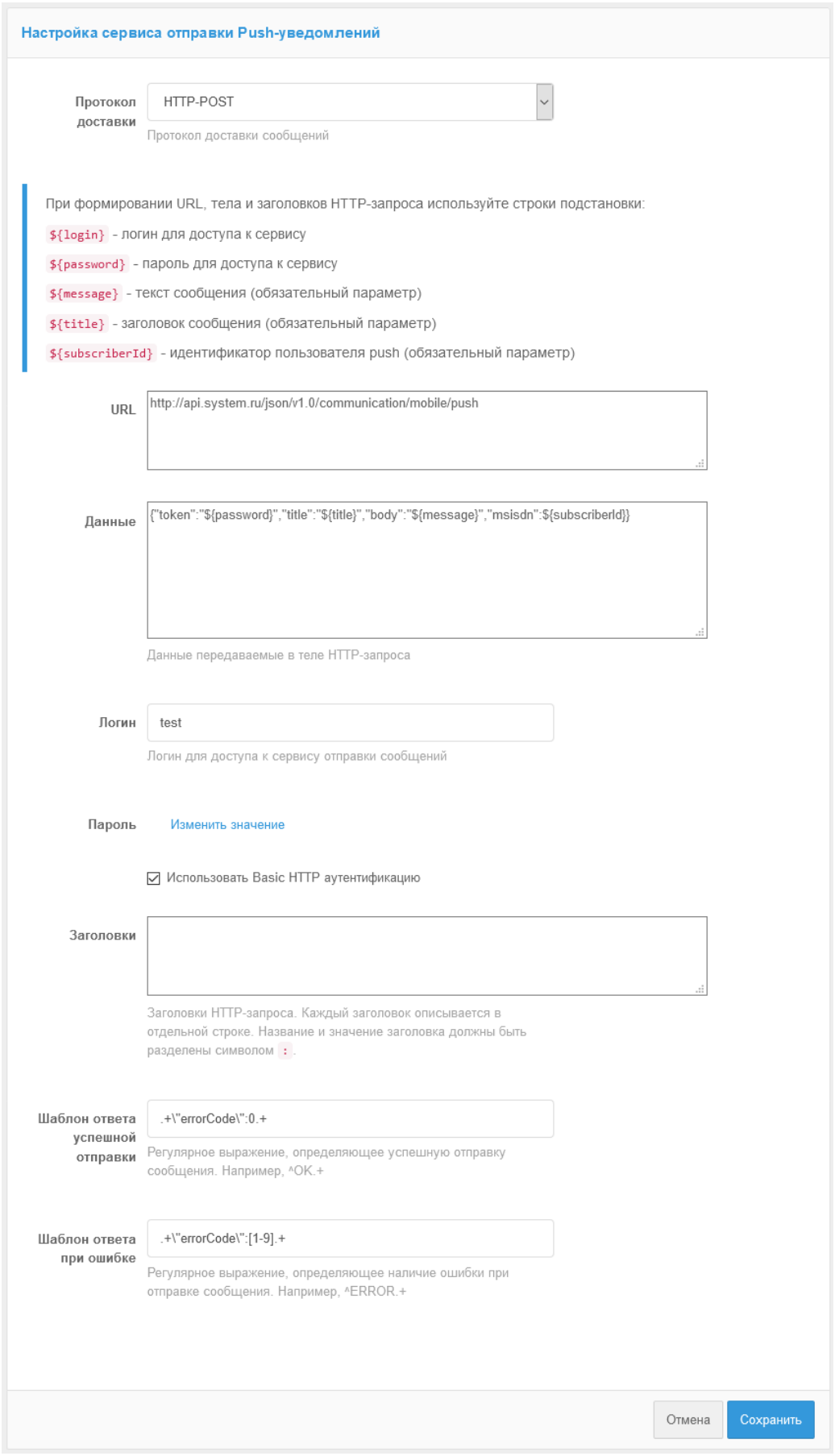

Рисунок 161 – Настройка интеграции с сервисом отправки push-уведомлений

13.3. Настройка подключения к SMTP-шлюзу

В Blitz Identity Provider необходимо настроить возможность отправлять по 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» (см. Рисунок 162).

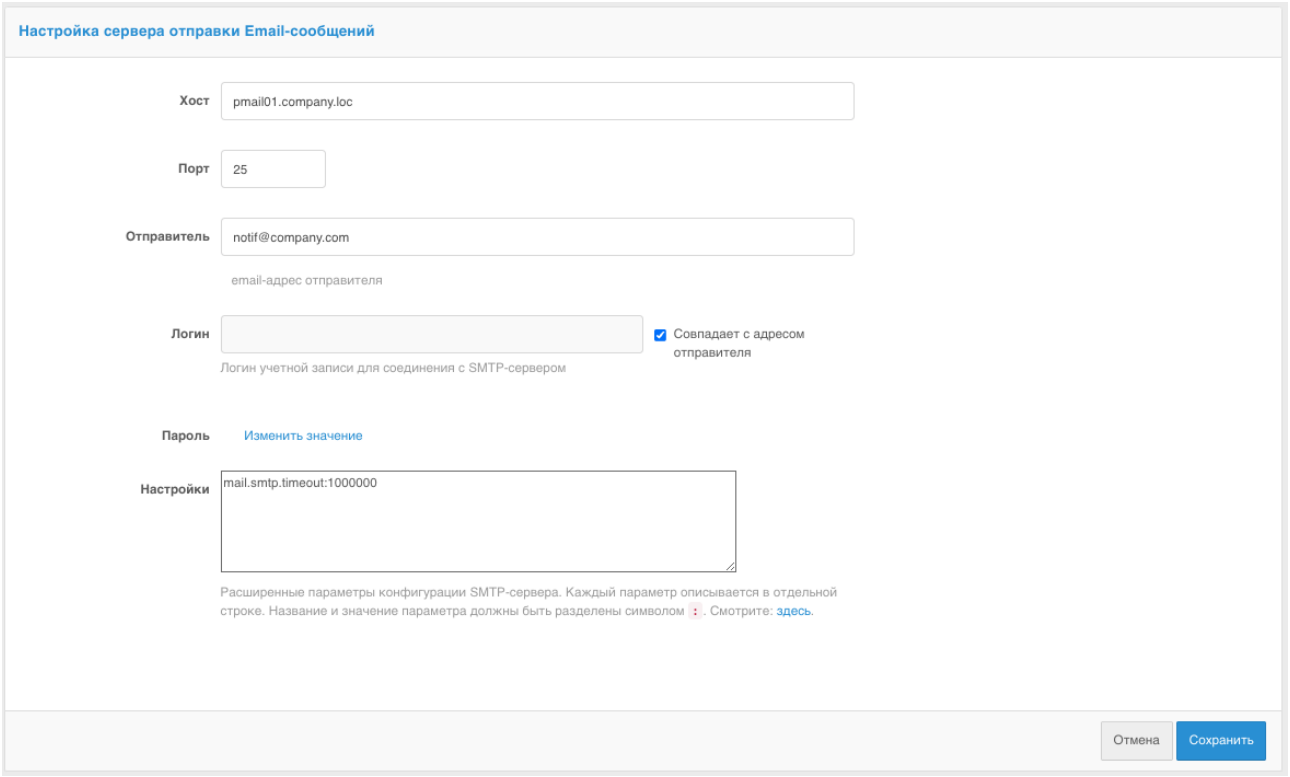

Рисунок 162 – Настройка подключения к SMTP-шлюзу

Необходимо задать следующие настройки:

- имя хоста SMTP-шлюза;
- порт хоста SMTP-шлюза;
- необходимо или нет использовать TLS для защищенного подключения к шлюзу;
- email отправителя сообщений;
- логин учетной записи на SMTP-шлюзе, от имени которой Blitz Identity Provider будет производить отправку email (если логин совпадает с email отправителя, то следует отметить соответствующий чекбокс);

- пароль от учетной записи на SMTP-шлюзе, от имени которой Blitz Identity Provider будет производить отправку email;
- настройки – дополнительные параметры конфигурации взаимодействия с SMTP-шлюзом[60].

[60] См.: https://javaee.github.io/javamail/docs/api/com/sun/mail/smtp/package-summary.html

14. Настройка внешнего вида страницы входа

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 тэгов;
- перечнем приложений;
- описанием.

По умолчанию создан шаблон с идентификатором default – он используется для всех приложений, подключенных к Blitz Identity Provider, а также для страниц единого логаута.

Редактирование шаблона по умолчанию осуществляется с помощью специального конструктора (см. п. 14.1).

Также имеется возможность:

- создавать и изменять новые шаблоны с помощью конструктора и назначать их разным приложениям (п. 14.2);
- создавать и изменять новые шаблоны в ручном режиме (п. 14.3).

14.1. 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 (Рисунок 163 – Рисунок 167).

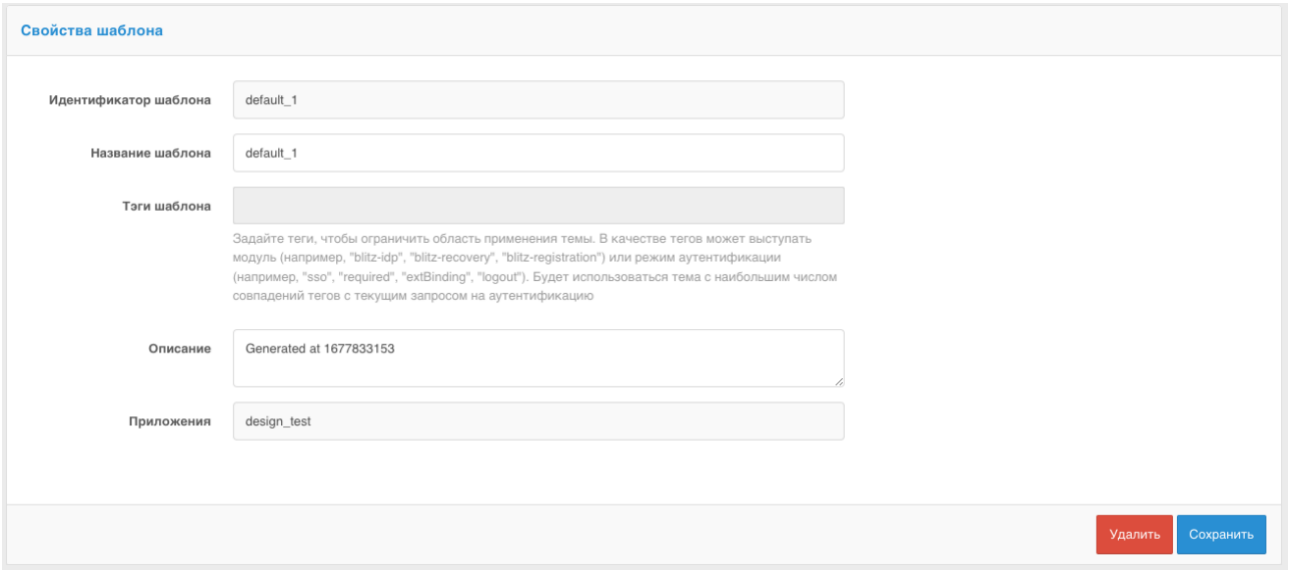

Рисунок 163 – Настройка внешнего вида страницы входа (свойства шаблона)

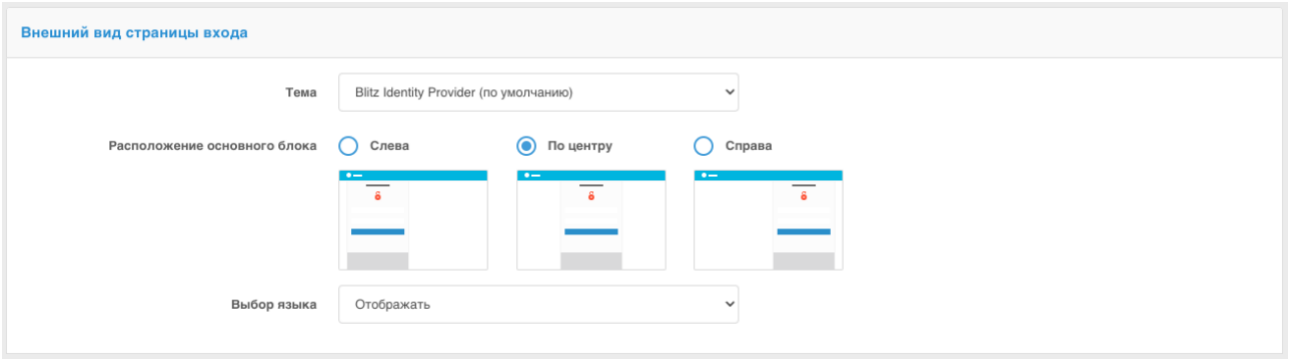

Рисунок 164 – Настройка внешнего вида страницы входа (внешний вид страницы входа)

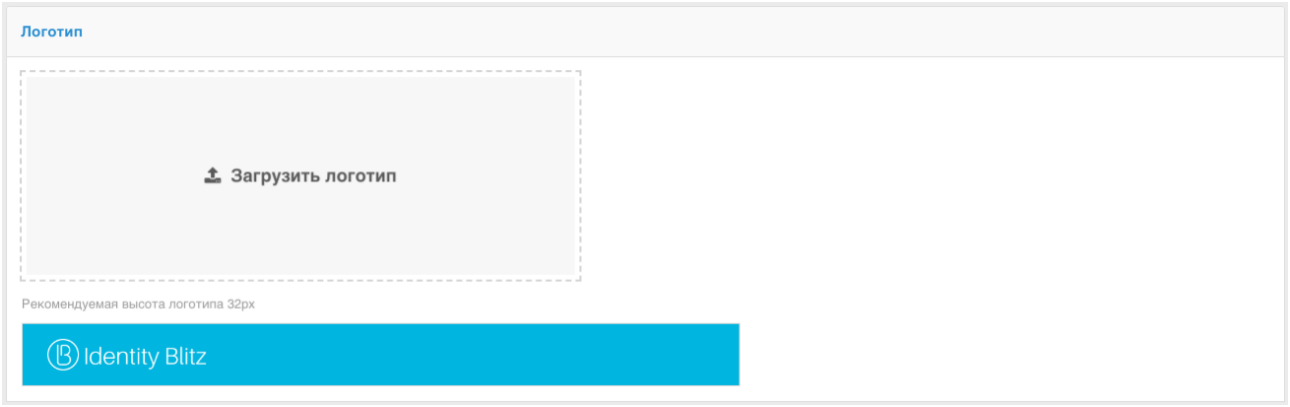

Рисунок 165 – Настройка внешнего вида страницы входа (логотип)

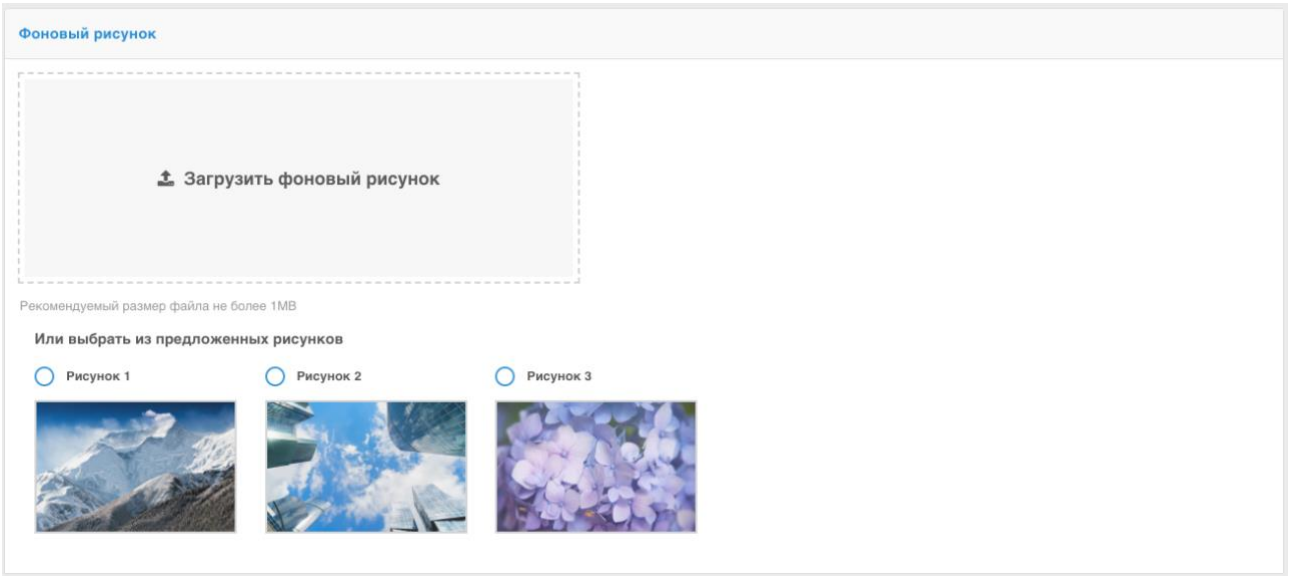

Рисунок 166 – Настройка внешнего вида страницы входа (фоновый рисунок)

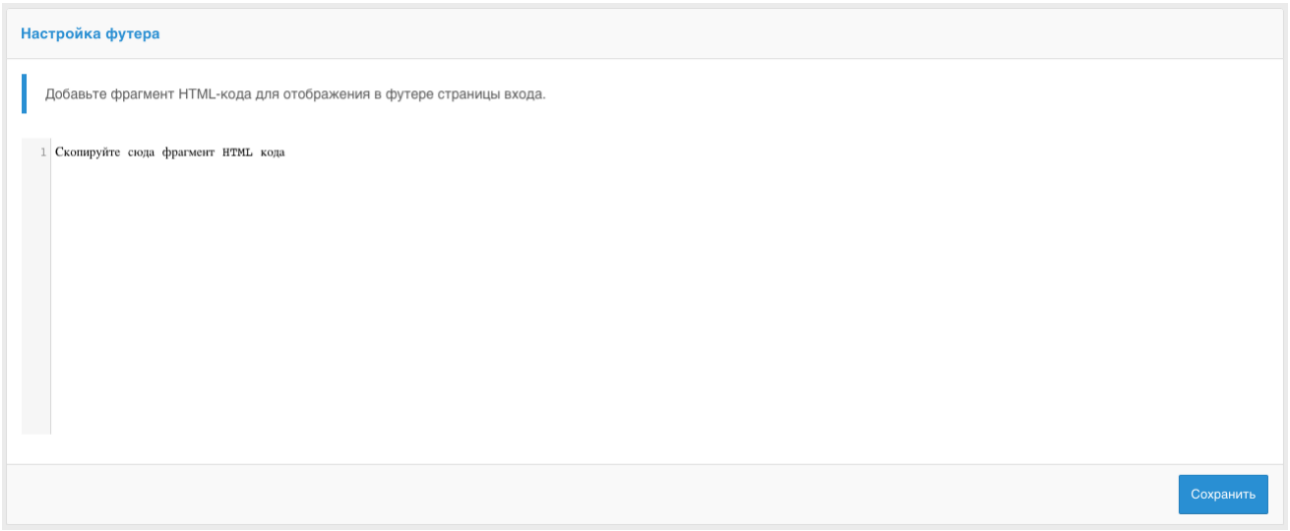

Рисунок 167 – Настройка внешнего вида страницы входа (настройка футера)

В стандартной поставке конструктор Blitz Identity Provider предоставляет следующие возможности:

- три цветовых темы оформления элементов интерфейса;
- возможность определить местоположения блока ввода сведений (идентификации и аутентификации, регистрации, восстановления пароля);
- возможность загрузки логотипа компании для отображения в заголовке страницы;
- выбор фонового рисунка (можно выбрать из 3 стандартных рисунков в каждой теме оформления, либо загрузить свой собственный фоновый рисунок);
- настройка содержания футера страницы входа.

На рисунках 168 и 169 приведены примеры страниц входа в результате стандартной настройки.

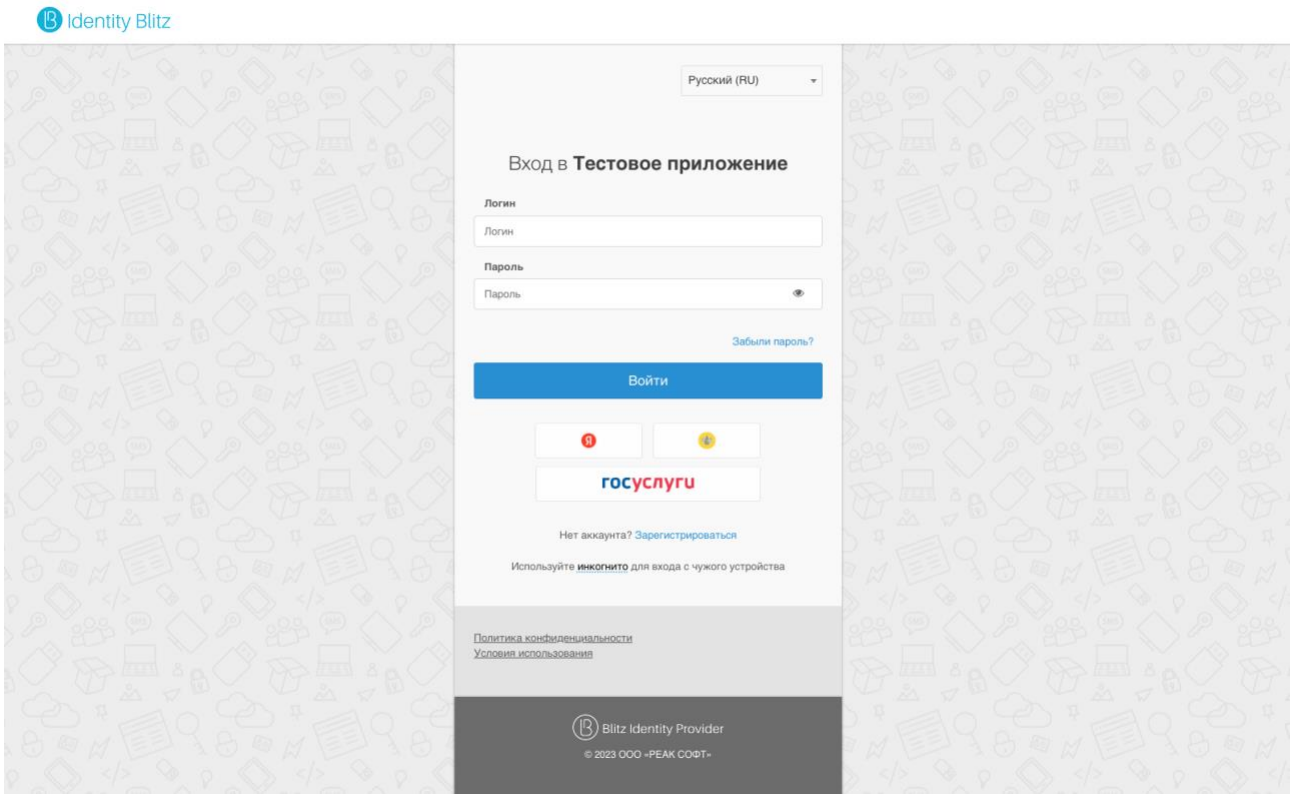

Рисунок 168 – Пример страницы входа с social login, дополнительным футером и выбором языка

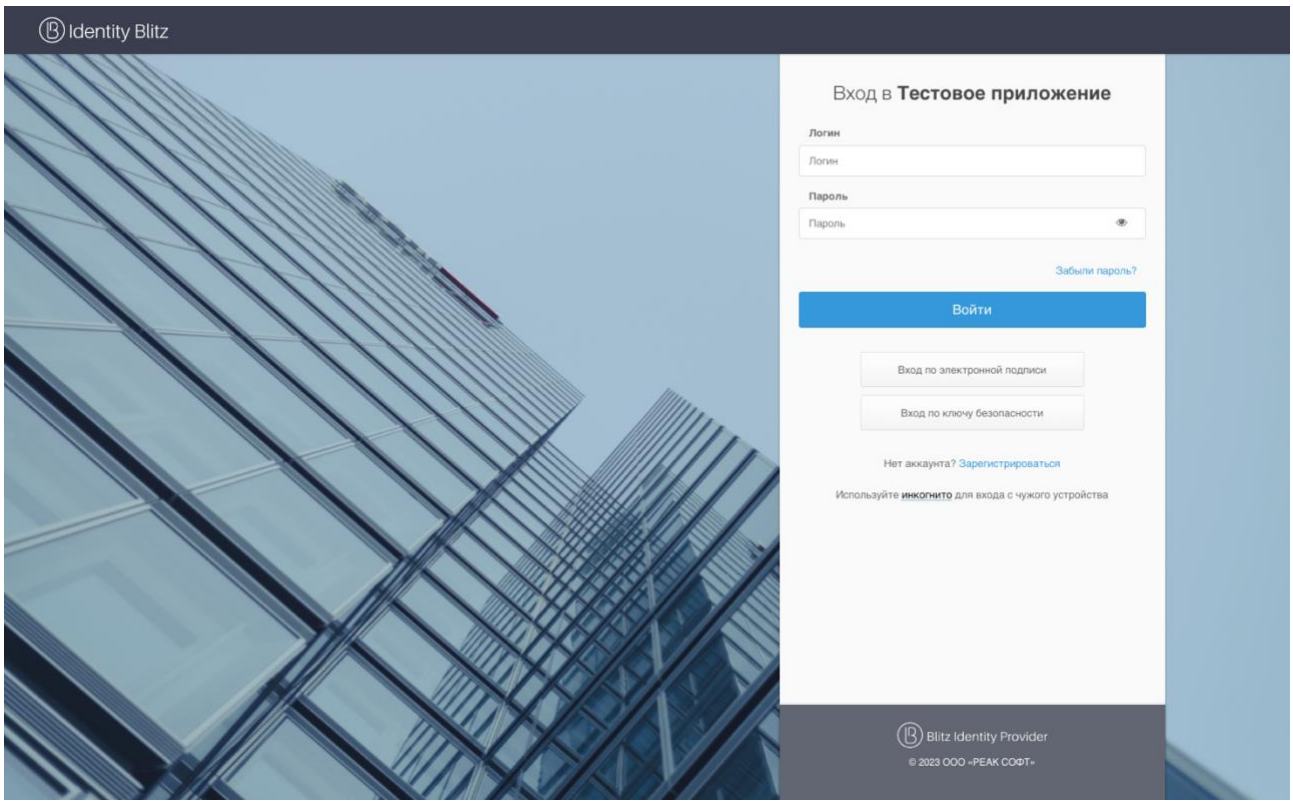

Рисунок 169 – Пример страницы входа в темном интерфейсе и с режимами входа по электронной подписи и ключам безопасности

14.2. 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.

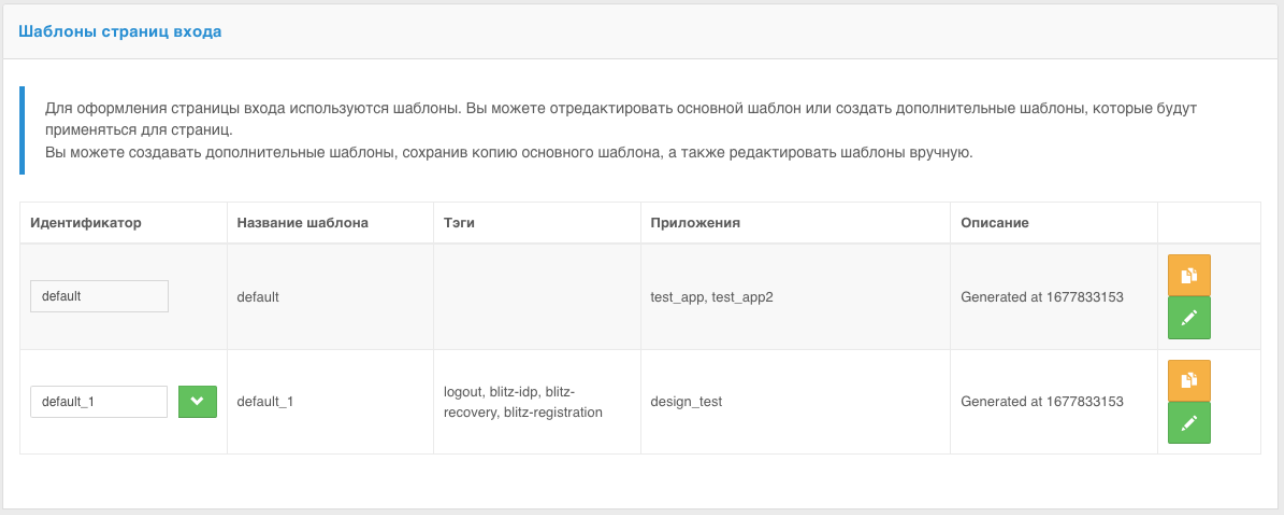

Рисунок 170 – Создание нового шаблона на базе существующего основного шаблона

Чтобы новый шаблон использовался при входе в некоторое приложение, необходимо в разделе «Приложения» перейти к редактированию нужного приложения и выбрать требуемый шаблон страниц.

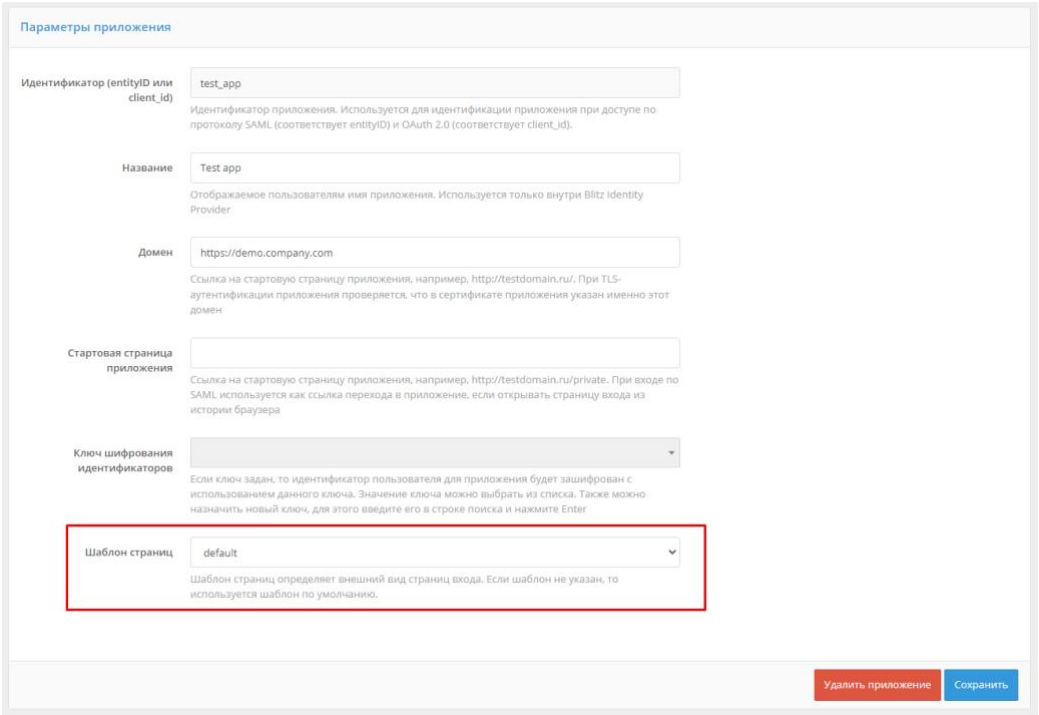

Рисунок 171 – Назначение шаблона страницы входа приложению

14.3. 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  ;
- перейти в соответствующую директорию с шаблонами;
- распаковать архив с только что созданным шаблоном;
- отредактировать файл meta.conf, содержащийся в архиве, удалив параметр builder (см. Рисунок 172);
- обратно заархивировать файлы шаблона, убедившись, что файл meta.conf находится в корневой директории.

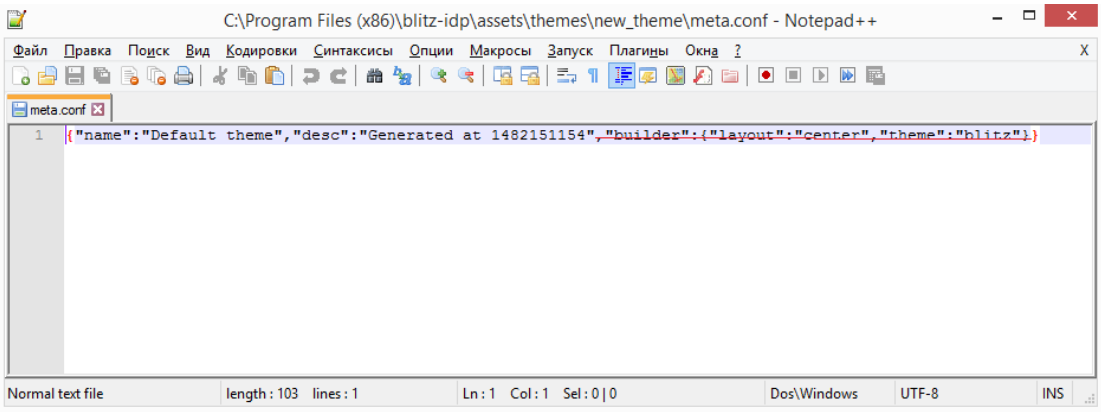

Рисунок 172 – Удаление параметра builder из файла meta.conf

После выполнения этих шагов появится возможность редактирования темы в ручном режиме. Помимо стандартных полей, описывающих саму тему, доступен блок «Шаблон страниц». Он позволяет создать или изменить шаблон – текстовый файл, который компилируется с помощью шаблонизатора Twirl[61].

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 надо расположить в теге head;

[61] См.: https://www.playframework.com/documentation/2.5.x/ScalaTemplates

- form – HTML-код основной формы, который необходимо расположить в теге body;
- scripts – HTML-код с JavaScript, который необходимо расположить в теге body;
- pathAssets – контекстный путь к ресурсам шаблона.

Функция @fBuilder() добавляет на страницу код основной формы аутентификации (пример основной формы приведен на рисунках 168 и 169). Форма аутентификация (перечень и состав полей, расположение кнопок) не настраивается за исключением изменений, реализуемых средствами CSS. Иными словами, через 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>

<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 173.

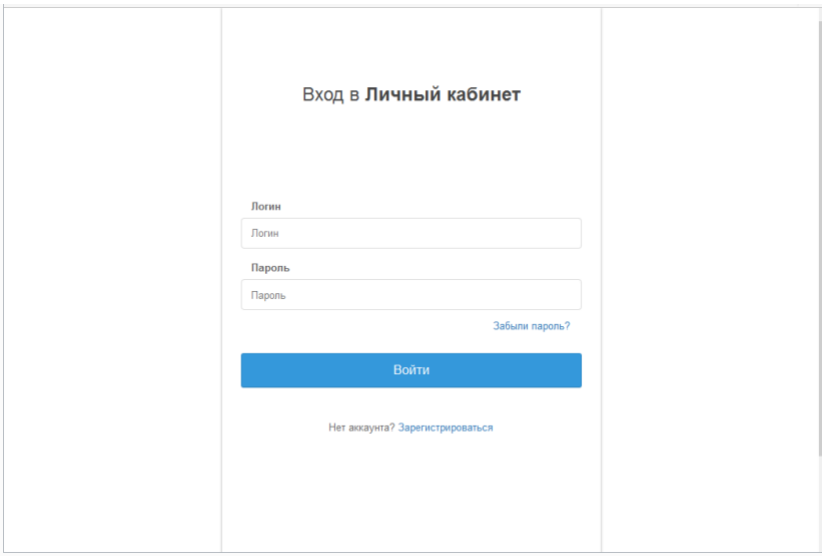

Рисунок 173 – Внешний вид простейшей страницы входа

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.

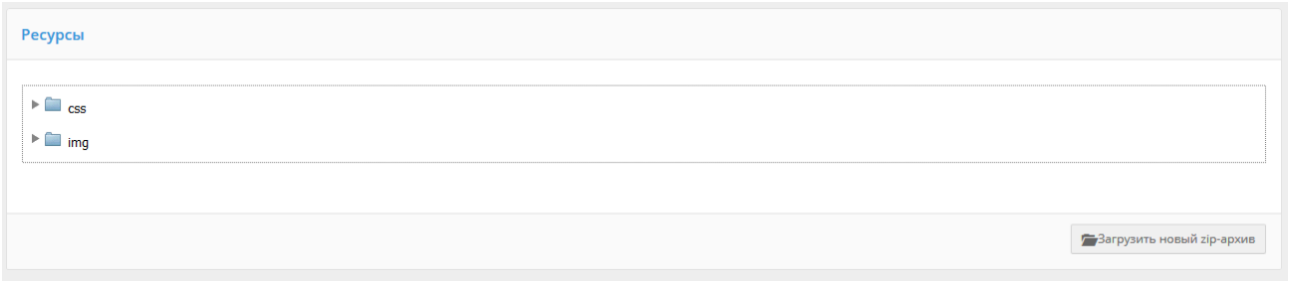

Рисунок 174 – Внешний вид: доступные ресурсы

15. Настройки шлюза безопасности

С помощью Blitz Identity Provider можно осуществлять контроль доступа при вызове приложениями защищаемых сервисов.

Обеспечение авторизации при вызове приложениями сервисов основано на спецификациях OAuth 2.0. Перед использованием сервисов приложение должно получить у Blitz Identity Provider маркер доступа (access_token). Для получения маркера доступа приложению доступны различные способы взаимодействия (см. «Руководство по интеграции»). При этом маркер доступа может быть получен:

- в контексте входа пользователя – маркер будет включать информацию о пользователе и наборе согласий (разрешений), предоставленных пользователем приложению;
- на приложение вне контекста входа пользователя – маркер будет включать набор согласий (разрешений) из числа разрешенных приложению.

Далее с использованием полученного маркера доступа приложение может вызывать сервисы. При этом будут следующие сложности:

- внутри каждого сервиса необходимо будет реализовывать собственную логику авторизации – проверять предоставленный маркер доступа, извлекать из него информацию о пользователе и предоставленных согласиях(разрешениях) и анализировать, достаточно ли их для выполнения сервиса или нет. Осуществлять протоколирование принятого решения по доступу.
- приложение будет использовать единый маркер доступа для вызова различных сервисов. Маркер доступа в таком случае может содержать больше информации о пользователе и больший набор согласий (разрешений), чем нужно конкретному вызванному сервису. Это будет нарушать принцип наименьших привилегий – сервис получит больше прав доступа, чем ему необходимо для выполнения своей задачи.

Чтобы решить вышеописанные сложности в Blitz Identity Provider предусмотрено специальное приложение – шлюз безопасности (blitz-keeper). Это приложение представляет собой специализированный прокси-сервер, используемый при вызове защищаемых сервисов – приложение вызывает сервисы не напрямую, а через шлюз безопасности. При этом шлюз безопасности берет на себя выполнение следующих задач:

- Проверяет включенный в вызов сервиса заголовок авторизации, извлекает из заголовка маркер доступа и, во взаимодействии с сервисом авторизации (blitz-idp) выполняет проверку, действителен ли маркер доступа, а также, достаточно ли у пользователя и приложения прав для вызова защищаемого сервиса.
- Во взаимодействии с сервисом авторизации (blitz-idp) заменяет маркер доступа таким

образом, чтобы передаваемый от шлюза безопасности к защищаемому сервису маркер безопасности содержал только тот набор сведений о пользователе и разрешений, который необходим для работы защищаемого сервиса. При этом из маркера безопасности могут быть как изъяты излишние разрешения и сведения о пользователе, так и наоборот, добавлены в маркер доступа дополнительные разрешения и сведения, если такое установлено политикой безопасности.

- Протоколирует в журнале событий безопасности Blitz Identity Provider события успешной и неуспешной проверки прав доступа.

Взаимодействие шлюза безопасности с сервисом авторизации осуществляется на основе спецификации OAuth 2.0 Token Exchange[62]. Иллюстрация взаимодействия приведена на схеме (см. Рисунок 175).

Рисунок 175 – Схема взаимодействия при вызове приложением защищаемого сервиса

Настройка использования шлюза безопасности для защиты сервисов заключается в выполнении следующих шагов и описана в последующих подразделах:

- Настройка blitz-keeper.
- Создание правил доступа к сервисам.
- Регистрация правил обмена маркеров доступа в blitz.conf.

15.1. Настройка blitz-keeper

Настройка blitz-keeper осуществляется путем редактирования конфигурационного файла blitz-keeper.conf, расположенного в каталоге /etc/blitz-keeper.

[62] Спецификация, описывающая способы обмена маркеров в целях делегирования вызовов между сервисами и для задач имперсонификации пользователей. См.: https://tools.ietf.org/html/rfc8693.

Пример конфигурационного файла:

```
{
  "authenticators": {
    "prod-auth": {
      "type": "token-exchange",
      "te": "https://blitz-host/blitz/oauth/te",
    },
  },
  "services" : {
    "api-1":{
      "display-name" : "secured services",
      "host": "service-host.com",
      "locations": {
        "/api/service1/**": {
          "methods" : ["GET","POST"],
          "authenticator": "prod-auth",
          "required-scopes": ["scope1","scope2"]
        },
        "/path/api/user/*/getdata/**": {
          "methods" : ["GET","PUT"],
          "authenticator": "prod-auth",
          "required-scopes": ["scope3"]
        }
      }
    }
  }
}
```

В блоке authenticators нужно зарегистрировать все используемые сервисы авторизации blitz-idp. Обычно достаточно использовать один единственный сервис авторизации для защиты сервисов, и тогда нужно заполнить только один блок как в примере (в примере зарегистрирован один сервис авторизации с именем prod-auth). Если в системе используется несколько раздельных установок Blitz Identity Provider (например, ПРОД- и ТЕСТ-среда или внутренний контур для сотрудников и внешний контур для клиентов), то можно использовать общий шлюз безопасности, который будет взаимодействовать с несколькими разными сервисами авторизации – тогда нужно в блоке authenticators задать настройки нескольких сервисов авторизации. Для каждого сервиса авторизации задается имя (в примере использован prod-auth, но можно задать любое имя). В блоке настроек сервиса авторизации задается тип взаимодействия (type) в значении token-exchange (пока это единственный поддерживаемый тип взаимодействия) и адрес (te) вызова обработчика Token Endpoint сервиса авторизации. Если blitz-keeper развернут на отдельных серверах, то рекомендуется задать адрес обработчика с https и доменным именем. Если приложение blitz-keeper развернуто на том же сервере что сервис авторизации blitz-idp, то рекомендуется задать в te локальное имя, например, http://localhost:9000/blitz/oauth/te.

В блоке services нужно зарегистрировать защищаемые сервисы. Для всех защищаемых сервисов можно создать общий блок настроек или несколько отдельных блоков. Каждый блок имеет имя (в примере, api-1). Внутри блока задаются настройки:

- display-name – текстовое описание сервиса (любой комментарий или описание);
- host – адрес сервера защищаемого сервиса;
- locations – допустимые пути и операции вызова сервиса.

В блоке locations указываются настройки всех путей сервиса и разрешенных методов. В качестве имени каждого вложенного блока указывается адрес сервиса. Допустимо в адресе использовать звезду (*), чтобы указать на пропуск отдельного компонента в адресе пути сервиса и допустимо использовать двойную звезду (**), чтобы указать, что вся оставшаяся часть пути сервиса может быть любая. Внутри вложенного блока с адресом сервиса можно опционально перечислить разрешенные методы сервиса (настройка methods), указать имя используемого сервиса авторизации (настройка authenticator) и перечень разрешений (настройка required-scopes) для целевого маркера доступа, которые будут включены в маркер доступа, передаваемый в защищаемый сервис.

После изменения настроек в blitz-keeper.conf необходимо перезапустить шлюз безопасности.

15.2. Создание правил доступа к сервисам

Правила доступа к сервисам создаются в директории /usr/share/identityblitz/blitz-config/token_exchange/rules/. Каждое правило создается как отдельный текстовый файл без расширения.

Пример файла с правилом доступа (тип specialize):

```
{
    "name": "rule-name",
    "type": "specialize",
    "desc": "",
    "subjectTokenCond": {
        "clientRights": [],
        "userRights": [],
        "scopes": ["openid"],
        "userClaims": {},
        "userGroups": []
    },
    "issue": {
        "ttlInSec": 3600,
        "allowedScopes": ["openid","profile"],
        "allowedClaims": ["sub","global_role","org_id","rights"],
        "addingScopes": [],
        "addingClaims": []
    }
}
```

Пример файла с правилом доступа (тип impersonate):

```
{
    "name": "rule-name",
    "type": "impersonate",
    "desc": "",
    "subjectTokenCond": {
        "clientRights": [],
        "userRights": [],
        "scopes": ["openid"],
        "userClaims": {},
        "userGroups": []
    },
    "authClientCond": {
        "requiredRights":[
            {
                "rights": ["right1"],
                "target": {
                    "type": "its",
```

```
                    "name": "app1"
                }
            }
    },
    "issue": {
        "ttlInSec": 3600,
        "allowedScopes": ["openid","profile"],
        "allowedClaims": ["sub","global role","org id","rights"],
        "addingScopes": [],
        "addingClaims": []
    }
}
```

Нужно заполнить следующие атрибуты правила доступа:

- name – имя правила, которое должно совпадать с именем файла с правилом доступа;
- type – тип правила. Поддерживаются следующие типы правил:
  - specialize – по такому правилу приложение запрашивает обмен маркера доступа, выданного этому же приложению. Обмен выполняется с целью специализации маркера доступа – замены в нем разрешений (scope), атрибутов (claims) и списка получателей (audience, aud), формат маркера (jwt или opaque);
  - impersonate – по такому правилу приложение запрашивает обмен маркера доступа, выданного другому приложению. Обмен выполняется при условии, что в предоставленном на обмен маркере доступа запрашивающее обмен приложение присутствует в списке получателей (audience, aud). Обмен используется в сценариях, когда приложение А получило исходно маркер доступа, подготовило его для передачи приложению Б (через обмен по типу правила specialize), передало приложению Б, так, что приложение Б выпустило на основе полученного маркера доступа свой собственный (через обмен по типу правила impersonate).
- desc – описание правила. Можно ввести любую текстовую информацию;
- subjectTokenCond – условия выполнения правила. Если все указанные в правиле условия будут выполняться, то правило считается выполненным. Если хотя бы одно из условий в правиле не будет выполнено, то все правило считается невыполненным. Условия выполнения правил могут быть следующие:
  - clientRights – проверка наличия у приложения указанных прав доступа;

    Пример правила:

```
"clientRights": [
  {
    "rights": ["right1"],
    "target": {
      "type": "its",
      "name": "app1"
    }
  }
]
```

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. п. 16.1.16).

  Пример 2 правила:

```
"userRights": [
  {
    "rights": ["security_administrator"],
    "target": {
      "type": "grps",
      "name": "${org id}",
      "ext": "orgs"
    }
  }
]
```

  В указанном правиле проверяется наличие у пользователя права доступа security_administrator в отношении группы пользователей из профиля orgs, имеющей идентификатор, совпадающий со значением атрибута org_id из состава исходного маркера доступа. В отличии от примера 1 в данном примере иллюстрируется возможность в качестве имени объекта права доступа указывать не конкретное значение объекта, а ссылаться на объект на основе значений из присланного маркера доступа ($org_id).

  Пример 3 правила:

```
"userRights": [
  {
    "rights": ["right3"],
    "target": {
      "type": "its",
      "name": "app1"
    }
  }
]
```

  В данном примере проверяется наличие у пользователя права доступа right3 в отношении приложения app1.

- scopes – проверка присутствия в маркере доступа требуемых разрешений (см. п. 5.3.2);

  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.

15.3. Настройка правил обмена маркеров доступа

Чтобы определить, для каких защищаемых сервисов какие должны применяться правила доступа, 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 (см. п. 15.2). 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 звезду (*) для пропуска одного компонента пути адреса и двойную звезду (**) 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.

16. Настройки конфигурационных файлов

Конфигурационные файлы всех приложений Blitz Identity Provider кроме приложения blitz-keeper расположены в каталоге /usr/share/identityblitz/blitz-config.

Используются следующие директории и конфигурационные файлы:

- apps/ – настройки подключенных приложений (см. п. 16.1.30);
- assets/ – настройки пользовательского интерфейса (см. п. 4.5.2, п. 7, п. 14);
- custom_messages/ – строки пользовательского интерфейса (см. п. 16.2);
- devices/ – вспомогательные каталоги для обработки загрузки HOTP и TOTP устройств (см. п. 4.17.1);
- dynamic/idstore/– пользовательские процедуры для кастомизации логики операций с хранилищами данных (см. п. 6.3);
- flows/ – процедуры входа (см. п. 6.1);
- saml/ – настройки SAML (см. п. 5.2);
- simple/ – настройки подключения приложений по протоколу Simple (см. п. 5.1);
- token_exchange/rules/ – настройки правил обмена маркеров доступа (см. п. 15);
- blitz.conf – основной файл конфигурации (см. п. 16.1);
- boot.conf – настройки путей к конфигурационным файлам;
- console.conf – настройки консоли управления (см. п. 16.3);
- credentials – учетные записи администраторов консоли управления (см. п. 16.3.3);
- play.conf – настройки серверов приложений (см. п. 2.1.5 и п. 16.1.12);
- 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 (см. п. 15);
- blitz-keeper-log4j.xml – настройки журналирования событий и ошибок.

16.1. Файл настроек 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 логаута;
- blitz.prod.local.idp.messages – настройки файлов сообщений;
- blitz.prod.local.idp.messaging – настройки вызова сервисов информирования;
- blitz.prod.local.idp.net – настройки сети;
- blitz.prod.local.idp.notifier – настройки уведомлений о событиях;
- blitz.prod.local.idp.oauth – настройки разрешений (scope);
- blitz.prod.local.idp.password-policy – настройки парольной политики;
- blitz.prod.local.idp.play – настройки сервера приложений Blitz Identity Provider;
- blitz.prod.local.idp.provisioning – настройки сервисов регистрации пользователей и восстановления забытого пароля;
- blitz.prod.local.idp.realms – настройки шифрования идентификаторов приложений («домены приватности);
- blitz.prod.local.idp.rights – настройки прав доступа;
- blitz.prod.local.idp.saml – настройки SAML;
- blitz.prod.local.idp.stores – настройки основной СУБД;
- blitz.prod.local.idp.tasks – настройки механизма обработки задач;
- blitz.prod.local.idp.user-profile – настройки личного кабинета;
- blitz.prod.local.idp.webAuthn – настройки ключей безопасности;
- home – путь к каталогу установки Blitz Identity Provider на сервере приложений.

Далее приведено описание настроек, недоступных из консоли управления, и проводимых посредством редактирования конфигурационного файла blitz.conf.

16.1.1. Ограничение количества одновременных проверок пароля пользователя

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
}
```

16.1.2. Отключение функции смены 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 не желательно. 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.

16.1.3. Настройка внешнего валидатора атрибута

Если возможностей, предоставляемых правилами преобразования входных значений с помощью регулярных выражений (см. п. 3.1.3) 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"
                },
                "unique" : false
            },
            …
        }
    ]
}
```

16.1.4. Настройка транслятора 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

При необходимости настроить транслятор для атрибута objectGUID из LDAP-каталога Active Directory, чтобы этот атрибут представлялся не в байтовом виде, а в форме строки GUID, можно использовать встроенный в Blitz Identity Provider 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 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"
```

16.1.5. Настройка вызова внешнего сервиса проверки электронной подписи

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 >"
}
```

16.1.6. Настройка вызова плагина электронной подписи

Для задания нестандартных настроек вызова плагина электронной подписи при запросе входа пользователем по электронной подписи необходимо в конфигурационном файле в разделе blitz.prod.local.idp.login.methods.x509 необходимо создать блок настроек plugin с переопределенными настройками вызова плагина:

```
"plugin" : {
    "allModulesEnabled" : false,
    "capi" : {
        "providers" : [
            {
                "name" : "Crypto-Pro GOST R 34.10-2001 Cryptographic Service Provider",
                "pinMode" : 1
            },
            {
                "name" : "Crypto-Pro GOST R 34.10-2012 Strong Cryptographic Service Provider",
                "pinMode" : 1
            },
            {
                "name" : "Infotecs Cryptographic Service Provider",
                "pinMode" : 1
            },
            {
                "name" : "Signal-COM GOST R 34.10-2012 (512) Cryptographic Provider",
                "pinMode" : 1
            },
            {
                "name" : "Signal-COM CPGOST Cryptographic Provider",
                "pinMode" : 1
            },
            {
                "name" : "Signal-COM GOST R 34.10-2012 (256) Cryptographic Provider",
```

```
                "pinMode" : 1
            }
        ],
        "stores" : []
    },
    "modules" : [
        "capi",
        "Aladdin R.D. Unified JaCarta",
        "ISBC ESMART",
        "Rutoken",
        "SafeNet"
    ]
}
```

В блоке конфигурации можно убрать лишние модули из modules и providers, чтобы ограничить перечень доступных средств электронной подписи. Также для используемых провайдеров можно настроить режим ввода пин-кода согласно документации на плагин[63].

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": []
}
```

16.1.7. Настройка CAPTCHA

Для отображения сервиса CAPTCHA при входе по логину и паролю необходимо внести изменения в конфигурационный файл, а также загрузить необходимые файлы (CSS и JS).

Изменения конфигурационного файла должны быть произведены

- в блоке настроек blitz.prod.local.idp.captcha. Пример записи настройки приведен ниже:

```
"captcha" : {
    "exampleCaptcha": {
        "operations": [
            {
                "call": {
                    "headers": [
                        "accept:application/json",
                        "Authorization:Bearer ${cfg.bearerToken}"
                    ],
                    "method": "post",
                    "url":
"https://captcha.example.com/captcha/1.0.0/check?uniqueFormHash=${ste.uniqueFormHash}&code=${ocp.cod
e}&options[system]=${cfg.system}&options[token]=${cfg.token}"
                },
                "check": {
                    "errRegExp": {},
                    "okRegExp": {
                        "error": "0"
                    }
                },
                "name": "check",
                "newState": {
                    "uniqueFormHash": "${rsp.result.uniqueFormHash-}"
                }
            },
            {
                "call": {
                    "headers": [
```

[63] См.: https://identityblitz.ru/products/smart-card-plugin/documentation/?ref=p_pl#document-9

```
                        "accept:application/json",
                        "Authorization:Bearer ${cfg.bearerToken}"
                    ],
                    "method": "get",
                    "url":
"https://captcha.example.com/captcha/1.0.0/create?type=${cfg.type}&options[system]=${cfg.system}&opt
ions[token]=${cfg.token}"
                },
                "name": "create",
                "newState": {
                    "uniqueFormHash": "${rsp.result.uniqueFormHash-}"
                }
            },
            {
                "call": {
                    "headers": [
                        "accept:application/json",
                        "Authorization:Bearer ${cfg.bearerToken}"
                    ],
                    "method": "post",
                    "url":
"https://captcha.example.com/captcha/1.0.0/refresh?uniqueFormHash=${ste.uniqueFormHash}&type=${cfg.t
ype}&options[system]=${cfg.system}&options[token]=${cfg.token}"
                },
                "name": "refresh"
            }
        ],
        "plainParams": {
            "type": "arithmetic"
        },
        "secureParams": {
            "bearerToken": "<access_token>",
            "system": "<system_id>",
            "token": "<system token>"
        }
    }
}
```

В этом блоке содержатся параметры вызова трех методов сервиса CAPTCHA (create, check, refresh), а также секретные параметры – маркер доступа (bearerToken), идентификатор системы (system), а также токен системы (token).

- в блоке настроек входа по логину и паролю blitz.prod.local.idp.password. Внутри этого блока следует добавить блок captcha и настроить согласно примеру:

```
"captcha" : {
  "enabled": true,
  "initJs": "require(['https://demo.reaxoft.ru/themes/default/assets/js/passwordCaptcha.js',
'captcha-conf'], function(captcha, conf){ captcha(conf,
'https://demo.reaxoft.ru/themes/default/assets/css/passwordCaptcha.css');});",
  "mode": {
    "type": "always_on"
  },
  "name": "exampleCaptcha"
}
```

В этом блоке следует настроить следующие параметры:

- enabled – признак того, включена CAPTCHA или нет (true/false);
- initJs – содержит ссылки на JS-скрипт и CSS-стили, загружаемые на странице входа и необходимые для отображения/вызова CAPTCHA на странице входа;
- mode – режим отображения CAPTCHA, предусмотрены следующие режимы:
  - always_on – CAPTCHA отображается всегда;
  - on_header – CAPTCHA отображается, если в запросе есть заголовок, указанный в параметре name, и значением, указанном в параметре value;
  - by_brute_force_protection – CAPTCHA отображается, если Blitz Identity Provider

обнаружил подбор пароля к конкретной учетной записи или массовый подбор пароля ко всем учетным записям.

При использовании режима by_brute_force_protection требуется дополнительно создать в блоке blitz.prod.local.idp.password блок настроек bruteForceProtection со 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" : {
                "turnOn" : 5
            }
        }
    }
}
```

В случае использования в качестве CAPTCHA сервиса Google reCAPTCHA v3[64] необходимо:

- задать следующие настройки в blitz.prod.local.idp.captcha:

```
"captcha" : {
  "reCAPTCHAv3" : {
    "operations" : [
      {
        "call" : {
          "headers" : [],
          "method" : "post",
          "url" :
"https://www.google.com/recaptcha/api/siteverify?secret=${cfg.secret}&response=${ocp.response}"
        },
        "check" : {
          "errRegExp" : {},
          "okRegExp" : {
            "score" : "1\\.0|0\\.(5|6|7|8|9)",
            "success" : "true"
          }
        },
        "name" : "verify"
      }
    ],
    "plainParams" : {
      "sitekey" : "SITE_KEY"
    },
    "secureParams" : {
      "secret" : "SITE_SECRET"
    }
  }
}
```

Вместо SITE_KEY и SITE_SECRET нужно заполнить значения, полученные при регистрации Google reCAPTCHA v3 на сайте https://g.co/recaptcha/v3. Также нужно скорректировать значение в параметре score – установить требуемый порог успешного прохождения проверки (в примере выставлен порог не ниже 0,5).

[64] См.: https://developers.google.com/recaptcha/docs/v3

- 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);});",
  "name" : "reCAPTCHAv3"
}
```

Для добавления CAPTCHA на страницу регистрации пользователей необходимо задать следующие настройки в blitz.prod.local.idp.provisioning.registration.captcha:

```
"captcha" : {
  "mode" : {
    "_name" : "X-Captcha-Check",
    "_value" : "true",
    "_type" : "on_header",
    "type" : "always_on"
  },
  "enabled" : true,
  "initJs" : "require(['/blitz/assets/blitz-common/javascripts/recaptcha v3.js', 'captcha-conf'],
function(captcha, conf){ captcha(conf);});",
  "name" : "reCAPTCHAv3"
}
```

Для добавления CAPTCHA на страницу восстановления пароля необходимо задать следующие настройки в blitz.prod.local.idp.provisioning.recovery.captcha:

```
"captcha" : {
  "mode" : {
    "_name" : "X-Captcha-Check",
    "_value" : "true",
    "_type" : "on_header",
    "type" : "always_on"
  },
  "enabled" : true,
  "initJs" : "require(['/blitz/assets/blitz-common/javascripts/recaptcha_v3.js', 'captcha-conf'],
function(captcha, conf){ captcha(conf);});",
  "name" : "reCAPTCHAv3"
}
```

16.1.8. Настройка отправки событий в сервер очередей

В сервер очередей могут быть отправлены следующие события:

- регистрация пользователя (USER_REGISTERED);
- смена пароля (USER_PASSWORD_SET);
- смена признака аннулирования сессий (USER_CRID_CHANGED);
- изменения атрибутов пользователя (USER_ATTRIBUTE_CHANGED);
- очистка атрибутов пользователя (USER_ATTRIBUTE_REMOVED);
- удаление пользователя (USER_REMOVED);
- привязка внешней учетной записи (FEDERATION_POINT_BOUND);
- отвязка внешней учетной записи (FEDERATION_POINT_UNBOUND);
- отзыв выданного приложению разрешения (scopes) (SCOPES_REVOKED);
- создание группы (GROUP_CREATED);
- изменение атрибутов группы (GROUP_UPDATED);

- удаление группы (GROUP_REMOVED);
- включение пользователя в группу (GROUP_MEMBER_ADDED);
- исключение пользователя из группы (GROUP_MEMBER_REMOVED).

Для отправки событий в очередь следует создать блок blitz.prod.local.idp.events следующего содержания (на примере регистрации пользователя и смены пароля):

```
"events" : {
    "drivers" : {
        "rabbit driver" : {
            "properties" : {},
            "server" : {
                "host" : "<RMQ HOST>",
                "port" : 5672
            },
            "type" : "RMQ",
            "user" : {
                "password" : "<RMQ_PASS>",
                "username" : "<RMQ_USERNAME>"
            }
        }
    },
    "routes" : {
        "USER_PASSWORD_SET" : [
            "password sync"
        ],
        "USER_REGISTERED" : [
            "registration"
        ]
    },
    "targets" : [
        {
            "discardList" : "PSWD_SYNC_DISCARD",
            "driver" : {
                "ext" : {
                    "exchange_name" : "users",
                    "routing_key" : "pwd_sync"
                },
                "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 blitz-keystore.bks),

которым следует шифровать пароли в сообщениях.

16.1.9. Настройка использования сервера очередей в качестве брокера сообщений

В Blitz Identity Provider для обработки асинхронных задач применяется встроенный брокер сообщений, использующий для отслеживания задач базу данных.

При большой интенсивности запросов к Blitz Identity Provider может быть целесообразным использование сервера очередей RabbitMQ в качестве брокера сообщений. Для этого нужно в консоли RabbitMQ (обычно, http://hostname:15672/) выполнить следующие настройки:

- создать queue с именем blitz-tasks (в меню «Queues» консоли);
- создать exchange с именем blitz-tasks-exh (в меню «Exchanges» консоли) и настроить binding на очередь blitz-tasks с routing_key с именем blitz-tasks;
- создать пользователя blitz (в меню «Admin» консоли) и назначить ему права на созданную очередь.

После настройки RabbitMQ необходимо скорректировать настройки в blitz.conf – в блоке blitz.prod.local.idp.tasks установить broker-type в значение rmq и задать настройки подключения к RabbitMQ в блоке broker-rmq:

- в параметре exchange задать имя blitz-tasks-exh;
- в параметре queue в блоке executionRules и в параметре name в блоке queues задать имя blitz-tasks;
- в параметре username в блоке user задать имя пользователя (blitz);
- в параметре password в блоке user задать пароль пользователя в открытом виде – после запуска Blitz Identity Provider пароль будет зашифрован;
- в параметрах host и port блока server указать адрес и порт подключения к RabbitMQ;
- при необходимости скорректировать остальные параметры, определяющие размер пула соединений (poolSize), количества каналов (channelSize), время ожидания отклика от сервера очередей (ackTimeout);
- при необходимости скорректировать настройки брокера обработки задачи, определяющие количество попыток (maxAttempts) повторной обработки задач в случае ошибки, время между попытками (redeliveryDelayInSec), размер обрабатываемой пачки сообщений (dequeueBatchSize), период проверки очереди (dequeuePeriodInSec), количество обработчиков (executorPoolSize):

  Пример конфига приведен ниже:

```
"tasks" : {
    "broker-type" : "rmq",
    "broker-rmq" : {
```

```
        "consumer" : {
            "poolSize" : 2
         },
         "exchange" : "blitz-task-exh",
         "publisher" : {
             "ackTimeout" : 15,
              "channelsSize" : 8,
              "poolSize" : 2
         },
         "server" : {
             "host" : "RMQ_HOST",
             "port" : 5672
         },
         "user" : {
             "password" : "CHANGE ME",
             "username" : "blitz"
         }
    },
    "executionRules" : [
        {
            "maxAttempts" : 2,
            "queue" : "blitz-tasks",
            "redeliveryDelayInSec" : 60
        }
    ],
    "queues" : [
        {
            "dequeueBatchSize" : 10,
            "dequeuePeriodInSec" : 30,
            "executorPoolSize" : 5,
            "name" : "blitz-tasks"
        }
    ]
}
```

16.1.10. Настройка хранения 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"
  },
  …
}
```

16.1.11. Настройка времени хранения объектов в базе данных

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 или blitz.prod.local.idp.internal-store-jdbc 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 записей с динамическими client_id и client_secret ограничение по сроку хранения записей в базе данных (по умолчанию записи хранятся бессрочно) от момента их последнего использования. Для этого в блоке blitz.prod.local.idp.internal-store-cb или blitz.prod.local.idp.internal-store-jdbc нужно добавить настройку ttlMapping с указанием doc_type записи (clt) и времени хранения в секундах.

Пример настройки (время хранения записей с динамическими client_id и client_secret ограничено до 90 суток):

```
"internal-store-cb": {
  …
  "ttlMapping": {
    "clt": 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,
  …
}
```

16.1.12. Настройка домена 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 в blitz.prod.local.idp.lang в блоке portal-lang-cookie значение настройки domain (см. п. 16.2.1):

Пример фрагмента конфигурационного файла:

```
"lang" : {
  …
  "portal-lang-cookie" : {
    "domain" : "identityblitz.com",
    …
  }
}
```

При необходимости можно изменить путь до приложений (по умолчанию приложения доступны с использованием пути /blitz). Отредактировать путь можно в конфигурационном файле play.conf. Нужно изменить параметр context в блоке play.http:

```
"http" : {
  "context": "/blitz",
  …
}
```

Изменить домен и путь Blitz Identity Provider в файлах /blitz-config/saml/conf/relying-party.xml, /blitz-config/saml/metadata/idp-metadata.xml.

Пример изменения настроек в relying-party.xml:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<ns18:RelyingPartyGroup …>
  <ns18:AnonymousRelyingParty
    provider="https://demo.identityblitz.com/blitz/saml"
    defaultSigningCredentialRef="IdPCredential"/>
  <ns18:DefaultRelyingParty
    provider="https://demo.identityblitz.com/blitz/saml"
    defaultSigningCredentialRef="IdPCredential">
    …
  </ns18:DefaultRelyingParty>
  …
</ns18:RelyingPartyGroup>
```

Пример изменения настроек в idp-metadata.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<EntityDescriptor … entityID="https://demo.identityblitz.com/blitz/saml">
  <IDPSSODescriptor …>
    …
    <ArtifactResolutionService
      Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML1/SOAP/ArtifactResolution"
      index="1"/>
    <ArtifactResolutionService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML2/SOAP/ArtifactResolution"
      index="2"/>
    <SingleLogoutService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML2/Redirect/SLO"
      ResponseLocation="https://demo.identityblitz.com/blitz/saml/profile/SAML2/Redirect/SLO"/>
    <SingleLogoutService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Plain-Redirect"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML2/Redirect/Plain/SLO"
      ResponseLocation=
        "https://demo.identityblitz.com/blitz/saml/profile/SAML2/Redirect/Plain/SLO"/>
    <SingleLogoutService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML2/SOAP/SLO" />
    …
```

```
    <SingleSignOnService
      Binding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"
      Location="https://demo.identityblitz.com/blitz/saml/profile/Shibboleth/SSO"/>
    <SingleSignOnService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML2/POST/SSO"/>
    <SingleSignOnService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-SimpleSign"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML2/POST-SimpleSign/SSO"/>
    <SingleSignOnService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML2/Redirect/SSO"/>
    <SingleSignOnService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Plain-Redirect"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML2/Redirect/Plain/SSO"/>
  </IDPSSODescriptor>
  <AttributeAuthorityDescriptor …>
    …
    <AttributeService
      Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML1/SOAP/AttributeQuery"/>
    <AttributeService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
      Location="https://demo.identityblitz.com/blitz/saml/profile/SAML2/SOAP/AttributeQuery"/>
    …
    </AttributeAuthorityDescriptor>
</EntityDescriptor>
```

16.1.13. Расширенные настройки подключения к хранилищам

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 а LDAP-каталоге соединение будет закрыто, а Blitz Identity Provider об этом уведомлен не будет. При попытке использования такого соединения из пула может возникнуть длительный таймаут, прежде чем Blitz Identity Provider расценит соединение как закрытое и исключит его из пула соединений. Чтобы такая ситуация не влияла на пользователей, в Blitz Identity Provider предусмотрен алгоритм периодической проверки действительности открытых LDAP-соединений. С периодом healthCheckInterval (в миллисекундах) выполняется проверка состояния соединения, а время таймаута при отсутствии ответа LDAP-каталога на запрос задается параметров connectionTimeout (в миллисекундах). Сам описанный режим оптимального взаимодействия с пулом соединений по умолчанию включен (настройка useSyncMode в значении false). В случае нестабильной работы соединений с LDAP-каталогом рекомендуется попробовать включить синхронный режим взаимодействия с каталогом (установить useSyncMode в значении true). Примеры рекомендуемых настроек приведены ниже:

```
"id-stores" : {
  "list" : [
    {
      "type" : "LDAP",
      …
      "connectionTimeout" : 3000,
      "healthCheckInterval" : 300000,
      "useSyncMode" : false
    }
  ]
}
```

В случае подключения к Blitz Identity Provider одновременно нескольких хранилищ атрибутов может возникнуть такая ситуация, что при идентификации и аутентификации пользователя по логину и паролю в нескольких хранилищах может обнаружиться несколько учетных записей, возможно принадлежащих разным людям, с совпадающими логинами. Необходимо избегать такой ситуации при внедрении Blitz Identity Provider, и по умолчанию при выявлении такой ситуации Blitz Identity Provider при выявленных дублях будет выдавать пользователю ошибку входа, указывающую на наличие некорректной ситуации с учетной записью пользователя. Тем не менее, в ряде случае может возникнуть ситуация, когда при внедрении намеренно допускают, что по одному логину может быть найдено несколько учетных записей разных пользователей в разных хранилищах. В этом случае можно указать в

блоке настроек blitz.prod.local.idp.login режим firstSucceded в настройке authStrategy. В этом случае все найденные учетные записи будут проверены, и к какой из них первой подойдет пароль пользователя, с этой учетной записью и будет выполнен вход.

Пример настройки:

```
"login" : {
  "authStrategy" : {
    "mode" : "firstSucceeded"
  },
  …
}
```

16.1.14. Блокирование неактивных пользователей

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 (см. п. 7.4).

16.1.15. Запрет повторного использования идентификатора 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
  }
}
```

16.1.16. Настройка групп пользователей

Чтобы включить возможность просмотра групп пользователей, необходимо добавить блок настроек blitz.prod.local.idp.groups следующего вида:

```
"groups": {
  "profiles": [
    {
      "type": "mirror",
      "id": "orgs",
      "groupStore": "389ds",
      "attrsMap": {
        "name": "displayname",
      },
      "filter": "objectClass=group"
    }
  ],
  "stores": {
    "list": [
      {
        "type": "ldap_based",
        "id": "389ds",
        "desc": "Группы",
        "ldapStore": "389ds",
        "baseDN": "ou=external,ou=groups,dc=test",
        "searchScope": "SUB",
        "idAttrName": "cn",
        "membersAttrName": "uniqueMember",
        "memberOfAttrName": "memberOf",
        "newGroupAttrs": [
          {
            "attr": "objectclass",
            "format": "strings",
            "value": "top,groupOfUniqueNames,group"
          },
          {
            "attr": "dn",
            "format": "string",
            "value": "cn=${id},ou=external,ou=groups,dc=test"
          }
        ]
      }
    ]
  }
}
```

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.

16.1.17. Настройка контейнера ключей для работы с ЕСИА и Цифровым профилем

Запросы на аутентификацию через внешний поставщики идентификации ЕСИА и ЦП ЕСИА должны быть подписаны электронной подписью с использованием контейнера ключей с алгоритмами ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012. Для использования в Blitz Identity Provider необходим файловый контейнер ключа электронной подписи в формате криптопровайдера КриптоПро CSP (версия 4.0 или выше). Для получения контейнера ключей необходимо обратиться в аккредитованный удостоверяющий центр. 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.

16.1.17.1. Конвертация контейнера в Windows

Для конвертации с помощью этого способа 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[65] (версия 4.0 и новее). Установить закрытый ключ электронной подписи в реестр Windows. Для этого выполнить шаги:
   - запустить КриптоПро CSP;

[65] См.: https://cryptopro.ru/

- выбрать вкладку «Сервис» и нажать на кнопку «Посмотреть сертификаты в контейнере» (см. Рисунок 176):

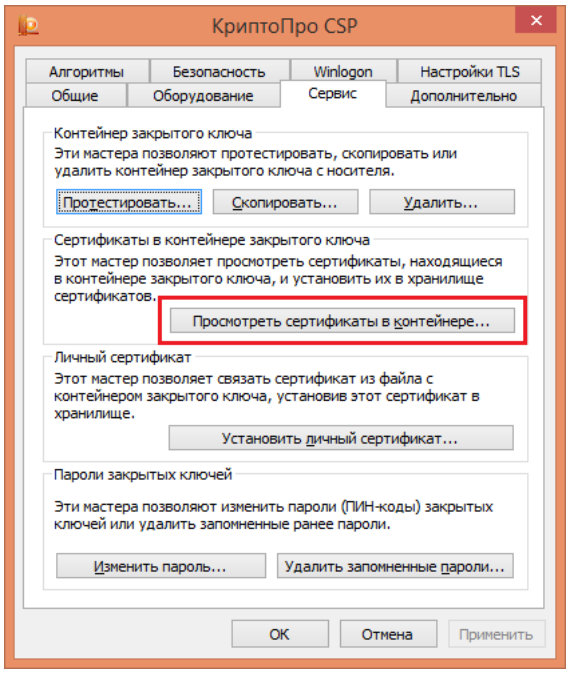

Рисунок 176 – Просмотр сертификатов в контейнере ключей

- нажать на кнопку «Обзор» (см. Рисунок 177):

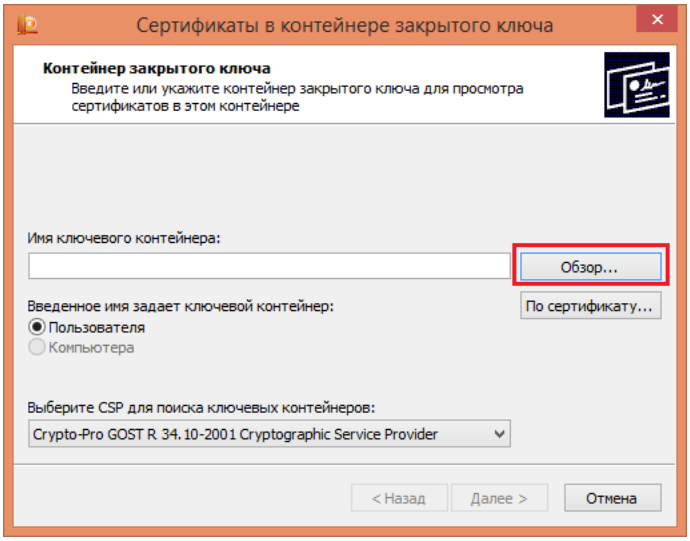

Рисунок 177 – Просмотр сертификатов в контейнере ключей (продолжение)

- выбрать контейнер с ключом ГОСТ Р 34.10-2012 (см. Рисунок 178):

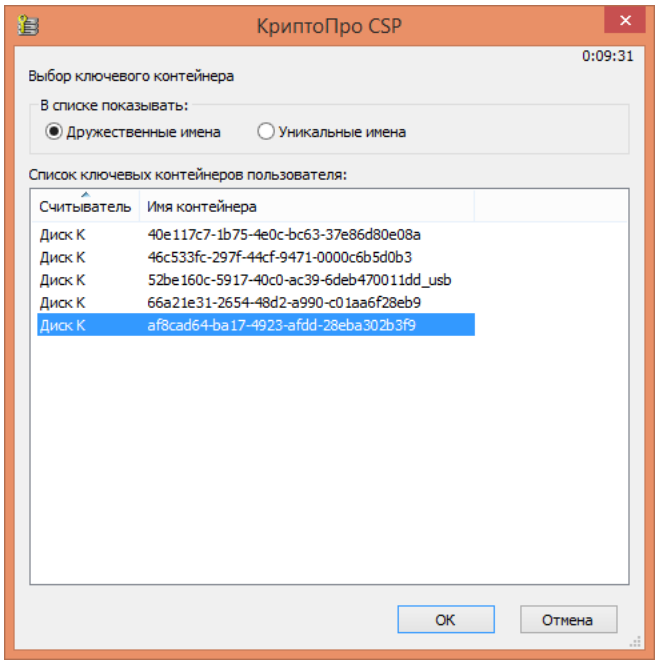

Рисунок 178 – Выбор контейнера ключа

- в открывшемся окне убедиться, что выбран нужный ключевой контейнер (см. Рисунок 179) и нажать далее:

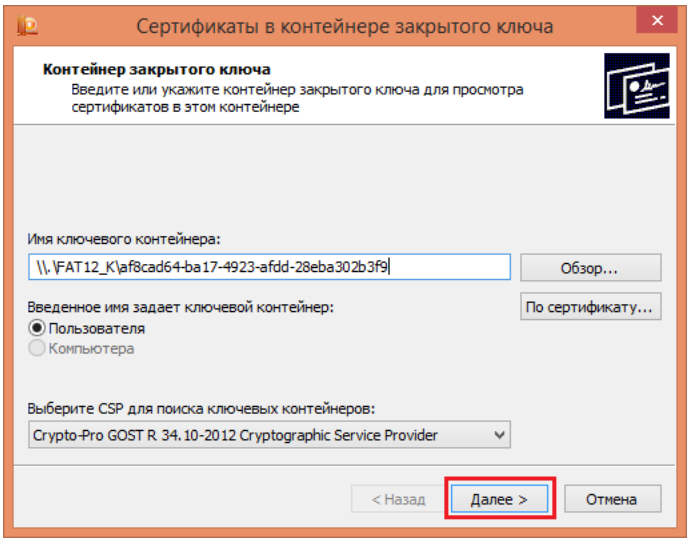

Рисунок 179 – Проверка выбранного контейнера ключа

- в окне со свойствами сертификата (см. Рисунок 180) нажать на кнопку «Установить»:

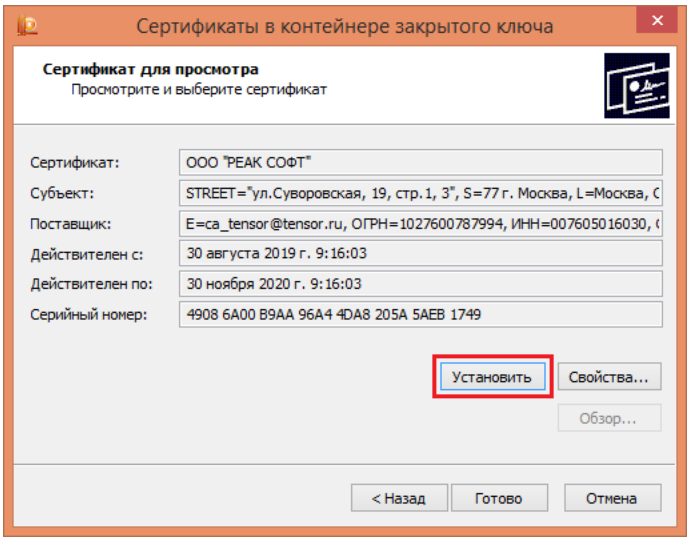

Рисунок 180 – Установка сертификата в реестр

- сообщение «Сертификат был установлен в хранилище «Личные» текущего пользователя» свидетельствует об успешном сохранении.

2. Экспортировать ключ КриптоПро из хранилища Windows в PKCS#12. Для этого приобрести и запустить утилиту P12FromGostCSP[66]. Нужна специальная версия утилиты – рекомендуется проконсультироваться с технической поддержкой Blitz Identity Provider.

   После запуска утилиты выполнить шаги:

   - выбрать сертификат (см. Рисунок 181):

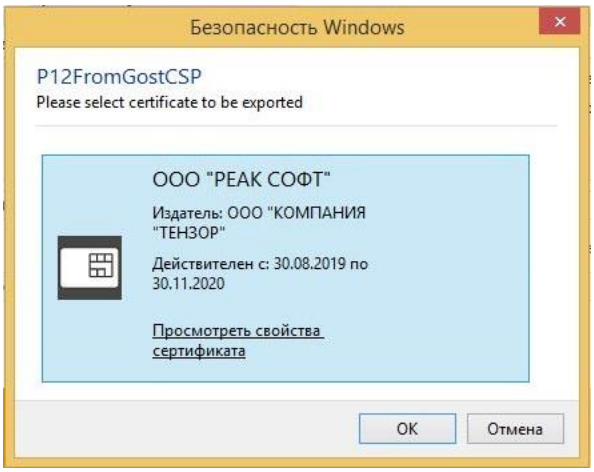

Рисунок 181 – Выбор сертификата

[66] См.: http://soft.lissi.ru/ls_product/utils/p12fromcsp/

- задать пароль от создаваемого контейнера PKCS#12 (см. Рисунок 182):

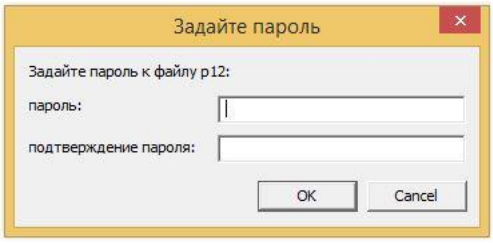

Рисунок 182 – Задание пароля от контейнера PKCS12

- указать файл (см. Рисунок 183) для сохранения PKCS#12. В качестве расширения обязательно указать .pfx.

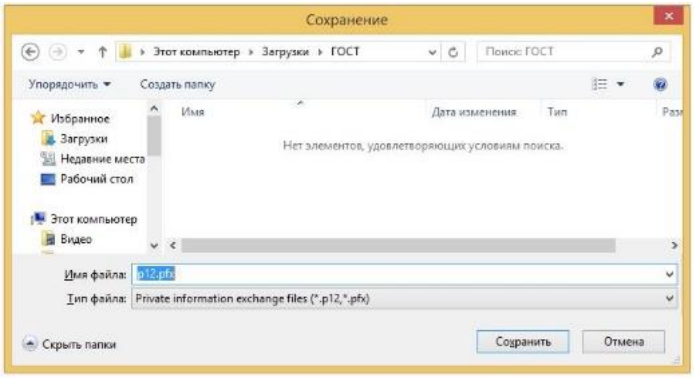

Рисунок 183 – Сохранение файла с контейнером PKCS#12

3. Установить на ПК окружение Java (JRE) версии 8.
4. Импортировать ключ из PKCS#12 в BKS-хранилище с помощью бесплатной утилиты gost-keytool[67] и актуальной версии библиотеки bcprov-jdk15on[68].

```
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;

[67] См.: https://identityblitz.ru/wp-content/uploads/2019/11/gost-keytool.zip
[68] См.: https://repo1.maven.org/maven2/org/bouncycastle/bcprov-jdk15on/

- 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.

16.1.17.2. Конвертация контейнера в Linux

Для конвертации с помощью этого способа понадобятся:

- ПК или сервер под управлением одного из следующих Linux: Astra Linux 1.7, РЕД ОС 7.3, Альт. В случае использование сервера рекомендуется использовать иной сервер, чем те, на которых развернут Blitz Identity Provider.
- Бесплатная утилита CryptoPro PFX Decoder by li0ard.

Перед началом конвертации нужно сохранить контейнер ключей КриптоПро CSP в формат pfx (файл sourcekey.pfx) и сертификат открытого ключа в формат cer (sourcecrt.cer).

Инструкция по конвертации контейнера:

1. Переключить ОС на использование ГОСТ в библиотеке OpenSSL согласно инструкции для используемой ОС: Astra Linux[69], РЕД ОС 7.3[70], Альт[71]. Инструкция далее приводится на примере РЕД ОС 7.3 с включенной настройкой для использования ГОСТ в OpenSSL.
2. Включить настройку ГОСТ в OpenSSL:

```
openssl-switch-config gost
```

3. Настроить зависимости (делается однократно перед первой конвертацией ключа на сервере):

   Установка модуля asn1:

```
pip3 install asn1
```

   Устанавливаем модуля pyderasn:

```
[fetch|wget] http://www.pyderasn.cypherpunks.ru/download/pyderasn-9.3.tar.zst
[fetch|wget] http://www.pyderasn.cypherpunks.ru/download/pyderasn-9.3.tar.zst.asc
gpg --verify pyderasn-9.3.tar.zst.asc pyderasn-9.3.tar.zst
zstd -d < pyderasn-9.3.tar.zst | tar xf -
cd pyderasn-9.3
python setup.py install
```

   Устанавливаем pygost:

```
[fetch|wget] http://www.pygost.cypherpunks.ru/pygost-5.12.tar.zst
[fetch|wget] http://www.pygost.cypherpunks.ru/pygost-5.12.tar.zst.asc
gpg --verify pygost-5.12.tar.zst.asc pygost-5.12.tar.zst
zstd -d < pygost-5.12.tar.zst | tar xf -
cd pygost-5.12
python setup.py install
```

[69] См.: https://wiki.astralinux.ru/pages/viewpage.action?pageId=27362269
[70] См.: https://redos.red-soft.ru/base/manual/safe-redos/gost-in-openssl/
[71] См.: https://www.altlinux.org/ГОСТ_в_OpenSSL

4. Загрузить утилиту CryptoPro PFX Decoder by li0ard:

```
[fetch|wget] https://raw.githubusercontent.com/li0ard/cpfx/pyderasn/cpfx.py
[fetch|wget] https://raw.githubusercontent.com/li0ard/cpfx/pyderasn/schemas.py
```

5. Конвертировать контейнер ключа из формата pfx в формат PEM с помощью CryptoPro PFX Decoder by li0ard:

```
python cpfx.py sourcekey.pfx
```

Ответ утилиты будет содержать информацию о том, в какой файл был сохранен преобразованный контейнер. Например, exported_4192476d-3e66-4963-8684-bd95d6be7967.pem.

6. Конвертировать сертификат открытого ключа из формата DER в формат PEM с помощью OpenSSL:

```
openssl x509 -in sourcecrt.cer -inform der -out sourcecrt.pem
```

7. Собрать новый pfx-контейнер из преобразованных ключа и сертификата:

```
openssl pkcs12 -export -in sourcecrt.pem -inkey 4192476d-3e66-4963-8684-bd95d6be7967.pem -out
correct.pfx -name gost2012
```

8. Установить окружение Java (JRE) версии 8.
9. Импортировать ключ из PKCS#12 в BKS-хранилище с помощью бесплатной утилиты gost-keytool[72] и актуальной версии библиотеки bcprov-jdk15on[73].

```
java -cp gost-keytool.jar:bcprov-jdk15on-1.70.jar ru.reaxoft.gost.Keytool import_pkcs12 --
srckeystore correct.pfx --srcstorepass 1234 --srckeypass 1234 --destkeystore blitz-keystore.bks --
deststoretype BKS --deststorepass pass --destalias gost2012 --destkeypass pass --srcalias gost2012
```

В этой операции используются параметры:

- srckeystore – путь к сконвертированному pfx-контейнеру (PKCS#12);
- srcstorepass – пароль от контейнера PKCS#12;
- srcalias – имя ключа в контейнере PKCS#12. Нужно указать значение gost2012;
- srckeypass – пароль к ключу в контейнере PKCS#12. Нужно указать то же самое значение, что и для параметра srcstorepass;
- destkeystore – путь к файлу BKS blitz-keystore.bks, взятому с сервера Blitz Identity Provider;
- deststoretype – тип хранилища. Нужно указать значение BKS;
- deststorepass – пароль к хранилищу BKS;
- destalias – имя (alias) ключа в BKS, например, gost2012;
- destkeypass – пароль к ключу в BKS. Нужно указать то же самое значение, что и для параметра deststorepass.

10. Заменить файл blitz-keystore.bks на сервере Blitz Identity Provider;
11. Перезапустить приложения Blitz Identity Provider.

[72] См.: https://identityblitz.ru/wp-content/uploads/2019/11/gost-keytool.zip
[73] См.: https://repo1.maven.org/maven2/org/bouncycastle/bcprov-jdk15on/

16.1.18. Вход через ЕСИА в режиме выбора сотрудника организации

Когда для входа в Blitz Identity Provider сконфигурирован внешний поставщик идентификации ЕСИА, то к обычному режиму входа пользователя (см. п. 8.9) 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 (см. Рисунок 184).

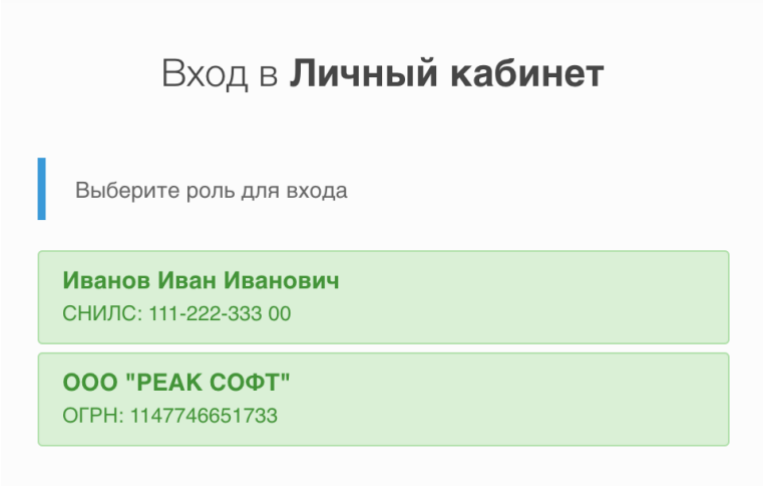

Рисунок 184 – Пример страницы выбора режима входа через ЕСИА

- Получение из ЕСИА сведений о выбранной при входе организации, автоматическое создание на основе этих сведений в LDAP-хранилище группы пользователей с атрибутами, соответствующими организации (если соответствующая организации группа не найдена в момент входа), добавление пользователя в созданную (или найденную) группу пользователей.
- Обновление атрибутов группы пользователя значениями атрибутов организации из ЕСИА в момент входа, если атрибуты в ЕСИА изменились.
- Возможность добавления в маркер доступа и маркер обновления сведений о выбранной в момент входа пользователя роли в ЕСИА (физическое лицо, индивидуальный предприниматель, должностное лицо юридического лица, должностное лицо органа государственной власти).

Для настройки режимов входа необходимо предварительно настроить в Blitz Identity Provider использование групп доступа (см. п. 16.1.16) и вход через ЕСИА (см. п. 8.9). 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" : "${org.oktmo-}"
            },
            {
                "attr" : "org_leg",
                "master" : true,
                "value" : "${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"
      },
      "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"
      ]
    },
    …
  ]
 }
}
```

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",
            "value" : "org.type"
          },
          {
            "name" : "org_ogrn",
            "value" : "org.ogrn"
          },
          {
            "name" : "org inn",
            "value" : "org.inn"
          },
          {
            "name" : "org oktmo",
            "value" : "org.oktmo"
          }
        }
      },
      …
    ]
  }
}
```

16.1.19. Настройка доверенных сертификатов поставщиков ключей безопасности FIDO2 и U2F

Blitz Identity Provider позволяет переопределить перечень промежуточных и корневых сертификатов поставщиков ключей безопасности (WebAuthn, Passkey, FIDO2, U2F) (см. п. 4.13 и п. 4.22). 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"
        }
    ],
    …
}
```

16.1.20. Настройки сервиса OIDC Discovery

Blitz Identity Provider автоматически публикует сервис OIDC Discovery[74] 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.

16.1.21. Изменение адресов вызовов внешних поставщиков идентификации

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",
        …
      },
      …
    ]
  }
}
```

16.1.22. Настройка внешнего SAML поставщика входа

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

[74] См.: https://tools.ietf.org/html/rfc8414

SAML-запрос, отправляемый внешнему поставщику идентификации;

- checkAssertionSign – определяет, необходимо ли проверять подпись SAML-утверждений, полученных от внешнего поставщика идентификации (для ПРОД-сред обязательно необходимо включать проверку подписи);
- блок credentials с настройками доступа к ключевому контейнеру, используемому для подписывания запросов к поставщику идентификации SAML. Настраивается опционально, в случае если для взаимодействия с внешним поставщиком идентификации требуется использовать отдельный контейнер ключей. Если настройка не задана, то ключи будут браться из основного keystore, настроенного в блоке blitz.prod.local.idp.keystore (при этом в качестве alias будет использоваться имя поставщика идентификации из настройки name). Задаются настройки:
  - alias – имя ключа в контейнере;
  - keystore – блок настроек, содержащий тип контейнера (type), который может быть JCEKS или BKS, а также путь к контейнеру (path) и пароль к контейнеру (password);
- idpMetaPath – путь к файлу, в котором хранятся метаданные внешнего поставщика идентификации (XML-файл с метаданными IDP);
- блок настроек userMatching – задает правила сопоставления учетных записей:
  - в настройке type – признак, что используется базовая (значение builder) настройка связывания учетных записей;
  - в настройке mapping – правила сопоставления учетных записей из внешнего SAML-поставщика идентификации учетным записям в Blitz Identity Provider;
  - в настройке matchingRules – правила переноса SAML-утверждений из внешнего поставщика идентификации в атрибуты учетной записи в Blitz Identity Provider;
  - requireLogInToBind – признак «Предлагать пользователю ввести логин и пароль для привязки, если учетная запись не была идентифицирована»;
  - strictMatching – признак «Требовать ввод пароля, если учетная запись была идентифицирована»;
  - uniqueMatching – признак «Для привязки должна быть найдена только одна учетная запись по заданным правилам соответствия».

Пример настроек внешнего поставщика идентификации:

```
"federation" : {
  "points" : {
    "saml" : [
      {
       "name" : "demo-idp",
        "humanReadableName" : "External SAML IDP",
        "clientId" : "login.company.com",
        "signAuthnReq" : true,
```

```
        "checkAssertionSign" : true,
        " credentials" : {
            "alias" : "demo-idp",
            "keyStore" : {
              "password" : "*****",
              "path" : "demo-idp-key.jks",
              "type" : "JCEKS"
            }
        },
        "idpMetaPath" : "demo-idp-metadata.xml",
        "userMatching" : {
          "type" : "builder",
          "mapping" : [
            {
              "attr" : "urn:saml:mail",
              "master" : false,
              "value" : "${email}"
            }
          ],
          "matchingRules" : [
            [
              {
                "attr" : "urn:saml:mail",
                "value" : "${email}"
              }
            ]
          ],
          "requireLogInToBind" : false,
          "strictMatching" : false,
          "uniqueMatching" : false
        }
      }
    ],
    …
  }
}
```

После создания настроек внешнего поставщика необходимо включить его в списке доступных внешних поставщиков идентификации. Для этого в блок настроек blitz.prod.local.idp.login в перечень методов аутентификации (methods) в список внешних поставщиков входа externalIdps добавить внешний поставщик с fedPoint, соответствующий настроенному. Пример настройки для включения внешнего поставщика идентификации с типом saml и именем demo-idp:

```
"login" : {
  …
  "methods" : {
    …
    "externalIdps" : {
      "idps" : [
        …
        {
          "fedPoint" : "saml:demo-idp"
        },
        …
      ],
      …
    },
    …
  },
  …
}
```

Настроить логотип для кнопки входа через внешний поставщик входа согласно инструкции в п. 16.2.5.

16.1.23. Настройка внешнего поставщика входа 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 расшифрования ответов (значение cn из subject сертификата Blitz Identity Provider, зарегистрированного в СУДИС);
- idpEntityId – EntityID СУДИС;
- sidAttrName – имя атрибута, используемого в качестве идентификатора учетной записи в СУДИС;
- ssoServiceLocation – адрес обработчика запроса запросов на аутентификацию в СУДИС;
- sloServiceResponseLocation – адрес ответа на запросы логаута, инициированные в СУДИР из СУДИС;
- sloServiceLocation – адрес обработчика логаута в СУДИС при инициировании логаута из СУДИР в СУДИС;
- блок настроек userMatching – задает правила сопоставления учетных записей:
  - в настройке type – признак, что используется базовая (значение builder) настройка связывания учетных записей;
  - в настройке mapping – правила сопоставления учетных записей из внешнего SAML-поставщика идентификации учетным записям в Blitz Identity Provider;
  - в настройке matchingRules – правила переноса SAML-утверждений из внешнего поставщика идентификации в атрибуты учетной записи в Blitz Identity Provider;

- requireLogInToBind – признак «Предлагать пользователю ввести логин и пароль для привязки, если учетная запись не была идентифицирована»;
- strictMatching – признак «Требовать ввод пароля, если учетная запись была идентифицирована»;
- uniqueMatching – признак «Для привязки должна быть найдена только одна учетная запись по заданным правилам соответствия».

Пример настроек внешнего поставщика идентификации:

```
"federation" : {
  "points" : {
    "sudis" : [
    {
        "clientId" : "<ENTITY ID BLITZ>",
        "credentials" : {
            "endpoint" : "http://<CMS_HOST>:<CMS_PORT>",
            "fingerprint" : "{SHA1}<FINGERPRINT SUDIS CERT>",
            "recipientKey" : "<SUDIS_KEY SUBJECT.CN>",
            "senderKey" : "<BLITZ_KEY SUBJECT.CN>"
        },
        "humanReadableName" : "SUDIS",
        "idpEntityId" : "http://…",
        "name" : "sudis",
        "sidAttrName" : "oid",
        "ssoServiceLocation" : "http://…/idp/profile/SAML2/POSTGOST/SSO",
        "sloServiceLocation" : "http://…/idp/Logout",
        "sloServiceResponseLocation" : "http://…/idp/profile/SAML2/Redirect/SLO",
        "userMatching" : {
            …
        }
    }
]
```

После создания настроек внешнего поставщика необходимо включить его в списке доступных внешних поставщиков идентификации. Для этого в блок настроек blitz.prod.local.idp.login в перечень методов аутентификации (methods) в список внешних поставщиков входа externalIdps добавить внешний поставщик с fedPoint, соответствующий настроенному. Пример настройки для включения внешнего поставщика идентификации с типом sudis и именем sudis:

```
"login" : {
  …
  "methods" : {
    …
    "externalIdps" : {
      "idps" : [
        …
        {
          "fedPoint" : "sudis:sudis"
        },
        …
      ],
      …
    },
    …
  },
  …
}
```

16.1.24. Включение режима 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.

16.1.25. Настройка передачи 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 logback;
    - kafka – запись производится на сервер очередей Kafka.
  - enabled – необязательная настройка – определяет, включен или нет приемник;
  - include – необязательная настройка – перечисляются типы событий безопасности (см. таблицу 6), по которым осуществляется запись в приемник. Если настройка не указана, то пишутся все события безопасности;
  - exclude – необязательная настройка – перечисляются типы событий безопасности (см. таблицу 6), 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;

- 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[75].

Пример блока:

[75] См.: https://kafka.apache.org/documentation/#producerconfigs

```
"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",
                    "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[76]. Пример настройки логгера AUDIT в файле конфигурации logback.xml:

```
…
    <appender name="AUDIT" class="ch.qos.logback.core.rolling.RollingFileAppender">
        <file>${dir.logs}/audit-${app.name}.log</file>
        <encoder>
            <pattern>%date - [%level] -[%file:%line] - %message%n%xException{20}</pattern>
        </encoder>
        <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
            <fileNamePattern>${dir.logs}/archive/audit-${app.name}.%d{yyyy-MM-
dd}.log.gz</fileNamePattern>
            <maxHistory>90</maxHistory>
            <totalSizeCap>5GB</totalSizeCap>
        </rollingPolicy>
    </appender>

    <logger name="AUDIT" additivity="false">
        <appender-ref ref="AUDIT" />
    </logger>
…
```

Пример записи в логе:

```
2023-11-20 13:29:47,170 - [INFO] -[LoggerEventEmitterDriver.scala:37] -
{"ip st":"Ташкент","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":"Ташкент","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"}
```

Набор атрибутов записи может отличаться в зависимости от типа события безопасности и особенностей процесса входа. Назначения атрибутов в записи аудита приведены в таблице ниже:

Таблица 6

Назначение атрибутов в записи аудита

Атрибут	**Назначение и возможные значения**
id	Идентификатор записи о событии безопасности
type	Тип события безопасности: - admin_added – добавлен администратор - admin_pswd_changed – изменен пароль администратора - admin_removed – удален администратор - admin_roles_changed – изменены роли администратора - app_password_changed – задан пароль для приложения - attribute_changed – добавлен, изменен или удален атрибут - attribute_confirmed – атрибут подтвержден - auth – выполнена аутентификация (при OAuth 2.0 Resource Owner Password Credentials) - auth_failed – ошибка аутентификации - auth_req – запрос на аутентификацию - authz_granted – выдано OAuth-разрешение - authz_rejected – отказано в выдаче OAuth-разрешения - authz_revoked – отозвано OAuth-разрешение

[76] См.: https://logback.qos.ch/documentation.html

Атрибут	Назначение и возможные значения
	– 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

Атрибут	Назначение и возможные значения
	– web_authn_reg_key – добавлен ключ безопасности – web_authn_revoke_key – удален ключ безопасности
alt_pswd_cause	Причина, по которой пользователя просили сменить пароль Возможные значения: – password_expired - пароль просрочен – password_reset - пароль нужно сменить при первом входе – password_policy_violated - пароль не соответствует парольной политике
attr_name	Имя установленного, удаленного или измененного атрибута.
auth_methods	Содержит список пройденных пользователем методов аутентификации. Возможные значения: – password – парольная аутентификация – spnego – вход с помощью сеанса ОС – x.509 – вход с помощью средства электронной подписи – qrCode – вход по QR-коду – tls – вход с помощью HTTP-заголовков прокси-сервера – webAuthn – вход или подтверждение входа с помощью ключей безопасности – css – автоматический вход по результатам регистрации пользователя или восстановления пароля – sms – одноразовый пароль по SMS – email – одноразовый пароль по email – hotp – второй фактор аутентификации с помощью аппаратного брелока – totp – второй фактор аутентификации с помощью программного TOTP-генератора кодов подтверждения – externalIdps:<type>:<name> – вход с помощью внешнего поставщика идентификации (соцсети или ЕСИА) – userApp – вторичная аутентификация в мобильном приложении – outside_%NAME% – внешний метод входа с именем %NAME% Наличие перед методом префикса cls: означает, что вход был выполнен с помощью долгосрочной сессии, а ранее при первичном входе использовались те методы входа, что перечислены после cls:
auth_soft_id	Приложение-аутентификатор (при входе по QR-коду)
authnDone	Проводилась ли аутентификация при этом входе
captcha_passed	Признак, что при входе спрашивалась CAPTCHA
client_auth_method	Способ аутентификации вызвавшего Blitz Identity Provider приложения: – internal – для событий, вызванных внутренними приложениями Blitz Identity Provider – x.509 – для событий, вызванных SAML-приложениями, при условии, что SAML-запрос пришел подписанным – Basic – для приложений, вызывающих REST-сервисы,

Атрибут	Назначение и возможные значения
	использующие Basic-авторизацию – redirectUri – для приложений, которые идентифицировали себя в URL (например, указали свой client_id в URL-параметре), но чья аутентификация не проводилась (достоверно не известно, что это действительно вызывает Blitz Identity Provider именно это приложение) – Bearer – использование access_token для аутентификации мобильным приложением с динамическими client_id/client_secret
dcId	Динамический client_id
device	ID устройства
deviceFingerprint	Отпечаток устройства
dTyp	Тип устройства (при динамической регистрации)
email	Адрес электронной почты
entry_point	Тип интерфейса, использованного для регистрации пользователя: – WEB – при регистрации из веб-приложения Blitz Identity Provider – REST – при регистрации через REST-сервисы Blitz Identity Provider
error	Ошибка (при неуспешных событиях)
ext_account_id	Идентификатор внешней учетной записи
ext_account_name	Имя внешнего поставщика идентификации
ext_account_type	Тип внешнего поставщика идентификации
failed_method	Указывает, какой метод аутентификации не смог пройти пользователь
group_id	Идентификатор группы пользователей
group_profile	Идентификатор профиля использования групп пользователей
id_store	Хранилище учетной записи
ip	IP адрес пользователя
ip_ctr	Страна по IP адресу
ip_st	Регион по IP адресу
ip_ct	Город по IP адресу
ip_lat	Широта по IP адресу
ip_lng	Долгота по IP адресу
ip_rad	Окрестность по IP адресу

Атрибут	**Назначение и возможные значения**
lp_id	Идентификатор приложения (EntityId для SAML или client_id для OIDC), вызвавшего Blitz Identity Provider.
mobile	Номер мобильного телефона
module	Идентификатор измененного блока настроек
new_attr_value	Новое значение установленного или измененного атрибута
new_factor	Новое значение признака, указывающего на необходимость проверки второго фактора аутентификации
new_roles	Роли, добавленные учетной записи администратора
oauth_scopes	Список разрешений, которые выдал или отозвал пользователь
object_id	Идентификатор объекта операции (пользователь, по которому выполнялась операция)
old_attr_value	Прежнее значение удаленного или измененного атрибута
old_factor	Прежнее значение признака, указывающего на необходимость проверки второго фактора аутентификации
old_roles	Роли, отозванные у учетной записи администратора
origin_app	Идентификатор приложения, инициировавшего регистрацию пользователя или восстановление пароля
process_id	Идентификатор процесса
protocol	Протокол взаимодействия приложения с Blitz Identity Provider. Возможные значения: - SAML – для SAML и WS-Federation - OAuth – для OpenID Connect и OAuth 2.0 - simple – для прокси-аутентификации - internal – для входа в Личный кабинет (_blitz_profile)
pswd_changed	Признак, что рекомендовалась смена пароля
pswd_tmp_locked	Признак, что была временная блокировка
recovery_contact	Указанный при восстановлении контакт (email или номер мобильного телефона)
recovery_type	Тип восстановления пароля: email или mobile
right_name	Название права доступа
roles	Роли учетной записи администратора
session_id	Уникальный идентификатор сессии пользователя. Позволяет коррелировать все события пользователя, выполненные им в рамках общей пользовательской сессии.
subject_id	Идентификатор субъекта операции (пользователь, который вызвал операцию)

Атрибут	Назначение и возможные значения
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
user_agent	Данные о пользовательском устройстве (UserAgent)
wa_key_id	Идентификатор ключа безопасности
wa_key_name	Имя ключа безопасности
withDelay	Включалась задержка при входе

16.1.26. Изменение системных имен полей ввода логина и пароля

По умолчанию Blitz Identity Provider на странице ввода логина и пароля называет поля ввода логина и пароля идентификаторами login и password. При внедрении Blitz Identity Provider при миграции с существующей системы входа, в которой использовались другие названия полей, может существовать требование, что нужно сохранить в Blitz Identity Provider прежние используемые названия полей. Это может быть полезно, так как некоторые браузеры, сохранившие логины и пароли пользователей, и использующие их для автоподстановки, смогут продолжать осуществлять автоподстановку сохраненных значений и при переключении системы входа на использование Blitz Identity Provider, при условии сохранения домена страницы входа и названия полей на странице входа.

Для установки требуемых названий полей ввода логина и пароля необходимо в блок настроек blitz.prod.local.idp.password добавить следующие настройки:

- loginInputName – идентификатор поля ввода логина на странице входа;
- passwordInputName – идентификатор поля ввода пароля на странице входа.

Пример настроек:

```
"password" : {
    …
    "loginInputName" : "j_username",
    "passwordInputName" : "j password",
    …
}
```

16.1.27. Настройка использования базы геоданных

Можно подключить к Blitz Identity Provider базу данных в формате mmdb[77] с геоданными. В этом случае Blitz Identity Provider при регистрации событий безопасности, а также при запоминании устройств и браузеров пользователя дополнительно к сохранению

[77] См.: https://www.maxmind.com/en/geoip2-databases

IP-адреса будет записывать соответствующие IP-адресу данным о стране, регионе и городе, а также широту, долготу и радиусу точности, полученные из базы геоданных.

Сохраненные геоданные будут показываться администратору в консоли управления. Также можно включить отображение геоданных пользователю в «Личном кабинете» и включить их в тексты уведомлений, отправляемых по SMS или email.

Для подключения базы данных с геоданными необходимо выложить на серверах с Blitz Identity Provider файл формата mmdb с базой данных, а также создать блок настроек blitz.prod.local.idp.geoIp со следующими настройки в блоке driver:

- type – тип базы с геоданными. Поддерживается только тип geoIp2-db;
- path – путь на сервере к файлу с базой геоданных в формате mmdb.

Пример настроек:

```
"geoIp": {
    "driver": {
        "type": "geoIp2-db",
        "path": "geoIp/GeoIP2-City.mmdb"
    }
}
```

16.1.28. Настройки вспомогательных приложений (pipes)

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.

Для настройки отображения пользователю объявления нужно:

- создать процедуру входа на основе стандартной (см. п. 6.2.8);
- в конфигурационном файле blitz.conf добавить раздел blitz.prod.local.idp.built-in-pipes, в котором назначить вспомогательному приложению с типом info идентификатор id и тип объявления type. Возможны следующие типы объявлений:
  - news – отображается одна кнопка
  - agreement – отображается две кнопки.
- настроить тексты кнопок и объявления (см. п. 16.2.8).

  Пример конфигурации вспомогательных приложений info с идентификаторами alarm и user_agreement:

```
"built-in-pipes": {
    "info": [
        {
            "id": "alarm",
            "type": "news"
        },
        {
            "id": "user_agreement",
            "type": "agreement"
        }
    ]
}
```

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.

Для настройки отображения пользователю списка значений нужно:

- создать процедуру входа на основе стандартной (см. п. 6.2.12);
- в конфигурационном файле blitz.conf добавить раздел blitz.prod.local.idp.built-in-pipes, в котором назначить вспомогательному приложению с типом choice идентификатор id и имя атрибута claim, в который необходимо сохранять результат выбора.
- настроить тексты кнопок и объявления (см. п. 16.2.8).

Пример конфигурации вспомогательного приложения choice:

```
"built-in-pipes": {
    "choice": [
        {
            "id": "select_value",
            "claim": "role"
        }
    ]
}
```

16.1.29. Одновременное использование нескольких СУБД

В Blitz Identity Provider можно настроить одновременное использование СУБД Couchbase Sever и СУБД PostgreSQL для хранения разных типов объектов. Для этого необходимо в блоке настроек blitz.prod.local.idp.stores задать следующие настройки:

- default_type – используемая по умолчанию СУБД. Возможные значения: cb – Couchbase Server, jdbc – PostgreSQL или иная реляционная СУБД с поддержкой JDBC;
- list-of-types – идентификаторы классов объектов Blitz Identity Provider и используемые для размещения соответствующих им объектов СУБД (cb или jdbc). Включать в настройку нужно только те классы объектов, которые размещаются в СУБД, отличной от указанной в default_type. Доступны следующие классы объектов:
  - user-store – атрибуты учетных записей;
  - access-token-store – маркеры безопасности;
  - refresh-token-store – маркеры обновления;
  - 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"
  ]
}
```

16.1.30. Хранение настроек подключенных 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 каталог 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 %.

Например, для приложения с идентификатором https://example.com должен быть создан каталог с именем https%##example.com.

Обязательно должны быть созданы каталоги для служебных приложений _blitz_console, _blitz_idp, _blitz_reg, _blitz_recovery, _blitz_profile

3. Внутри каждого каталога приложения должен быть создан файл с именем app.conf, содержащий конфигурацию приложения из исходного blitz.conf. Раздел должен называться app, а не значение appId, как было в blitz.conf. Внутри каталога приложения также будет создан скрытый каталог .snapshot, куда будут копироваться предыдущие конфигурации приложений в случае изменения настроек приложения через консоль или API.

   Пример конфигурационного файла app.conf:

```
##############################################################################################
# version: 822
# modified: 2023-08-20 21:17:27 MSK
# author: admin
# ip: 127.0.0.1
# user agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 …
##############################################################################################
{
    "app": {
        "domain": "https://identityblitz.ru",
        "name": "Тестовое приложение",
        "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.

16.1.31. Настройка работы с контрольным вопросом

Для того, чтобы в методах аутентификации на вкладке «Второй фактор» появился метод аутентификации «Подтверждение ответом на контрольный вопрос», необходимо в блоке настроек blitz.prod.local.idp.login в блоке factors во втором списке добавить блок настроек с методом secQsn:

```
"login" : {
    "factors" : [
        [
            …
        ],
        [
```

```
            {
                "enabled" : false,
                "method" : "secQsn"
            },
            …
        ]
    ],
    …
}
```

Чтобы создать справочник контрольных вопросов, необходимо:

1. Создать на сервере директорию /etc/blitz-config/custom_messages/dics;
2. Создать 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
```

Число в справочнике используется для сортировки при отображении списка контрольных вопросов пользователю.

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 указать имя справочника securityQuestions. Например:

```
"dics" : {
    "dir" : "custom messages/dics",
    "names" : [
        "securityQuestions"
    ]
}
```

16.1.32. Включение метода автоматической идентификации

Для того чтобы метод автоматической идентификации (п. 4.14) отображался на вкладке **Аутентификация** -> **Первый фактор**, выполните следующие действия:

1. Откройте файл конфигурации /usr/share/identityblitz/blitz-config/blitz.conf. 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. Можно добавить несколько методов.

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


   "login" : {
       "factors" : [
           [
```

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

2. Перезагрузите сервисы.

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

16.2. Настройки текстов интерфейса

16.2.1. 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 (см. Рисунок 185).

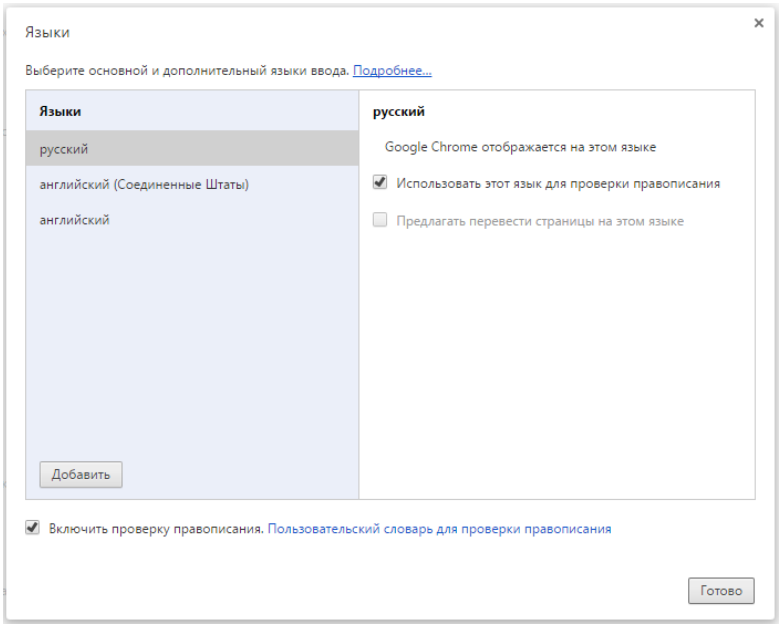

Рисунок 185 – Настройка языка для браузера Chrome

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:

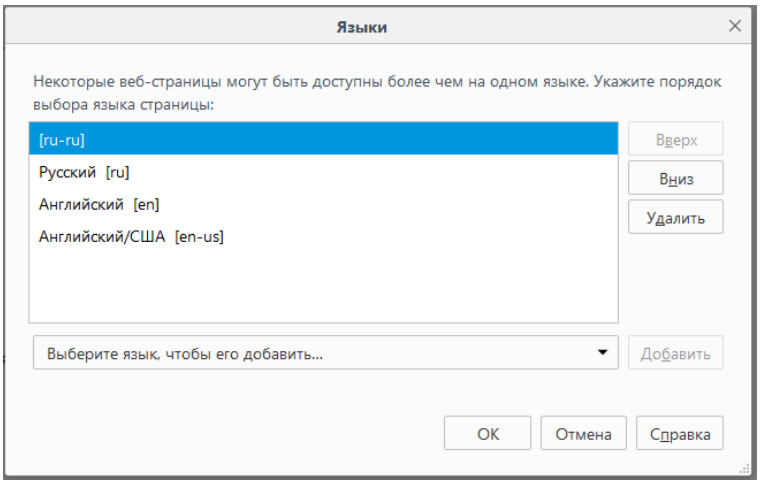

Рисунок 186 – Настройка языка для браузера Firefox

Дополнительно возможно провести настройку языка через конфигурационный файл blitz.conf. Для этого следует отредактировать раздел для настройки языка blitz.prod.local.idp.lang со следующими параметрами:

- languages – список доступных языков. Первый язык в списке считается языком по умолчанию;
- portal-lang-cookie – имя (name) и домен установки (domain) cookie с текущим языком портала (опциональный). Если портальная cookie задана, то смена языка в Blitz Identity

Provider сохраняется в указанной cookie;

- ignore-browser – включен или нет режим игнорирования языка браузера;
- lang-variants – перечень идентификаторов специальных наборов строк для отдельных приложений (см. п. 16.2.7).

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.

16.2.2. Модификация текстовых сообщений 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.

16.2.3. Кастомизация текстов для разных методов автоматической идентификации

Если вы используете несколько методов автоматической идентификации (п. 4.14), следует провести кастомизацию текстов интерфейса для каждого из них, руководствуясь алгоритмом в п. 16.2.2.

В идентификатор текстовой строки понадобится включить имя метода или идентификатор метода. Имя метода определено 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 (см. п. 16.1.32).

Для кастомизации используются следующие способы и строки:

1. Кастомизация формы входа 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
```

2. Кастомизация отображения метода в списке доступных методов 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
```

3. Кастомизация отображения метода в списке методов в консоли управления 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.
```

4. Кастомизация конфигурации метода в консоли 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>
```

5. Отображение результата 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
```

6. 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
```

16.2.4. Модификация шаблонов писем и SMS-сообщений

Шаблоны писем представляют собой текстовые строки, сохраняемые аналогично обычным строкам в веб-интерфейсе. Их изменение производится аналогичным образом (см п. 16.2.2 документа).

Используется унифицированный формат кодов сообщений, который имеет вид:

```
message.$[группа_сообщений].$[тип_сообщения].$[вариация].$[канал].$[часть]
```

Используются группы сообщения:

- notif – для информационных сообщений;

- auth – для взаимодействия с пользователем при аутентификации;
- reg – для взаимодействия с пользователем при регистрации;
- recovery – для взаимодействия с пользователем при восстановлении доступа;
- profile – для взаимодействия с пользователем в Личном кабинете;
- api – для взаимодействия с пользователем при использовании API;

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 почты. 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.

В таблицах ниже представлены описания типов сообщений из различных групп.

Таблица 7

Типы сообщений из группы «информирование» (notif)

Тип сообщения	Описание	Параметры
login_unknown_device	Информирование пользователя о входе с неизвестного устройства	– device – код устройства – device.msg – название устройства, вычисленное через

Тип сообщения	Описание	Параметры
		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)
link_social_network	Информирование пользователя о присоединении к	– fp.humanReadableName – название внешнего поставщика идентификации

Тип сообщения	Описание	Параметры
	социальной сети	– атрибуты пользователя
change_pwd	Информирование пользователя о смене пароля	– атрибуты пользователя
changed_pwd_to_object	Информирование о смене пароля в зависимой учетной записи	– атрибуты зависимой учетной записи с префиксом obj.
access_recovery	Информирование пользователя о восстановлении пароля	– атрибуты пользователя
access_recovery_by_object	Информирование о восстановлении пароля в зависимой учетной записи	– атрибуты зависимой учетной записи с префиксом obj.
set_2factor_auth	Информирование пользователя о назначении второго фактора аутентификации	– method – код метода аутентификации – method.msg – имя метода аутентификации, полученное через строку msg(message.method.name.$[method] – атрибуты пользователя
granted_access_to	Информирование субъекта о предоставление доступа к объекту	– blitz_right – код права доступа – атрибуты субъекта – атрибуты объекта с префиксом obj.
granted_access_on	Информирование объекта о предоставление доступа к нему	– blitz_right – код права доступа – атрибуты объекта – атрибуты субъекта с префиксом obj.
revoked_access_to	Информирование субъекта об отзыве доступа к объекту	– blitz_right – код права доступа – атрибуты субъекта – атрибуты объекта с префиксом obj.
revoked_access_on	Информирование объекта об отзыве доступа к нему	– blitz_right – код права доступа – атрибуты объекта – атрибуты субъекта с префиксом obj.

Тип сообщения	Описание	Параметры
on_registration	Информирование пользователя о регистрации его учетной записи	– _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.
```

Таблица 8

Типы сообщений из группы «регистрация» (reg)

Тип сообщения	Описание	Параметры
vrf_code	Отправка кода подтверждения контакта при регистрации	– 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)	– 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)	– pwd – сгенерированный пароль – req.ip – IP-адрес – req.userAgent – userAgent пользователя – cfg.domain – домен – атрибуты пользователя из контекста регистрации с префиксом attrs.

Таблица 9

Типы сообщений из группы «восстановление доступа» (recovery)

Тип сообщения	Описание	Параметры
vrf_code	Отправка кода подтверждения контакта при восстановлении доступа	– code – код подтверждения – link – ссылка для подтверждения (только для email)

Таблица 10

Типы сообщений из группы «аутентификация» (auth)

Тип сообщения	Описание	Параметры
vrf_code	Отправка кода подтверждения мобильного номера (каналы: SMS/push)	– code – код подтверждения

Таблица 11

Типы сообщений из группы «личный кабинет» (profile)

Тип сообщения	Описание	Параметры
vrf_code	Отправка кода подтверждения контакта при изменении его в Личном кабинете	– attr.msg – наименование атрибута в форме профиля – attr – код атрибута – link – ссылка для подтверждения (только для email) – code – код подтверждения

Таблица 12

Типы сообщений из группы «программный интерфейс» (api)

Тип сообщения	Вариации	Описание	Параметры
vrf_code	– $attr.$rpId – отдельно для данного приложения и атрибута – $attr – отдельно для данного атрибута	Отправка кода подтверждения контакта через API	– code – код подтверждения – link – ссылка (только для email) – attr.value – новый контакт (email или мобильный номер) – attr – код атрибута контакта

16.2.5. Настройка логотипов кнопок входа через внешние поставщики идентификации

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 (см. п. 8).

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
```

16.2.6. Модификация имен 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
```

16.2.7. Модификация сообщений для разных приложений

Возможно изменение всех текстовых сообщений и шаблонов таким образом, чтобы использовались специфические тексты и 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 в него специфические строки для данного приложения (подробнее см. п. 16.2.2). Все остальные строки будут взяты из базы строк по умолчанию.
3. Отредактировать файл blitz.conf следующим образом:
   - в разделе blitz.prod.local.idp.apps файла найти идентификатор приложения, который должен использовать созданный файл шаблона;
   - добавить в настройки приложения параметр вида "lang-variant" : "special1", где special1 – использованная для маркировки шаблона последовательность символов.

   Пример:

```
"demo-application" : {
    "domain" : "http://testdomain.ru",
    "lang-variant" : "special1",
    "name" : "test",
    "oauth" : {
        "autoConsent" : false,
        "clientSecret" : "1234567890",
        "defaultScopes" : [],
        "enabled" : true,
        "redirectUriPrefixes" : [
            "http://localhost"
        ]
    },
    "theme" : "default"
}
```

4. Зарегистрировать в разделе blitz.prod.local.idp.lang в настройке lang-variant все используемые для маркировки различных приложений последовательности символов (special1, special2).

После этого при входе в приложение будет использоваться специально созданный файл сообщений.

16.2.8. Настройка сообщений вспомогательных приложений (pipes)

В Blitz Identity Provider можно настроить сообщения вспомогательного приложения, выпускающего ключ безопасности (Passkey, WebAuthn, FIDO2) при входе пользователя. Можно настроить разные тексты сообщений в зависимости от устройств (операционных систем), используемых пользователем. Для этого в нужно создать в директории custom_messages в файле messages строки, имена которых соответствуют следующим паттернам:

- pipes.conf.webAuthn.addKey.<message-path>.<device-type>.<os>;
- login.outside.flow.error.internal.webAuthn.addKey.<device-type>.<os>.

В качестве <message-path> указывается имя строки (см. ниже пример). В качестве <device-type> указывается тип устройства: mobile, tablet, iphone, pc, ipad. В качестве <os> можно указывать: Android, iOS, Windows, macOS, Linux, ChromeOS. Если для <device-type> и <os> не определена частная строка, то берется более общая строка.

Примеры строк:

```
pipes.conf.webAuthn.addKey.page.title.pc.macOS=Вход по Touch ID
pipes.conf.webAuthn.addKey.head.title.pc.macOS=Вход по Touch ID
pipes.conf.webAuthn.addKey.info.pc.macOS=Использовать Touch ID или пароль компьютера Mac для входа в
приложения?
pipes.conf.webAuthn.addKey.finishInfo.pc.macOS=Вход по Touch ID для учетной записи настроен. Нажмите
кнопку "Продолжить"
pipes.conf.webAuthn.addKey.name.pc.macOS=Touch ID на Mac
login.outside.flow.error.internal.webAuthn.addKey.pc.macOS=Произошла ошибка при настройке входа по
Touch ID на Mac

pipes.conf.webAuthn.addKey.page.title.pc.Windows=Вход через Windows Hello
pipes.conf.webAuthn.addKey.head.title.pc.Windows=Вход через Windows Hello
pipes.conf.webAuthn.addKey.info.pc.Windows=Использовать PIN-код компьютера, распознавание лица или
отпечатка пальца для входа в приложения?
pipes.conf.webAuthn.addKey.finishInfo.pc.Windows=Вход через Windows Hello для учетной записи
настроен. Нажмите кнопку "Продолжить"
pipes.conf.webAuthn.addKey.name.pc.Windows=Windows Hello
login.outside.flow.error.internal.webAuthn.addKey.pc.Windows=Произошла ошибка при настройке входа
через Windows Hello

pipes.conf.webAuthn.addKey.page.title.iphone.iOS=Вход по Face ID
pipes.conf.webAuthn.addKey.head.title.iphone.iOS=Вход по Face ID
pipes.conf.webAuthn.addKey.info.iphone.iOS=Использовать Face ID или Touch ID телефона для входа в
приложения?
pipes.conf.webAuthn.addKey.finishInfo.iphone.iOS=Вход через Face ID для учетной записи настроен.
Нажмите кнопку "Продолжить"
pipes.conf.webAuthn.addKey.name.iphone.iOS=Face ID на iPhone
login.outside.flow.error.internal.webAuthn.addKey.iphone.iOS=Произошла ошибка при настройке входа
через Face ID

pipes.conf.webAuthn.addKey.page.title.ipad.iOS=Вход по Touch ID
pipes.conf.webAuthn.addKey.head.title.ipad.iOS=Вход по Touch ID
pipes.conf.webAuthn.addKey.info.ipad.iOS=Использовать Touch ID планшета для входа в приложения?
pipes.conf.webAuthn.addKey.finishInfo.ipad.iOS=Вход через Touch ID для учетной записи настроен.
Нажмите кнопку "Продолжить"
pipes.conf.webAuthn.addKey.name.ipad.iOS=Touch ID на iPad
login.outside.flow.error.internal.webAuthn.addKey.ipad.iOS=Произошла ошибка при настройке входа
через Touch ID

pipes.conf.webAuthn.addKey.page.title.mobile.Android=Вход по распознаванию лица или отпечатка пальца
pipes.conf.webAuthn.addKey.head.title.mobile.Android=Вход по распознаванию лица или отпечатка пальца
pipes.conf.webAuthn.addKey.info.mobile.Android=Использовать распознавание лица или отпечатка пальца
для входа в приложения?
```

```
pipes.conf.webAuthn.addKey.finishInfo.mobile.Android=Вход через распознавание лица или отпечатка
пальца для учетной записи настроен. Нажмите кнопку "Продолжить"
pipes.conf.webAuthn.addKey.name.mobile.Android=Smart Lock на Android
login.outside.flow.error.internal.webAuthn.addKey.mobile.Android=Произошла ошибка при настройке
входа через 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 1-й кнопки (для вспомогательных приложений с типом "agreement")
pipes.info.disagree.{id} - название 2-й 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 - пекст 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
```

16.3. Файлы настроек консоли управления

Консоль управления настраивается с помощью файлов console.conf и credentials. Далее

в подразделах описаны возможные настройки.

16.3.1. Настройка входа в консоль управления 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 по логину/паролю учетных записей, заведенных в разделе «Администраторы» (см. п. 2.1.12);
- 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 необходимо:

- В настройках внешнего поставщика идентификации (SSO) зарегистрировать приложение. В разрешенные префиксы возврата (redirect_uri) нужно, чтобы был прописан домен установки Blitz Identity Provider. По итогам регистрации получить client_id и client_secret приложения для консоли управления;
- в конфигурационном файле console.conf создать 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), d 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 187); internal – вход только по логину и паролю из настроек консоли управления (см. Рисунок 3); если параметр не задан, то доступны оба варианта на выбор пользователя (см. Рисунок 188). При входе в режиме «Войти через SSO» не требуется предварительно создавать администратору учетные записи в меню «Администраторы».

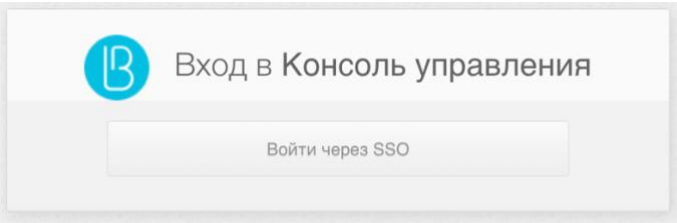

Рисунок 187 – Окно входа в консоль при включенном режиме SSO

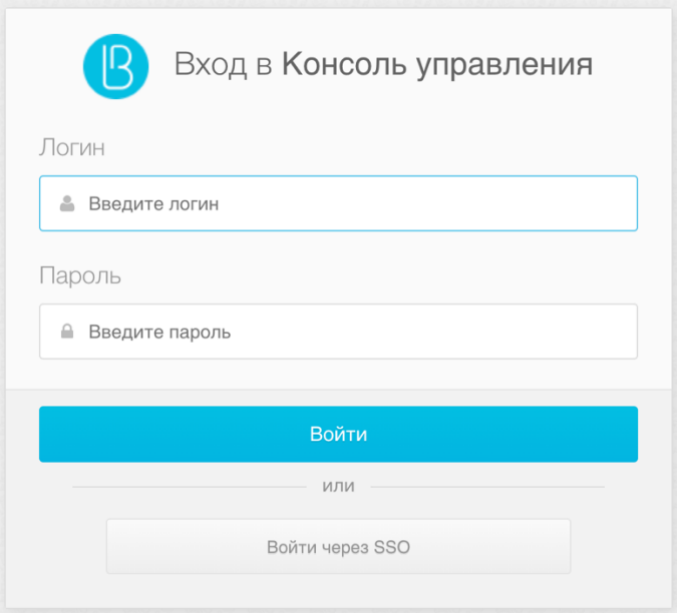

Рисунок 188 – Окно входа в консоль при всех включенных режимах входа

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.

16.3.2. 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 (см. п. 6.2.6).

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,
  …
}
```

16.3.3. Настройка ролей и прав доступа в консоль управления

Стандартные роли администраторов описаны в п. 2.1.12. 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 (см. п. 16.2). 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:

Таблица 13

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)

17. Мониторинг функционирования приложений

17.1. 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[78].

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
# 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="bl
itz.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",s
tatus="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
```

[78] См.: https://prometheus.io/

```
…
blitz idp authn method latency bucket{blitz host="papp01.loc",method="sms",le="+Inf",} 2.0
blitz idp authn method latency count{blitz host="papp01.loc",method="sms",} 2.0
blitz_idp_authn_method_latency_sum{blitz_host="papp01.loc",method="sms",} 28.686999999999998
blitz_idp_authn_method_latency_bucket{blitz_host="papp01.loc",method="password",le="1.0",} 0.0
…
blitz idp authn method latency bucket{blitz host="papp01.loc",method="password",le="+Inf",} 4.0
blitz idp authn method latency count{blitz host="papp01.loc",method="password",} 4.0
blitz_idp_authn_method_latency_sum{blitz_host="papp01.loc",method="password",} 1835.901
# HELP blitz_idp_datasource_connections Amount connections to datasource
# TYPE blitz_idp_datasource_connections gauge
blitz_idp_datasource_connections{blitz_host="papp01.loc",ds_type="ldap",ds_name="389-ds",} 10.0
# HELP blitz_idp_version Application version
# TYPE blitz idp version gauge
blitz idp version{blitz host="papp01.loc",part="major",} 5.0
blitz_idp_version{blitz_host="papp01.loc",part="minor",} 16.0
blitz_idp_version{blitz_host="papp01.loc",part="patch",} 1.0
# HELP blitz idp notify user total Amount of user notifications by channel
# TYPE blitz idp notify user total counter
blitz_idp_notify_user_total{blitz_host="papp01.loc",channel="email",} 3.0
blitz_idp_notify_user_total{blitz_host="papp01.loc",channel="sms",} 4.0
blitz_idp_notify_user_total{blitz_host="papp01.loc",channel="push",} 2.0
```

Имя каждой метрики начинается с имени приложения (дефис в имени заменен на подчеркивание): blitz_idp_%%%, blitz_registration_%%%, blitz_recovery_%%%, blitz_console_%%%.

Список доступных метрик приведен в таблице:

Таблица 14

Метрики функционирования Blitz Identity Provider

Право доступа	**Тип**	**Пояснение**
uptime_seconds	gauge	Время с момента запуска приложения (в секундах)
licence_exp_seconds	gauge	Время до истечения срока действия лицензии в секундах)
config_mtime	gauge	Время последнего изменения конфига
datasource_latency	histogram	Задержки ответа от хранилища УЗ по операциям чтения и записи (могут быть типы ldap, jdbc, couch)
mq_connections	gauge	Количество коннектов к MQ (rmq, kafka)
mq_latency	histogram	Задержки ответа от MQ (rmq, kafka)
authn_method_app_total	counter	Количество успешных и неуспешных аутентификаций каждым методом входа в различные приложения
authn_method_total	counter	Общее кол-во успешных и нет аутентификаций разными методами
authn_method_latency	histogram	Длительность аутентификации по разным методам входа
datasource_connections	gauge	Кол-во коннектов к хранилищам
version	gauge	Версия приложения
notify_user_total	counter	Кол-во направленных сообщений по разным каналам
authn_method_app_created	служебные	Эти метрики (с суффиксом _created) генерируются в связи с особенностями Prometheus и содержат время в unix timestamp момента создания метрики.
authn_method_created		
authn_method_latency_created		
datasource_latency_created		
mq_latency_created		

notify_user_created		

17.2. 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).

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.
2) Модифицируйте шаблон дашборда blitz-dashboard.json. Настройте Grafana и добавьте[79] дашборд.

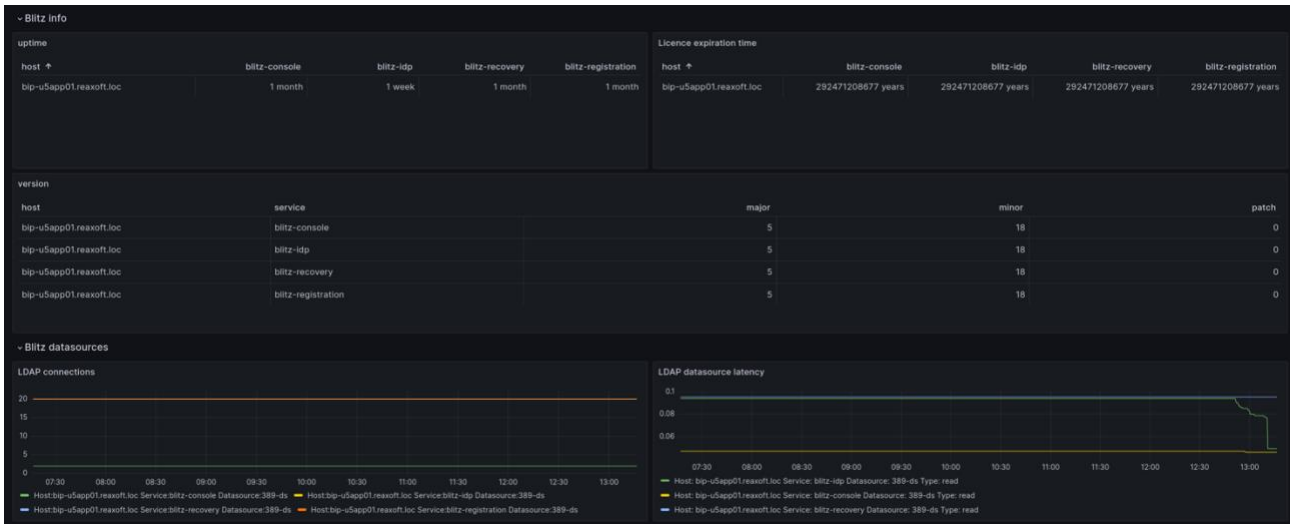

Рисунок 189 Пример визуализации данных в Grafana

[79] https://prometheus.io/docs/visualization/grafana/

Рисунок 190 Пример визуализации данных в Grafana

18. 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 /usr/share/identityblitz/blitz-config/blitz.conf в блоке logger изменить уровни логирования. По умолчанию выставлены следующие уровни логирования:

```
"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: 0:0:0:0:0:0:0:1
# user agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
################################################################################################
```

Приложение 1. Функциональная спецификация Blitz Identity Provider

Группа функций	Функция	Доступность функции в редакции		
		Standard	Enterprise	Ultimate
Технологии единого входа				
OpenID Connect и OAuth 2.0	RFC 6749 "The OAuth 2.0 Authorization Framework"	да	да	да
	OpenID Connect Core 1.0	да	да	да
	Передача атрибутов пользователя в составе id_token/access_token в JSON Web Token (JWT)	да	да	да
	Конфигурируемый REST-сервис UserInfo, настройка возвращаемых атрибутов в зависимости от scope	да	да	да
	RFC 7636 "Proof Key for Code Exchange by OAuth Public Clients"	да	да	да
	RFC 7662 "OAuth 2.0 Token Introspection"	да	да	да
	RFC 7591 "OAuth 2.0 Dynamic Client Registration Protocol"	нет	да	да
	RFC 7592 "OAuth 2.0 Dynamic Client Registration Management Protocol"	нет	да	да
	RFC 8252 "OAuth 2.0 for Native Apps"	нет	да	да
	RFC 8414 "OAuth 2.0 Authorization Server Metadata"	да	да	да
	OpenID Connect RP-Initiated Logout 1.0	да	да	да
	OpenID Connect Front-Channel Logout 1.0	нет	да	да
	OpenID Connect Back-Channel Logout 1.0	нет	да	да
	RFC 8693 "OAuth 2.0 Token Exchange"	нет	да	да
SAML	SAML Web Browser SSO Profile	да	да	да
	SAML Single Logout Profile	да	да	да
RADIUS	RFC 2865 "Remote Authentication Dial In User Service (RADIUS)"	нет	да	да
WS-Federation	WS-Federation (для подключения Microsoft-приложений)	да	да	да
Proxy SSO	Подключения веб-приложений, получающих состояние сессии из HTTP-заголовков и cookies	нет	нет	да
	Поддержка возможности заполнения за пользователя логина/пароля от учетной записи в размещенное за proxy веб-приложение, не поддерживающее стандартным образом подключения к SSO	нет	нет	да
Другое	Единый вход работает между приложениями, которые подключены к IDP с использованием любых поддерживаемых технологий (например, SSO между OpenID Connect и SAML-приложениями)	да	да	да
	Поддержка SSO-входа с использованием Kerberos SSO	да	да	да
	Поддержка единого SSO с приложениями IBM, использующими для единого входа Ltpa2Token	нет	нет	да
Идентификация и аутентификация пользователей				
Вход по логину и паролю	Проверка логина и пароля при аутентификации	да	да	да
	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)	да	да	да
	Запоминание логина, если пользователь ранее уже входил с этого устройства	да	да	да
	Запоминание на устройстве нескольких пользователей. Возможность сменить текущую учетную запись пользователя без необходимости логаута	нет	опция	да
	Обработка события «пароль требует смены» при входе. Возможности сменить пароль в момент входа	да	да	да
	Проверка соответствия пароля действующей парольной политике при входе. Рекомендация сменить пароль	да	да	да
	Встроенная защита от подбора пароля (перебор паролей на одну учетную запись) и подбора логина (попытка подбора	нет	да	да

	пароля на набор учетных записей):			
	– проверка CAPTCHA (reCAPTCHA или иной сервис, выбранный Заказчиком);	нет	да	да
	– временное блокирование входа по паролю учетной записи при выявленных попытках перебора	нет	да	да
	– замедление входа пользователя (задержка входа, решение браузером вычислительно сложной задачи – Proof of Work)	нет	да	да
	Предупреждение пользователя о попытке входа с паролем, который был недавно изменен	да	да	да
Вход на основе сеанса ОС	Идентификация пользователя на основе результата входа в домен (Kerberos)	да	да	да
	Возможность подключения системы входа одновременно к нескольким доменам и обеспечения сквозного входа пользователей из разных доменов	нет	да	да
	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	нет	да	да
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	Apple ID, Facebook, Google, Mail ID, VK ID, Одноклассники, Яндекс		
	Вход через ЕСИА в режиме физического лица	нет	да	да
	Вход через ЕСИА в режиме представителя организации (с выбором организации при входе)	нет	да	да
	Вход через ЕСИА в режиме цифрового профиля	нет	опция	да
	Вход через банки: Сбер ID, Tinkoff ID, ВТБ ID, СберБизнес ID, Альфа ID	нет	опция	да
	Вход через Mos ID (СУДИР)	нет	да	да
	Вход через внешний поставщик идентификации с поддержкой OIDC	нет	да	да
	Вход через внешний поставщик идентификации с поддержкой SAML	нет	да	да
	Вход через СУДИС	нет	опция	да
	Сопоставление/регистрация учетной записи в процессе первичного входа через социальную сеть	да	да	да
	Возможность привязки к одной учетной записи пользователя нескольких учетных записей внешних поставщиков	да	да	да
	Возможность привязки к одной учетной записи внешнего поставщика нескольких учетных записей пользователей	да	да	да
	Возможность программирования собственного алгоритма привязки учетных записей и сопоставления атрибутов	нет	да	да
Вход на основе запомненного устройства	Автоматическая идентификация пользователя, если он уже входил с этого устройства и согласился запомнить свой вход	да	да	да
	Возможность пользователю отследить, на каких устройствах запомнен вход, и выйти с этих устройств	да	да	да
	Автоматический выход с запомненных устройств при смене/восстановлении пароля пользователем	да	да	да
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.	нет	да	да
Вход с помощью WebAuthn, Passkey, FIDO2	Вход с помощью платформо-независимых ключей безопасности FIDO2	нет	да	да
	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	нет	да	да
Вход с помощью смарт-карты / USB-ключа	Вход с помощью средств квалифицированной электронной подписи	нет	да	да
	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	нет	да	да
	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.	нет	да	да
	Поддерживаемые браузеры: Chrome, Firefox, Yandex	нет	да	да
	Возможность сопоставления/регистрации учетных записей в процессе первичного входа на основе данных из сертификата	нет	да	да

	квалифицированной электронной подписи			
	Возможность проверки действительности подписи и сертификата встроенными возможностями ПО	нет	да	да
	Возможность проверки действительности подписи и сертификата через вызов внешнего сервиса проверки	нет	да	да
Двухфакторная аутентификация	Подтверждение входа разовым паролем из SMS (SMS-шлюз предоставляет Заказчик)	да	да	да
	Подтверждение входа разовым паролем из email	да	да	да
	Подтверждение входа разовым паролем TOTP-приложения (RFC 6238 "TOTP: Time-Based One-Time Password Algorithm")	да	да	да
	Подтверждение входа разовым паролем из аппаратного брелока. Поддержка брелоков HOTP (RFC 4226 "HOTP: An HMAC-Based One-Time Password Algorithm"). Брелоки предоставляет Заказчик.	нет	да	да
	Подтверждение входа ключом безопасности WebAuthn, Passkey, FIDO2	нет	да	да
	Подтверждение входа ключом безопасности U2F	нет	да	да
	Подтверждения входа разовым паролем в push-уведомлении в мобильном приложение Заказчика (сервис для отправки push и мобильное приложение предоставляет Заказчик)	нет	нет	да
	Подтверждение по входящему звонку	да	да	да
Другое	Возможность Заказчику самостоятельно добавить собственный метод аутентификации	нет	да	да
	Возможность Заказчику самостоятельно настроить внешний вид страницы входа раздельно для каждого приложения, в которое осуществляется вход	да	да	да
	Предоставление API, позволяющее мобильным приложениям зарегистрировать событие входа и получить маркеры безопасности при входах с использованием ПИН-кода, Touch ID, Face ID	нет	да	да
	Блокирование учетных записей в случае длительной неактивности	нет	да	да
	Запрет на повторное использование идентификатора удаленной учетной записи в течение установленного времени	нет	да	да
	Возможность кастомизации операций с хранилищами данных	нет	да	да
Логаут				
	Завершение пользовательской сессии при инициировании логаута пользователем	да	да	да
	Завершение пользовательской сессии при смене пароля пользователя в другой сессии или при сбросе/восстановлении пароля пользователю	да	да	да
	Ограничение допустимых ссылок для возврата в приложение после успешного логаута	да	да	да
	Информирование приложений о произведенном едином логауте через браузер (front channel)	нет	да	да
	Информирование приложений о произведенном едином логауте через сервер (back channel)	нет	да	да
Контроль доступа				
	Проверка правил доступа при входе пользователя в приложения. Проверка наличия у пользователя прав доступа, членства в группах пользователей, наличия атрибутов с требуемыми значениями	да	да	да
	Проверка правил доступа при вызове приложениями защищаемых REST-сервисов через шлюз безопасности Blitz Keeper (API Security Gateway)	нет	опция	да
Возможности пользователя по управлению своей учетной записью				
Регистрация	Настраиваемое веб-приложение самостоятельной регистрации пользователей. Можно настроить набор атрибутов, заполняемых пользователем при регистрации, требования к подтверждению email/телефона, настроить внешний вид страницы регистрации, вызов сервисов проверки Заказчика	да	да	да

	Можно задать различные настройки веб-приложения самостоятельной регистрации пользователя для различных сценариев вызова регистрации	нет	да	да
	Возможность вызова внешнего приложения регистрации с передачей ему контекста входа и сведений, полученных из внешнего поставщика (ЕСИА, Mos ID, соц.сети, банки) в процессе входа	нет	да	да
	По результатам успешной регистрации пользователь автоматически входит в приложение, при попытке входа в которое изначально была инициирована регистрация	да	да	да
	Проверка CAPTCHA (reCAPTCHA или иной сервис, выбранный Заказчиком)	да	да	да
Настройки безопасности учетной записи	Веб-приложение, позволяющее пользователю управлять настройками безопасности его учетной записи:	да	да	да
	– возможность самостоятельно сменить пароль	да	да	да
	– 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	да	да	да
	– возможность настроить двухфакторную аутентификацию для своей учетной записи	да	да	да
	– возможность посмотреть/отредактировать список запомненных устройств, привязанных учетных записей внешних поставщиков входа	да	да	да
	– возможность посмотреть события безопасности со своей учетной записью	да	да	да
	предоставление API для возможности встраивания всех вышеперечисленных функций управления настройками безопасности учетной записи в стороннее веб-приложение	нет	да	да
Восстановление забытого пароля	Веб-приложения, позволяющего восстановить забытый пароль, с подтверждением email или телефона	да	да	да
	Дополнительные проверки при восстановлении пароля от учетной записи, для которой включена двухфакторная аутентификация	да	да	да
	По результатам успешного восстановления пароля пользователь автоматически входит в приложение, при попытке входа в которое изначально была инициирована процедура восстановления	да	да	да
	Проверка CAPTCHA (reCAPTCHA или иной сервис, выбранный Заказчиком)	да	да	да
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)	нет	да	да
	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)	нет	да	да
	Возможность в момент входа выпустить Passkey (настроить вход по Face ID / Touch ID)	нет	да	да
	Возможность показать пользователю объявление	нет	да	да
	Возможность запросить у пользователя согласие	нет	да	да
	Возможность запросить у пользователя заполнение текстового атрибута	нет	да	да
	Возможность запросить у пользователя выбор значения из списка значений	нет	да	да
	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)	нет	да	да
Парольные политики	Проверка пароля на соответствие парольной политике: минимальная длина, требования к алфавиту, запрет словарных паролей, запрет повтора паролей, проверка срока действия паролей	да	да	да
Мониторинг и аудит				
Оповещения пользователей о	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	да	да	да

событиях безопасности	социальной сети, включение/выключение двухфакторной аутентификации			
	Возможность настроить набор событий оповещения и тексты оповещений для SMS и для email	да	да	да
Регистрация событий безопасности	Регистрация успешных и неуспешных событий безопасности с учетной записью: события входа, регистрации, изменения настроек безопасности, восстановления пароля. Должны регистрироваться как действия, инициированные пользователем, так и действия, инициированные администратором	да	да	да
	Сопоставление IP-адресам геоданных в событиях и уведомлениях (БД в формате mmdb с геоданными предоставляет Заказчик)	нет	да	да
	Интерфейс администратора для поиска/просмотра событий безопасности	да	да	да
	Запись событий безопасности: в БД, в лог-файл, в Kafka	да	да	да
Мониторинг	Возможность в момент входа пользователя вызывать системы сбора метрик и статистики, антифрод системы	нет	да	да
	Возможность осуществлять мониторинг компонент из внешней системы мониторинга (Zabbix и аналоги). Предоставление метрик в формате Prometheus	нет	да	да
	Шаблоны job-задания Prometheus и дашборда Grafana в поставке	да	да	да
Очереди	Возможность передавать в очередь RabbitMQ события, связанные с учетными записями пользователей и групп доступа	нет	да	да
	Возможность передавать в очередь Kafka события безопасности	нет	да	да
Администрирование				
	Веб-приложение администрирования:	да	да	да
	– задание настроек подключенных приложений (параметры приложений, разрешенные режимы взаимодействия, правила контроля доступа)	да	да	да
	– настройка атрибутов пользователей и сопоставление атрибутов хранилищам учетных записей	да	да	да
	– настройка подключения к хранилищам учетных записей на основе LDAP	да	да	да
	– настройка подключения к произвольным хранилищам (через предоставленный Заказчиком сервис)	нет	да	да
	– поддержка работы одновременно с несколькими хранилищами учетных записей	нет	да	да
	– настройка методов идентификации/аутентификации и внешних поставщиков входа	да	да	да
	– настройка подключения к SMTP-службе и к SMS-шлюзу	да	да	да
	– поддержка ролевого доступа для входа в веб-приложение администратора. Возможность для разных пользователей задать разный набор доступных действий	нет	да	да
	– управление настройками веб-приложений регистрации, управления настройками безопасности, восстановления пароля	да	да	да
	– 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)	да	да	да
	– администрирование групп пользователей, управления членством пользователей в группах	нет	да	да
	– настройка внешнего вида страниц входа в приложения	да	да	да
	– просмотр и фильтрация зарегистрированных событий безопасности	да	да	да
	– возможность входа в веб-приложение администрирования через SSO	нет	да	да
	Интерфейс администратора на русском и английском языках	да	да	да
	Возможность добавления переводов на дополнительные языки	да	да	да

Приложение 2. Рекомендации по обеспечению мер защиты информации согласно требованиям ФСТЭК

Условное обозначение и номер меры	Мера защиты информации в информационных системах	Рекомендации по настройке
Идентификация и аутентификация субъектов доступа и объектов доступа (ИАФ)		
ИАФ.1	Идентификация и аутентификация пользователей, являющихся работниками оператора	Настроить методы аутентификации пользователей, см. п. 4. Для администраторов консоли управления настроить вход через Blitz (см. п. 16.3.1). Настроить через процедуры входа для администраторов и пользователей требования к прохождению двухфакторной аутентификации (см. п. 6) Задать для подключаемых приложений client_id и client_secret (см. п. 5)
ИАФ.3	Управление идентификаторами, в том числе создание, присвоение, уничтожение идентификаторов	Задать атрибут, который будет использоваться в качестве идентификатора учетной записи (см. п. 3.1.5). Настроить запрет на повторное использование идентификатора после удаление учетной записи (см. 16.1.15) и блокирование неактивных учетных записей (см. 16.1.14)
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	Установить парольную политику (см. п. 4.1). Управлять учетными записями пользователей (см. п. 9).
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)	Настроить вход через внешние поставщики идентификации (см. п. 8.9, п. 8.12, п. 8.17)
Управление доступом субъектов доступа к объектам доступа		
УПД.1	Управление (заведение, активация, блокирование и уничтожение) учетными записями пользователей, в том числе внешних пользователей	При необходимости разделения учетных записей на внешние и внутренние завести атрибут с типом учетной записи (см. п. 3) и настроить политику доступа пользователей в приложения (см. п. 6). Для временных учетных записей настроить правила блокирования входа по истечение срока действия записей (см. п 6.2.4). Для использования функций работы с группами пользователей настроить группы пользователей (см. п. 16.1.16). Управлять учетными записями через консоль управления (см. п. 9)
УПД.2	Реализация необходимых методов (дискреционный, мандатный, ролевой или иной метод), типов (чтение, запись, выполнение или иной тип) и правил разграничения доступа	Настроить права доступа (см. п. 11), разрешения (см. п. 5.3.2), установить разрешения приложений (см. п. 5.3.1), настроить шлюз безопасности (см. п. 15), настроить атрибуты пользователей (см. п. 3) и группы пользователей (см. п. 16.1.16), процедуры входа в приложения (см. п. 6).

УПД.4	Разделение полномочий (ролей) пользователей, администраторов и лиц, обеспечивающих функционирование информационной системы	Использовать сервисы изменения атрибутов, включения и исключения пользователей в группы, назначения и отзыва прав доступа (см. «Руководство по интеграции»)
УПД.5	Назначение минимально необходимых прав и привилегий пользователям, администраторам и лицам, обеспечивающим функционирование информационной системы	Назначать роли администраторов (см. п. 2.1.12), управлять атрибутами пользователей (см. п. 9), назначать права доступа с использованием сервисов (см. «Руководство по интеграции»)
УПД.6	Ограничение неуспешных попыток входа в информационную систему (доступа к информационной системе)	Настроить политику ограничения числа попыток входа с последующим блокированием учетной записи (см. п. 4.4)
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	Настроить для приложений экран согласия пользователя (см. п. 5.3.1, п. 5.3.2, п. 16.2)
УПД.8	Оповещение пользователя после успешного входа в информационную систему о его предыдущем входе в информационную систему	Настроить доступ к аудиту по себе для пользователей (см. п. 7.3.2)
УПД.9	Ограничение числа параллельных сеансов доступа для каждой учетной записи пользователя информационной системы	Настроить политику ограничения числа параллельных сеансов (см. п. 16.3.2, п. 6.2.6)
УПД.10	Блокирование сеанса доступа в информационную систему после установленного времени бездействия (неактивности) пользователя или по его запросу	Настроить период неактивности (см. п. 4)
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	Просмотр событий безопасности периодически осуществлять в консоли управления (см. п. 11)
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	Просматривать журналы событий на предмет возникновения ошибок (см. п. 17)
РСБ.5	Мониторинг (просмотр, анализ) результатов регистрации событий безопасности и реагирование на них	Просмотр событий безопасности периодически осуществлять в консоли управления (см. п. 11)
РСБ.6	Генерирование временных меток и (или) синхронизация системного времени в информационной системе	При установке ПО сконфигурировать использование сервиса точного времени (NTP)
РСБ.7	Защита информации о событиях безопасности	Настроить резервное копирование СУБД (см. п. 2.1.3)
РСБ.8	Обеспечение возможности просмотра и анализа информации о действиях отдельных пользователей в информационной системе	Просмотр событий безопасности периодически осуществлять в консоли управления (см. п. 11)

Приложение 3. Методика проверки основных параметров и характеристик ПО

№ пункта ТУ	Требование технических условий	Методика проверки
2.1.1.1	ПО «Blitz Identity Provider» должно осуществлять идентификацию и аутентификацию пользователей и администраторов ПО «Blitz Identity Provider». К внутренним пользователям относятся пользователи и администраторы, выполняющие свои функции в соответствии с инструкциями и которым в ПО «Blitz Identity Provider» присвоены учетные записи. Аутентификация осуществляется с использованием паролей, аппаратных средств, одноразовых паролей, или в случае многофакторной (двухфакторной) аутентификации – определенной комбинации указанных средств. В ПО «Blitz Identity Provider» должна обеспечиваться многофакторная (двухфакторная) аутентификация для удаленного доступа с правами администраторов и с правами пользователей с использованием сети связи общего пользования, в том числе сети Интернет, а также для локального доступа с правами администраторов и с правами пользователей. Идентификация осуществляется по идентификатору (имени учетной записи), связанному с учётной записью пользователя или администратора ПО «Blitz Identity Provider». ПО «Blitz должно осуществлять идентификацию и аутентификацию приложения-клиента. Идентификация приложений-клиентов в ПО «Blitz Identity Provider» обеспечивается по логическим именам (client_id). Аутентификация приложений-клиентов в ПО «Blitz Identity Provider» обеспечивается с использованием протокола OAuth 2.0 с помощью проверки присвоенного приложению-клиенту пароля (client_secret) (мера защиты ИАФ.1).	Проверить, что идентификация и аутентификация пользователей работает в соответствии с заданными (см. п. 4) в системе настройками. Выполнять проверки в части используемых в конкретной установке методами идентификации и аутентификации. Проверить, что для администраторов консоли управления настроен вход через Blitz (см. п. 16.3.1) и настроено прохождение двухфакторной аутентификации при удаленном доступе (см. п. 6). Проверить, что работает функция задания для подключаемых приложений client_id и client_secret (см. п. 5)
2.1.1.2	ПО «Blitz Identity Provider» должно обеспечивать функции формирования администратором идентификатора, который однозначно идентифицирует пользователя и (или) приложение-клиента, присвоение идентификатора пользователю и (или) приложению-клиенту, предотвращение повторного использования идентификатора пользователя в течение установленного администратором периода времени, блокирование идентификатора пользователя после установленного оператором времени неиспользования. (мера защиты ИАФ.3).	Проверить, что настроены запрет на повторное использование идентификатора после удаление учетной записи (см. п. 16.1.15) и блокирование неактивных учетных записей (см. п. 16.1.14). Проверить, что невозможно создать учетную запись с идентификатором, который ранее использовался – для этого попробовать создать учетную запись через консоль управления (см. п. 9.2).
2.1.1.3	ПО «Blitz Identity Provider» должно осуществлять возможность изменения администратором аутентификационной информации (средств аутентификации), заданной их изготовителями и используемой при внедрении ПО «Blitz Identity Provider», генерации средств аутентификации пользователям, установления характеристик пароля: – задание минимальной сложности пароля с определяемыми администратором требованиями к регистру, количеству символов, сочетанию букв верхнего и нижнего регистра, цифр и специальных символов; – задание минимального количества измененных символов при создании новых паролей; – задание максимального времени действия пароля (срок, не позже которого пользователь должен сменить пароль); – задание минимального времени действия пароля (срок, в течение которого данный пароль не должен быть использован повторно); – запрет на использование пользователями определенного администратором числа последних использованных паролей при создании новых паролей. ПО «Blitz Identity Provider» должно обеспечивать: – блокирование (прекращение действия) и замену утерянных, скомпрометированных или поврежденных средств аутентификации:	Проверить, что нельзя установить пользователю пароль (см. п. 9), который не соответствует настроенной парольной политике (см. п. 4.1).

	– для паролей – сброс пароля админом, самостоятельное восстановление (замена) утерянного пароля пользователем; – для HOTP и TOTP генераторов одноразовых паролей – удаление администратором привязанного к пользователю прежнего генератора, привязка нового генератора; – для SMS-кодов и email-кодов – замена в профиле пользователя номера мобильного телефона и/или адреса email на актуальные; – для входа с помощью внешних систем (ЕСИА и др.) – удаление администратором или пользователем связи учетной записи пользователя с учетной записью во внешней системе входа. – назначение необходимых характеристик средств аутентификации (в том числе механизма паролей), обновление аутентификационной информации (замена средств аутентификации); – защиту аутентификационной информации от неправомерных доступа к ней и модифицирования; – единую аутентификацию при доступе пользователей в подключенные к ПО «Blitz Identity Provider» приложения-клиенты (мера защиты ИАФ.4).	
2.1.1.4	ПО «Blitz Identity Provider» должно осуществлять защиту обратной связи при вводе аутентификационной информации. Защита обратной связи в процессе аутентификации должна обеспечиваться исключением отображения действительного значения аутентификационной информации. Вводимые символы пароля должны отображаться условными знаками «•» (мера защиты ИАФ.5).	Проверить, что во всех полях, предназначенных для ввода пароля (страница входа, страницы настроек в консоли управления), при вводе пароля отображаются символы маскирования.
2.1.1.5	ПО «Blitz Identity Provider» должно осуществлять однозначную идентификацию и аутентификацию пользователей, не являющихся администраторами (внешних пользователей). К внешним пользователям, относятся все пользователи информационной системы, не указанные в ИАФ.1 в качестве внутренних пользователей. ПО «Blitz Identity Provider» должно обеспечивать однозначную идентификацию и аутентификацию пользователей для всех видов доступа. ПО «Blitz Identity Provider» должно осуществлять идентификацию и аутентификацию внешних пользователей в том числе следующими способами: – с использованием единой системы идентификации и аутентификации, созданной в соответствии с постановлением Правительства Российской Федерации от 28 ноября 2011 г. № 977. – с использованием системы управления доступом к информационным ресурсам и системам города Москвы. – с использованием внешних систем входа сторонних организаций, построенных с использованием ПО «Blitz Identity Provider» (мера защиты ИАФ.6).	Проверить, что работает вход через настроенные внешние поставщики идентификации (см. п. 8.9, п. 8.12, п. 8.17). Для этого на странице входа нажать кнопку входа через соответствующий внешний поставщик и ввести данные от учетной записи внешнего поставщика.
2.1.2.1	ПО «Blitz Identity Provider» должно обеспечивать следующие функции управления учетными записями пользователей, в том числе внешних пользователей: – определение типа учетной записи (внутреннего пользователя, внешнего пользователя, временная); – объединение учетных записей в группы (при необходимости); – подтверждение верификации пользователя (факта проверки личности пользователя) при заведении учетной записи пользователя; – заведение, активация, блокирование и уничтожение учетных записей пользователей; – изменение учетных записей пользователей; – блокировка временных учетных записей пользователей, предоставленных для однократного (ограниченного по времени) выполнения задач в информационной системе. ПО «Blitz Identity Provider» должно обеспечить поддержку управления учетными записями пользователей администратором. Должно обеспечиваться автоматическое блокирование временных учетных записей по окончании установленного периода времени для их использования (мера защиты	Проверить в консоли управления (см. п. 9), что по атрибутам пользователей можно различать внешние, внутренние и временные учетные записи. Для различия используются настроенные атрибуты (см. п. 3). Проверить вход с использованием тестовых учетных записей различного типа в приложения – убедиться, что вход работает в соответствии с заданными настройками политики доступа пользователей в приложения (см. п. 6). Проверить, что для временных учетных записей настроены правила блокирования входа по истечение срока действия записей (см. п 6.2.4). Для тестовой временной учетной записи установить истекший срок действия (см. п. 9), и убедиться, что вход такой учетной записью в приложения станет

	УПД.1).	невозможным. При использовании функций работы с группами пользователей проверить, что для включенных в группы учетных записей отображается их членство в группах (см. п. 9).
2.1.2.2	ПО «Blitz Identity Provider» должно осуществлять: – дискреционный метод управления доступом субъектов доступа к объектам доступа: – Субъектами доступа выступают пользователи, идентифицируемые с использованием идентификатора пользователя и идентификатора группы/организации, в которую включен пользователь. Объектами доступа выступают защищаемые с использованием подсистемы шлюза безопасности (blitz_keeper) сервисы, идентифицируемые своими URL и HTTP-методами (GET/POST/PUT/DELETE). Приложения запрашивают в Blitz Identity Provider идентификацию/аутентификацию пользователя, и получают в результате маркер доступа (access_token). С использованием маркера доступа приложения вызывают защищаемые сервисы. Вызов осуществляется через подсистему шлюза безопасности (blitz_keeper). Подсистема шлюза безопасности осуществляет проверку настроенных в Blitz Identity Provider правил доступа, описывающих, каким субъектам доступа (пользователям) какие объекты доступа (сервисы) разрешено вызывать. Если нет разрешений на вызов сервиса, то подсистема шлюза безопасности возвращает приложению ошибку доступа. – Субъектами доступа выступают приложения, идентифицируемые с использованием идентификатора приложения (client_id), присвоенного приложению при регистрации в Blitz Identity Provider и аутентифицируемые с помощью секрета приложения (client_secret). Объектами доступа выступают защищаемые с использованием подсистемы шлюза безопасности (blitz_keeper) сервисы, идентифицируемые своими URL и HTTP-методами (GET/POST/PUT/DELETE). ПО «Blitz Identity Provider» позволяет определить произвольный набор типов доступа (разрешений scope), позволяющих выразить операции с защищаемым сервисом, разрешенные приложению. Приложения запрашивают в Blitz Identity Provider маркер доступа (access_token) с определенным набором разрешений. С использованием маркера доступа приложения вызывают защищаемые сервисы. Вызов осуществляется через подсистему шлюза безопасности (blitz_keeper). Подсистема шлюза безопасности осуществляет проверку заданных в Blitz Identity Provider правил доступа, описывающих, каким субъектам доступа (приложениям) какие объекты доступа (сервисы) разрешено вызывать. Если нет разрешений на вызов сервиса или маркер доступа получен без необходимого набора разрешений (scope), то подсистема шлюза безопасности возвращает приложению ошибку доступа. – Субъектами доступа выступают приложения, идентифицируемые с использованием идентификатора приложения (client_id), присвоенного приложению при регистрации в Blitz Identity Provider и аутентифицируемые с помощью секрета приложения (client_secret). Объектами доступа выступают учетные записи пользователей (совокупность атрибутов пользователя). Приложения запрашивают в Blitz Identity Provider идентификацию/аутентификацию пользователя и определенный набор разрешений (scope), и получают в результате маркер доступа (access_token). Затем приложения запрашивают в Blitz Identity Provider получение данных о пользователе и получают только тот набор данных, который соответствует заданным в Blitz Identity Provider правилам доступа (перечню разрешенных приложению разрешений (scope) и связанным с разрешениями атрибутам пользователя). – ролевой метод управления доступом, предусматривающий управление доступом субъектов доступа	Проверить на тестовых учетных записях, что вход в приложения осуществляется в соответствии с заданными настройками прав доступа (см. п. 15.2). Модифицируя через браузер запрос приложения на получение кода авторизации к серверу проверить, что если в запрос добавить неразрешенное (см. п. 5.3.1) приложению разрешение (см. п. 5.3.2), то Blitz вернет приложению ошибку. Проверить, что вызов защищаемых шлюзом безопасности сервисов (см. п. 15) проходит успешно только при соответствии запроса установленным правилам доступа. Проверить, что вход пользователей в приложения работает в соответствии с настроенными при внедрении правилами ограничения доступа, основанными на проверке атрибутов пользователей (см. п. 3) или членства в группах пользователей (см. п. 16.1.16) или иных заданных в процедуре входа в приложения условиях (см. п. 6).

	к объектам доступа на основе ролей субъектов доступа. Предусмотрены следующие возможности назначения пользователем ролей и управления доступом на основе ролей: – В качестве ролей можно использовать группы доступа. Blitz Identity Provider предоставляет программные интерфейсы для создания/изменения/удаления групп доступа, включения/исключения пользователей в группы доступа. В настраиваемых в Blitz Identity Provider правилах доступа пользователей к защищаемым сервисам можно настраивать разрешение на доступ к сервисам только пользователям, включенным в определенные группы доступа. – В качестве ролей можно использовать значение выбранного атрибута учетной записи пользователя. Blitz Identity Provider позволяет настроить атрибут (например, role) и описать правила доступа пользователя к защищаемому сервису на основании проверки наличия у пользователя атрибута (role) с определенным значением; – В качестве ролей можно использовать назначаемые пользователям полномочия (rights). Blitz Identity Provider позволяет определить список полномочий (rights), предоставляет сервисы назначения/отзыва пользователям полномочий относительно приложения (отражает наличие у пользователя роли в приложении) или группы пользователей (отражает наличие у пользователя роли в организации, если группа пользователей используется для описания организаций). Правила доступа пользователей к защищаемым сервисам могут быть описаны так, чтобы осуществлять проверку наличия у пользователей определенных полномочий (ролей) для возможности вызова указанных сервисов. ПО «Blitz Identity Provider» должно обеспечивать: – управление доступом субъектов (пользователей) при входе в приложения-клиенты; – управление доступом субъектов (пользователей и приложений-клиентов) к защищаемым с использованием подсистемы шлюза безопасности сервисам (предоставляемым защищаемыми приложениями API) (мера защиты УПД.2).	
2.1.2.3	ПО «Blitz Identity Provider» должно предоставлять сервисы для назначения, хранения, отзыва пользователям полномочий (ролей). Доступ к объектам доступа с учетом разделения полномочий (ролей) обеспечивается в соответствии с УПД.2 (мера защиты УПД.4).	Проверить работоспособность сервисов изменения атрибутов, включения и исключения пользователей в группы, назначения и отзыва прав доступа (см. «Руководство по интеграции»)
2.1.2.4	ПО «Blitz Identity Provider» должно обеспечивать назначение прав и привилегий пользователям (через редактирование их атрибутов или через использование сервисов назначения/отзыва полномочий rights)), приложениям (через определение разрешений приложений – матрицы разрешений scope), и администраторам. Доступ к объектам доступа с учетом минимально необходимых прав и привилегий обеспечивается в соответствии с УПД.2 (мера защиты УПД.5).	Проверить работоспособность функции назначения роли администратору (см. п. 2.1.12), управления атрибутами пользователей (см. п. 9), назначение права доступа с использованием сервисов (см. «Руководство по интеграции»)
2.1.2.5	ПО «Blitz Identity Provider» должно ограничивать количество неуспешных попыток входа в приложения-клиенты за период времени, установленный администратором, а также обеспечивать блокирование учетной записи пользователя при превышении пользователем ограничения количества неуспешных попыток входа. Ограничение количества неуспешных попыток входа должно обеспечиваться в соответствии с ИАФ.4 ПО «Blitz Identity Provider» должно обеспечивать автоматическое блокирование учетной записи пользователя при превышении пользователем ограничения количества неуспешных попыток входа за установленный период времени с возможностью разблокирования только администратором или самим пользователем при прохождении им процедуры восстановления пароля (мера защиты УПД.6).	Проверить, что при последовательных попытках ввода неправильного пароля учетная запись временно блокируется в соответствии с заданными настройками (см. п. 4.4)
2.1.2.6	ПО «Blitz Identity Provider» должно обеспечивать возможность отображения пользователю в виде сообщения («окна») при его входе в приложение-клиент (до завершения процесса аутентификации) о том, что в информационной системе реализованы меры защиты информации, а также о том, что при	Проверить, что при входе нового пользователя в приложение отображается экран запроса согласия пользователя (см. п. 5.3.1, п. 5.3.2, п. 16.2)

	работе в информационной системе пользователем должны быть соблюдены установленные оператором правила и ограничения на работу с информацией. Успешное завершение входа в приложение-клиент осуществляется только после подтверждения пользователем ознакомления с предупреждением (мера защиты УПД.7).	
2.1.2.7	ПО «Blitz Identity Provider» должно обеспечивать (после успешного входа пользователя) оповещение пользователя о дате и времени предыдущего входа в информационную систему от имени этого пользователя, количестве неуспешных попыток входа, зафиксированных с момента последнего успешного входа в информационную систему; об изменениях сведений, относящихся к учетной записи пользователя (в том числе изменении прав доступа), произведенных за период времени не менее, чем с момента предыдущего успешного входа в информационную систему (мера защиты УПД.8).	Проверить, что в приложении личного кабинета пользователь видит в аудите события безопасности по своей учетной записи (см. п. 7.3.2)
2.1.2.8	ПО «Blitz Identity Provider» должно обеспечивать возможность ограничить число параллельных сеансов доступа для каждой учетной записи пользователя. Для пользователей можно ограничить число активных параллельных (одновременных) сеансов (сессий) максимально разрешенным одним сеансом. Должна обеспечиваться возможность задать отдельно ограничение для привилегированных учетных записей (администраторов) и для обычных (непривилегированных) пользователей (мера защиты УПД.9).	Проверить, что невозможна параллельная работа пользователя одновременно в нескольких сеансах, в случае если настроена соответствующая политика ограничения числа параллельных сеансов (см. п. 16.3.2, п. 6.2.6)
2.1.2.9	ПО «Blitz Identity Provider» должно осуществлять блокирование сеанса доступа пользователя после установленного оператором времени его бездействия (неактивности) в ПО «Blitz Identity Provider» или по запросу пользователя (инициированию логаута). Блокирование сеанса доступа пользователя обеспечивает временное приостановление работы пользователя с ПО «Blitz Identity Provider». Для заблокированного сеанса должно осуществляться блокирование любых действий по доступу к информации и устройствам отображения, кроме необходимых для разблокирования сеанса. После блокировки сеанса, на устройстве отображения (мониторе) не должна отображаться информация сеанса. Блокирование сеанса доступа пользователя в ПО «Blitz Identity Provider» сохраняется до прохождения пользователем повторной идентификации и аутентификации в соответствии с ИАФ.1. ПО «Blitz Identity Provider» обеспечивается блокирование сеанса доступа пользователя после времени бездействия (неактивности) пользователя до 5 минут (мера защиты УПД.10).	Проверить, что при неактивности пользователя его сеанс входа завершается после настроенного периода неактивности (см. п. 4)
2.1.3.1	ПО «Blitz Identity Provider» должно осуществлять регистрацию и хранение следующих событий безопасности: – вход (выход), а также попытки входа субъектов доступа в приложения-клиенты; – результаты проверки прав доступа приложений к сервисам при вызове приложениями защищаемых сервисов через подсистему шлюза безопасности; – регистрация учетных записей пользователей; – изменение атрибутов учетных записей пользователей; – изменение аутентификационной информации учетных записей пользователей (изменение/сброс/восстановление пароля, настроек двухфакторной аутентификации, привязок учетных записей внешних систем входа); – блокирование/разблокирование/удаление учетной записи пользователя. Состав и содержание информации о событиях безопасности, подлежащих регистрации, определяются в соответствии с РСБ.2. ПО «Blitz Identity Provider» должно регистрировать события, связанные с действиями от имени привилегированных учетных записей (администраторов), события, связанные с изменением привилегий учетных записей (мера защиты РСБ.1).	Проверить в консоли управления отображение событий безопасности с различными типами событий (см. п. 11).

2.1.3.2	ПО «Blitz Identity Provider» должно регистрировать следующий состав и содержание информации о событии безопасности: – типа события безопасности; – даты и временя события безопасности; – идентификационная информации источника события безопасности; – результат события безопасности (успешно или неуспешно); – субъекты доступа (пользователь и приложение-клиент), связанные с данным событием безопасности. При регистрации входа субъектов доступа содержание информации должно дополнительно включать идентификатор, предъявленный при попытке доступа. При регистрации попыток доступа приложений-клиентов к защищаемым сервисам содержание регистрационных записей должно включать дату и время попытки доступа к сервису с указанием ее результата (успешная, неуспешная), идентификатор субъекта доступа (приложения-клиента), URL запрашиваемого защищаемого сервиса. При изменении администратором конфигурации ПО «Blitz Identity Provider» обеспечивается запись предыдущего набора настроек с пометкой информации о том, какой администратор и когда изменил конфигурацию. Записываемой информации должно быть достаточно для отслеживания действий привилегированных пользователей в ПО «Blitz Identity Provider» (мера защиты РСБ.2).	Проверить в консоли управления состав сведений в зарегистрированных событиях безопасности для различных типов событий (см. п. 11).
2.1.3.3	ПО «Blitz Identity Provider» должно обеспечивать сбор, запись и хранение информации о событиях безопасности (мера защиты РСБ.3).	Проверить в консоли управления отображение событий безопасности (см. п. 11).
2.1.3.4	ПО «Blitz Identity Provider» должно предупреждать администратора о сбоях при регистрации событий безопасности, в том числе аппаратные и программные ошибки, сбои в механизмах сбора информации и достижение предела или переполнения объема (емкости) памяти (мера защиты РСБ.4).	Проверить, что события функционирования в системе фиксируются в журналах событий (см. п. 17)
2.1.3.5	ПО «Blitz Identity Provider» должно предоставлять администратору возможность просмотра записей событий безопасности, подлежащих регистрации в соответствии с РСБ.1 (мера защиты РСБ.5).	Проверить в консоли управления отображение событий безопасности (см. п. 11).
2.1.3.6	ПО «Blitz Identity Provider» должно осуществлять синхронизацию системного времени с NTP-сервером (мера защиты РСБ.6).	Проверить в консоли управления (см. п. 11), что зарегистрированное в событиях безопасности время по проверочным действиям (например, тестовому входу в приложение) соответствует точному времени события.
2.1.3.7	ПО «Blitz Identity Provider» должно обеспечивать защиту информации о событиях безопасности. Доступ к записям аудита должен предоставляться администраторам (просмотр записей аудита всех пользователей), и пользователям (просмотр только собственных записей аудита). Должно предусматриваться резервное копирование записей регистрации (аудита) (мера защиты РСБ.7).	Проверить, что регулярно создаются файлы с резервными копиями СУБД (см. п. 2.1.3)
2.1.3.8	ПО «Blitz Identity Provider» должно предоставлять возможность просмотра информации о действиях отдельных пользователей в информационной системе (мера защиты РСБ.8).	Проверить в консоли управления отображение событий безопасности (см. п. 11).