Identity Blitz

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «BLITZ IDENTITY PROVIDER»

Версия 5.11.3

РУКОВОДСТВО ПО ИНТЕГРАЦИИ

1147746651733.62.01.000.001.И3

Москва, 2023

Содержание

ВВЕДЕНИЕ

В данном документе содержится техническая информация о процессе подключения приложений к Blitz Identity Provider. Этот процесс включает следующие этапы:

1. Выбор протокола взаимодействия.
2. Получение регистрационных данных для подключения приложения.
3. Конфигурирование приложения или его доработка для подключения к Blitz Identity Provider.

1. Выбор протокола взаимодействия

При интеграции приложения с Blitz Identity Provider для проведения идентификации и аутентификации пользователя следует выбрать один из протоколов взаимодействия:

- OpenID Connect 1.0 (OIDC)[1]/OAuth 2.0[2] – современный SSO-протокол, изначально ориентированный на работу с веб-приложениями в сети Интернет. Если создается новое приложение, то рекомендуется подключить его к Blitz Identity Provider с использованием OIDC/OAuth 2.0.
- SAML 1.0/1.1/2.0[3] – SSO-протокол, позволяющий подключить различное корпоративное ПО или облачные приложения к сервису входа. Подключаемое приложение возможно имеет встроенную поддержку SAML или такая поддержка может быть добавлена в качестве дополнительной опции или через установку интеграционного коннектора/плагина.

Выбор протокола во многом зависит от того, какое приложение требуется подключить:

- если приложение поддерживает один из SSO-протоколов, то стоит подключать его с использованием данного протокола;
- если предложение не поддерживает протоколы, то следует провести его доработку – в этом случае рекомендуется поддержать взаимодействие по OIDC;
- если приложение только создается, то на этой стадии целесообразно поддержать один из SSO-протоколов – поддержку OIDC реализовать проще, однако при использовании доступных библиотек SAML можно использовать и этот протокол.

В таблице ниже приведены некоторые особенности протоколов OIDC и SAML.

Таблица 1 — Особенности протоколов подключения

	OIDC/OAuth 2.0	**SAML 1.0/1.1/2.0**
Способ обеспечения доверия между приложением и Blitz Identity Provider	Секрет приложения (обычно в виде строки), известный Blitz Identity Provider	Электронная подпись. И запросы на аутентификацию, и ответы – это подписанные XML-документы
Способ взаимодействия	Через веб-браузер пользователя проходит аутентификация. Для завершения аутентификации серверная часть приложения должна	Обычно запрос на аутентификацию и ответ проходят через веб-браузер пользователя. Приложение и

[1] OpenID Connect Core 1.0, см.: https://openid.net/specs/openid-connect-core-1_0.html

[2] RFC 6749 «The OAuth 2.0 Authorization Framework», см.: https://tools.ietf.org/html/rfc6749

[3] См.: https://www.oasis-open.org/standards#samlv2.0

	сформировать HTTP-запрос в адрес Blitz Identity Provider	Blitz Identity Provider могут не иметь сетевой связности
Получение сведений о пользователе	Два способа получения данных о пользователе: – Приложение обращается к REST-сервису Blitz Identity Provider и получает данные о пользователе в формате JSON. Приложение может продолжать получать данные о пользователе, даже когда пользователь завершает свою онлайн-сессию – Приложение получает данные пользователя из маркера идентификации (id_token в форме JWT), полученного от Blitz Identity Provider по результатам входа	Данные о пользователе содержатся в ответе на запрос на аутентификацию в формате XML. Приложение может получать от Blitz Identity Provider данные только в момент входа пользователя
Поддерживаемые приложения	Веб-приложения и мобильные приложения	Веб-приложения

OIDC позволяет реализовать все основные сценарии SAML, но при этом используется более простой JSON/REST-протокол. Существенное преимущество OIDC – поддержка мобильных приложений.

Если подключаемое к Blitz Identity Provider приложение доработать невозможно, но при этом приложение представляет собой веб-приложение, развернутое в собственной инфраструктуре (On-Premise), то подключить приложение к Blitz Identity Provider можно с использованием веб-прокси и специально реализованного в Blitz Identity Provider протокола Simple. Инструкции по такой интеграции приведены в руководстве администратора в главе «Настройка Simple».

2. Подготовка заявки на подключение

2.1. Подключение приложений по OIDC

Взаимодействие приложения с Blitz Identity Provider при проведении идентификации и аутентификации пользователя осуществляется по спецификации OpenID Connect 1.0 (далее – OIDC)[4], в основе которой лежит фреймворк OAuth 2.0[5].

OIDC является современной спецификацией, описывающей порядок использования OAuth 2.0 при обеспечении идентификации/аутентификации пользователей. OIDC изначально ориентирован на работу с веб-приложениями и мобильными приложениями в сети Интернет.

В таблице ниже приведена краткая характеристика спецификации OIDC.

Таблица 2 – Краткая характеристика спецификации OIDC

Способ обеспечения доверия между приложением и Blitz Identity Provider	Секрет приложения (обычно в виде строки), известный Blitz Identity Provider и приложению
Способ взаимодействия	Через браузер пользователя проходит инициируется идентификация/аутентификация. Для завершения идентификации серверная часть приложения должна сформировать HTTP-запрос в адрес Blitz Identity Provider
Получение сведений о пользователе	Два способа получения данных о пользователе: – приложение обращается к REST-сервису Blitz Identity Provider (пп. 3.6) и получает данные о пользователе в формате JSON. Приложение может продолжать получать данные о пользователе, даже когда пользователь завершает свою онлайн-сессию – приложение получает данные пользователя из маркера идентификации (*id_token* в форме JWT), полученного от Blitz Identity Provider по результатам аутентификации
Поддерживаемые приложения	Веб-приложения и мобильные приложения

Аутентификация в терминологии OIDC/OAuth 2.0 является результатом взаимодействия трех сторон:

– сервиса авторизации (*Authorization Server*) или поставщика ресурса (*Resource Server*), в качестве которых выступает Blitz Identity Provider;

[4] «OpenID Connect Core 1.0», см.: https://openid.net/specs/openid-connect-core-1_0.html

[5] RFC 6749 «The OAuth 2.0 Authorization Framework», см.: https://tools.ietf.org/html/rfc6749

- системы-клиента (*Client*), в качестве которой выступает приложение, которое запрашивает доступ ресурсу (информации и данным пользователя);
- владельца ресурса (*Resource Owner*), в качестве которого выступает пользователь, так как в ходе аутентификации он разрешает доступ к данным о себе.

2.1.1. Регистрация приложения в Blitz Identity Provider

Приложение как система-клиент должно зарегистрироваться в Blitz Identity Provider и получить следующие данные, необходимые для формирования запросов на проведение аутентификации.

При подключении к Blitz Identity Provider веб-приложений:

- идентификатор приложения (*client_id*);
- секрет приложения (*client_secret*).

При подключении к Blitz Identity Provider мобильных приложений:

- идентификатор мобильного приложения (*software_id*);
- первичный маркер доступа (*Initial Access Token*);
- метаданные приложения[6] в форме JWS-токена (*software_statement*).

Для регистрации приложения в Blitz Identity Provider необходимо направить заявку, которая включает в себя следующие сведения:

1. Тип подключаемого приложения (веб-приложение или мобильное приложение).
2. Разрешенные адреса возврата (списки *redirect_uri* и *post_logout_redirect_uri*).
3. Перечень запрашиваемых разрешений (список *scope*).
4. Указание нестандартных режимов, необходимых приложению:
   - приложению необходимо получать *refresh_token* – по умолчанию приложению *refresh_token* возвращаться не будет; при выборе этого режима нужно дополнительно указать требуемый срок действия *refresh_token* (по умолчанию срок действия маркера будет 1 день, максимально можно запросить срок действия 365 дней);
   - приложению необходимо использовать нестандартный сценарий взаимодействия (например, Implicit Flow, Hybrid Flow) – по умолчанию приложению разрешено использовать только Authorization Code Flow;

[6] При разработке мобильного приложения можно использовать как общие Initial Access Token и software_statement для своих iOS/Android-реализаций, так и запросить получение различных наборов Initial Access Token и software_statement для каждой ОС и, возможно, каждой редакции (телефон/планшет) и даже версии приложения. Для простоты дальнейшего изложения в тексте документа будет подразумеваться, что мобильное приложение использует один общий Initial Access Token и один общий software_statement.

- приложению нужно получать маркер доступа в формате JWT – по умолчанию маркер доступа предоставляется в формате opaque;
- приложению нужно получать маркер доступа (*access_token*) с нестандартным сроком действия – стандартно маркер доступа действует 1 час.

> **П р и м е ч а н и е —** При указании в заявке на подключение к Blitz Identity Provider необходимости нестандартных режимов, перечисленных выше, **необходимо предоставить обоснование** причин их необходимости.

5. Перечень дополнительных атрибутов, которые Blitz Identity Provider должен добавить в маркер идентификации (дополнительные атрибуты для передачи в составе *id_token*).
6. Режим входа (вход как физического лица или как представителя организации).

2.1.2. Определение разрешенных адресов возврата

Запрос на проведение идентификации/аутентификации пользователя содержит ссылку возврата при авторизации (*redirect_uri*), куда должен быть возвращен пользователь после прохождения идентификации/аутентификации. Допустимые ссылки возврата должны соответствовать зарегистрированным в Blitz Identity Provider разрешенным префиксам. Если в запросе на идентификацию/аутентификацию указана ссылка возврата, и она не соответствует ни одному из указанных префиксов, то в идентификации/аутентификации будет отказано.

В зависимости от типа подключаемого приложения рекомендуется использовать следующие префиксы ссылок возврата:

- При подключении веб-приложений в качестве префиксов ссылок возврата использовать доменные имена приложений. Например, если после проведения аутентификации требуется вернуть пользователя на *https://domain.com/callback*, то в качестве префикса ссылки возврата следует указать в заявке *https://domain.com/*.

> **П р и м е ч а н и е —** при подключении к продуктивной среде Blitz Identity Provider веб-приложение должно использовать в качестве redirect_uri и post_logout_redirect_uri только HTTPS-обработчики. Использование HTTP для взаимодействия с продуктивной средой Blitz Identity Provider запрещено.

- При подключении мобильных приложений в качестве префиксов ссылок возврата рекомендуется указать сами ссылки возврата одного из типов: ссылки типа «private-use URI scheme» (например,

com.example.app:/oauth2redirect/example-provider) или ссылки типа «Universal links»[7] (например, *https://app.example.com/oauth2redirect/example-provider*).

Запрос на проведение логаута содержит ссылку возврата при логауте (*post_logout_redirect_uri*). Эта ссылка указывает, куда должен быть возвращен пользователь после успешно выполненного логаута. Допустимые ссылки возврата должны соответствовать зарегистрированным в Blitz Identity Provider разрешенным префиксам (префикс должен содержать доменное имя приложения и часть пути, минимум, *https://domain.com/*). Если в запросе на логаут указана ссылка возврата, и она не соответствует ни одному из указанных префиксов, то будет отображена ошибка.

2.1.3. Определение запрашиваемых разрешений

Разрешения (*scope* в терминологии OIDC/OAuth 2.0) определяют, какие данные и какие именно права на выполнение каких операций получит приложение по результатам аутентификации. Перечень предусмотренных в Blitz Identity Provider разрешений представлен в таблице 3.

Таблица 3 – Доступные разрешения (scope)

№	Разрешение	Описание	Состав получаемых атрибутов[8]
1.	openid	Техническое разрешение, указывающее на то, что аутентификация проводится согласно спецификации OIDC	При запросе этого *scope* Blitz Identity Provider предоставляет приложению *id_token*. Из *id_token* приложение может получить нужные ему атрибуты пользователя (см. пп. 3.3.4).
2.	profile	Основные данные профиля пользователя	− *sub*– уникальный идентификатор − *family_name* – фамилия − *given_name* – имя − *middle_name* – отчество − *email* – служебный адрес электронной почты − *phone_number* – номер мобильного телефона
3.	usr_grps	Получение списка групп пользователя	− *groups* – список групп, в которые включен пользователь. Каждая

[7] Ссылки типа «Universal links» доступны начиная с iOS 9 и Android 6.0 и являются предпочтительными для использования. Ссылки «private-use URI scheme» рекомендуется использовать только в случае, если приложение должно работать на более ранних версиях iOS/Android.

[8] Для получения сведений о пользователе используется сервис, описанный в пп. 3.6.

			запись в списке включает следующие атрибуты организации: • *id* – идентификатор группы • *name* – имя группы
4.	native	Разрешение на выполнение сквозного входа в веб-приложение из мобильного приложения	Актуально только для мобильных приложений (пп. 3.4.8).
Список разрешений и набор предоставляемых атрибутов пользователя приведены в качестве образца. Содержание таблицы необходимо скорректировать в зависимости от конкретных настроек, сделанных при внедрении Blitz Identity Provider. См. «Руководство администратора»			

2.1.4. Определение дополнительных атрибутов для включения в id_token

Обычно нет необходимости получать атрибуты пользователя непосредственно из маркера идентификации (*id_token*) – более простым и рекомендуемым способом является получение данных пользователя через вызов REST-сервиса (пп. 3.6).

Если все же необходимо получить сведения о пользователе в составе *id_token* (пп. 3.3.4), то в заявке на подключение необходимо перечислить необходимые для включения в маркер идентификации дополнительные атрибуты. Доступные для получения атрибуты приведены в таблице 4.

Таблица 4 – Возможные дополнительные атрибуты пользователя в id_token

№	Атрибут	Описание
1.	family_name	Фамилия
2.	given_name	Имя
3.	middle_name	Отчество
4.	email	Адрес электронной почты
5.	phone_number	Мобильный телефон
Следующие атрибуты заполняются только если пользователь вошел в Blitz Identity Provider через ЕСИА в качестве сотрудника организации.		
6.	org_id	Идентификатор организации в Blitz Identity Provider
7.	global_role	Выбранная роль при входе через ЕСИА: – P – физическое лицо – B – индивидуальный предприниматель – L – сотрудник юридического лица – A – сотрудник органа государственной власти
8.	org_shortname	ОГРН организации (по сведениям из учетной записи ЕСИА)
9.	org_fullname	ОГРН организации (по сведениям из учетной записи ЕСИА)

10.	org_ogrn	ОГРН организации (по сведениям из учетной записи ЕСИА)
11.	org_inn	ИНН организации[9] (по сведениям из учетной записи ЕСИА)
Список дополнительных атрибутов приведен в качестве образца. Содержание таблицы необходимо скорректировать в зависимости от конкретных настроек, сделанных при внедрении Blitz Identity Provider. См. «Руководство администратора»		

2.2. Подключение приложений по SAML

Аутентификация в терминологии SAML является результатом взаимодействия трех сторон:

- поставщик идентификации (*Identity Provider*), в качестве которого выступает Blitz Identity Provider;
- поставщик услуги (*Service Provider*), в качестве которого выступает подключаемое приложение;
- веб-браузер пользователя (*User Agent*).

Приложение (поставщик услуг) должно быть зарегистрировано в Blitz Identity Provider. Для этого необходимо направить заявку согласно Регламенту подключения к Blitz Identity Provider. Заявка включает XML-файл с метаданными поставщика услуг или значения параметров, необходимые для самостоятельной подготовки метаданных администраторами Blitz Identity Provider.

2.2.1. Подготовка метаданных поставщика услуг

Метаданные поставщика услуг описывают настройки подключения приложения к Blitz Identity Provider (например, URL конечных точек приложения, ключи для проверки ЭП). Для описания метаданных используется язык XML[10].

Метаданные должны быть подготовлены по результатам выполнения работ по добавлению поддержки протокола (п. 4.1).

Если приложение является готовым ПО, поддерживающим SAML, то метаданные должны быть получены согласно документации на это ПО. Обычно такое ПО предоставляет URL, по которому может быть получены метаданные.

[9] При аутентификации юридического лица с помощью электронной подписи ИНН организации передается в формате «00 + 10 цифр ИНН юридического лица», при аутентификации юридического лица с помощью учетной записи ЕСИА - в формате «10 цифр ИНН юридического лица».

[10] Подробнее про метаданные SAML см.: https://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

Если ПО подключаемого приложения не предусматривает выгрузку метаданных, но в документации на ПО описаны параметры, которые должны быть настроены для подключения приложения, то можно указать эти параметры, так, чтобы метаданные на их основе были самостоятельно подготовлены Администратором Blitz Identity Provider. В этом случае необходимо указать следующие параметры:

1. Идентификатор поставщика услуг (*entityID*) – следует указать, только если приложению необходим конкретный *entityID*. Иначе *entityID* будет самостоятельно присвоен Администратором Blitz Identity Provider.
2. Сертификат[11] открытого ключа приложения (поставщика услуг) – должен быть указан только в случае, если приложение подписывает SAML-запрос при отправке к Blitz Identity Provider. Должны использоваться ключи RSA-2048. Допустимо использовать самоподписанные сертификаты с длительным сроком действия.
3. URL для приема от Blitz Identity Provider SAML-ответа – приложение должно предоставлять обработчик, осуществляющий прием от Blitz Identity Provider SAML-ответов с результатами входа. Обычно эта настройка приложения называется *Assertion Consumer Service*.
4. URL для приема от Blitz Identity Provider запроса на логаут – выборочная настройка. Если приложение поддерживает единый логаут, то оно может предоставлять обработчик единого логаута. Обычно эта настройка приложения называется *Single Logout Service Location*.
5. URL для перенаправления пользователя в приложение после успешного логаута – опциональная настройка. Если приложение поддерживает единый логаут и может инициировать единый выход, то оно может предоставлять URL для возврата пользователя после логаута. Обычно эта настройка приложения называется *Single Logout Service Response Location*.
6. Перечень запрашиваемых атрибутов (*SAML Assertion*).
7. Признак необходимости передачи атрибутов в зашифрованном виде[12].

2.2.2. Перечень запрашиваемых атрибутов (SAML Assertion)

В метаданных указывается перечень атрибутов пользователя (*SAML Assertion*). Предусмотренные в Blitz Identity Provider атрибуты пользователя приведены в таблице ***5***.

[11] Сертификат поставщика услуг отличается от TLS-сертификата подключаемого веб-сайта. Обычно это самоподписанный сертификат с длительным сроком действия.

[12] Атрибуты в SAML-сообщении всегда передаются подписанными. Включать шифрование атрибута целесообразно, если пользователь не должен иметь возможности прочитать значение атрибута.

Таблица 5 — Доступные атрибуты пользователя (SAML Assertion)

№	Атрибут	Описание
1.	logonname	Логин пользователя в домене
2.	surname	Фамилия
3.	firstname	Имя
4.	middlename	Отчество
5.	email	Служебный адрес электронной почты
Список SAML атрибутов приведен в качестве образца. Содержание таблицы необходимо скорректировать в зависимости от конкретных настроек, сделанных при внедрении Blitz Identity Provider. См. «Руководство администратора».		

2.3. Особенности подключения мобильных приложений

При создании в мобильных приложениях функции входа с использованием Blitz Identity Provider рекомендуется учитывать следующие особенности:

- пользователям мобильных приложений неудобно вводить при каждом входе логин и пароль на веб-странице аутентификации Blitz Identity Provider. Вместо этого им привычнее при повторных входах использовать ПИН-код приложения или Touch ID/Face ID;
- пользователь может использовать свою учетную запись Blitz Identity Provider для входа в несколько установок одного и того же мобильного приложения (например, войти в приложение, установленное на iPhone, и войти в это же приложение, установленное на iPad). Пользователь должен иметь возможность отозвать выданные этим установкам приложений права доступа к своим сведениям в Blitz Identity Provider;
- по причинам безопасности нежелательно хранить на устройстве пользователя (внутри сборки мобильного приложения) пароль приложения (*client_secret*), используемый для взаимодействия приложения с Blitz Identity Provider.

Чтобы учесть изложенные выше особенности, в Blitz Identity Provider предусмотрен ряд специальных механизмов, предназначенных для использования мобильными приложениями. Рекомендуемый сценарий взаимодействия мобильного приложения с Blitz Identity Provider описан в пп. 3.4.

2.4. Добавление элементов дизайна приложения на страницу входа

Blitz Identity Provider позволяет поместить элементы дизайна приложения на страницу входа Blitz Identity Provider. При желании создать для подключаемой системы персонифицированную страницу входа нужно отметить соответствующий пункт в заявке на подключение. В этом случае при обработке заявки Администратор Blitz Identity Provider

вместе с данными о зарегистрированных параметрах подключения вышлет шаблон оформления страницы входа.

Шаблон оформления страницы входа представляет собой zip-архив, внутри которого записаны HTML каркаса страницы входа и используемые на странице таблица стилей, изображения, JavaScript обработчики.

Ответственные за подключаемую систему могут скорректировать присланное содержимое, адаптировав тем самым верстку страницы входа под требования дизайна подключаемой системы.

Подготовленный архив темы страницы входа нужно передать Администратору Blitz Identity Provider для загрузки и проверки в тестовом контуре. Если полученный внешний вид страницы входа будет приемлемым, то Администратор Blitz Identity Provider перенесет настройки внешнего вида в ПРОД-контур Blitz Identity Provider.

Ответственность за дальнейшее сопровождение нестандартной темы сохраняется за ее разработчиками (ответственными за подключаемую систему). В случае системных изменений в основной теме Администратор Blitz Identity Provider обращается к ответственным с требованием внести аналогичные изменения в нестандартную тему. В случае если изменения не внесены в отведенный срок, то приложение переключается на использование стандартной темы Blitz Identity Provider.

3. Подключение приложений по OIDC

3.1. Настройки подключения

По результатам рассмотрения направленной заявки и регистрации веб-приложения в Blitz Identity Provider Администратором Blitz Identity Provider будет предоставлена следующая информация:

– идентификатор, присвоенный приложению в Blitz Identity Provider (*client_id*);
– секрет приложения (*client_secret*);
– зарегистрированные для приложения URL возврата при авторизации;
– зарегистрированные для приложения URL возврата при логауте;
– зарегистрированные для приложения разрешения (*scope*).

По результатам исполнения заявки на подключение к Blitz Identity Provider мобильного приложения будет предоставлена следующая информация:

– идентификатор, присвоенный приложению в Blitz Identity Provider (*software_id*);
– первичный маркер доступа (*Initial Access Token*);
– метаданные приложения (*software_statement*);
– зарегистрированные для приложения URL возврата при авторизации;
– зарегистрированные для приложения URL возврата при логауте;
– зарегистрированные для приложения разрешения (*scope*).

Все URL в данном разделе необходимо скорректировать в зависимости от конкретных настроек, сделанных при внедрении Blitz Identity Provider. См. «Руководство администратора».

В целях взаимодействия с Blitz Identity Provider приложение должно использовать следующие адреса:

– URL для проведения авторизации и аутентификации:
  • *https://login-test.company.com/blitz/oauth/ae* (тестовая среда)
  • *https://login.company.com/blitz/oauth/ae* (продуктивная среда)
– URL для получения и обновления маркера доступа:
  • *https://login-test.company.com/blitz/oauth/te* (тестовая среда)
  • *https://login.company.com/blitz/oauth/te* (продуктивная среда)
– URL для получения данных пользователя:
  • *https://login-test.company.com/blitz/oauth/me* (тестовая среда)
  • *https://login.company.com/blitz/oauth/me* (продуктивная среда)
– URL для динамической регистрации экземпляра мобильного приложения:
  • *https://login-test.company.com/blitz/oauth/register* (тестовая среда)
  • *https://login.company.com/blitz/oauth/register* (продуктивная среда)

– URL для получения кода подтверждения авторизации (OAuth 2.0 Device Authorization Grant):
  - *https://login-test.company.com/blitz/oauth/da* (тестовая среда)
  - *https://login.company.com/blitz/oauth/da* (продуктивная среда)
– URL для получения данных о маркере доступа:
  - *https://login-test.company.com/blitz/oauth/introspect* (тестовая среда)
  - *https://login.company.com/blitz/oauth/introspect* (продуктивная среда)
– URL для выполнения логаута:
  - *https://login-test.company.com/blitz/oauth/logout* (тестовая среда)
  - *https://login.company.com/blitz/oauth/logout* (продуктивная среда)

Все эти URL, а также дополнительные сведения, размещены по адресу динамически обновляемых настроек (метаданных) каждой среды Blitz Identity Provider[13]:

– *https://login-test.company.com/blitz/oauth/.well-known/openid-configuration* (тестовая среда)
– *https://login.company.com/blitz/oauth/.well-known/openid-configuration* (продуктивная среда)

Разработчики приложений могут не прописывать все указанные URL в конфигурации своего приложения, а использовать в настойках единую ссылку на метаданные Blitz Identity Provider.

Также в зависимости от настроек, сделанных при развертывании Blitz Identity Provider, могут быть доступны дополнительные сервисы, описанные в Приложениях.

3.2. Готовые библиотеки

Для интеграции приложения с Blitz Identity Provider можно использовать одну из множества готовых OAuth 2.0 библиотек[14] или реализовать взаимодействие самостоятельно.

Для интеграции мобильных приложений с Blitz Identity Provider будет полезен информационный ресурс *https://appauth.io/*, предоставляющий SDK для iOS/Android.

В подразделе 3.3 описан сценарий интеграции с Blitz Identity Provider веб-приложения для проведения идентификации/аутентификации пользователя.

В подразделе 3.4 описан сценарий интеграции с Blitz Identity Provider мобильного приложения для проведения идентификации/аутентификации пользователя.

[13] RFC 8414 «OAuth 2.0 Authorization Server Metadata», см.: https://tools.ietf.org/html/rfc8414

[14] См.: https://oauth.net/code/#client-libraries

В разделе 4 описаны предоставляемые Blitz Identity Provider сервисы доступа к информации о пользователе.

3.3. Подключение веб-приложения

3.3.1. Общие сведения

Взаимодействие веб-приложения с Blitz Identity Provider по OIDC включает в себя следующие этапы[15]:

- приложение через веб-браузер отправляет запрос на идентификацию и аутентификацию пользователя в адрес Blitz Identity Provider;
- Blitz Identity Provider проводит идентификацию/аутентифицирует пользователя;
- Blitz Identity Provider получает согласие пользователя на передачу информации о нем в приложение (для приложений, которые размещены на домене *company.com*, согласие предоставляется автоматически без запроса пользователя);
- Blitz Identity Provider через веб-браузер перенаправляет пользователя обратно в приложение и передает в приложение код авторизации;
- приложение с использованием кода авторизации формирует запрос на получение маркера идентификации, маркера обновления, маркера доступа;
- приложение получает ответ, содержащий необходимые маркеры;
- приложение запрашивает данные пользователя по маркеру доступа. При необходимости приложение может провести проверку маркера идентификации и извлечь из этого маркера идентификатор пользователя и дополнительные атрибуты.

На рисунках 1 – 3 представлены процессы получения кода авторизации, маркеров, данных пользователя.

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

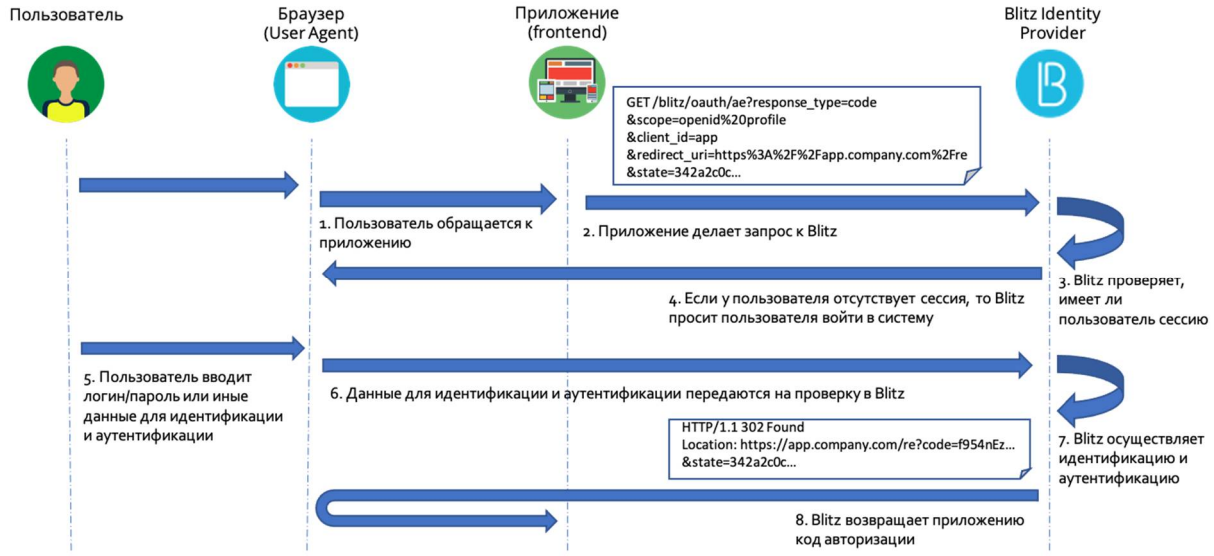

Рисунок 1 – Получение кода авторизации

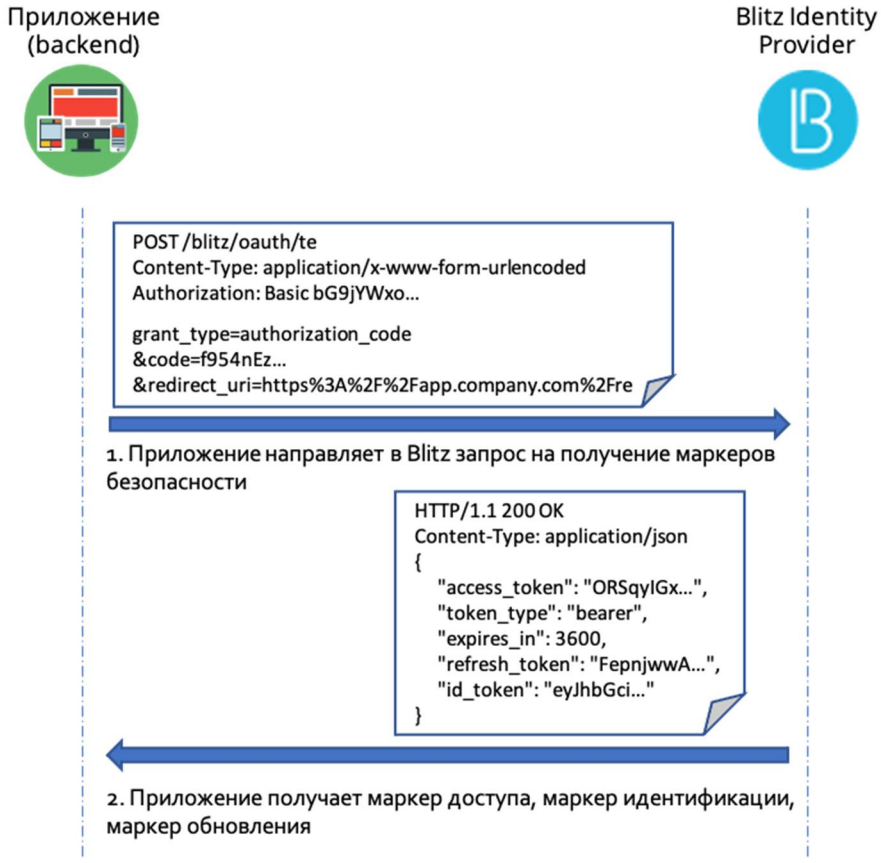

Рисунок 2 – Получение маркеров безопасности

Рисунок 3 – Получение данных пользователя

3.3.2. Запрос на получение кода авторизации

Для проведения идентификации и аутентификации пользователя приложение должно направить пользователя на URL для получения в Blitz Identity Provider кода авторизации, передав в качестве параметров:

- *client_id* – идентификатор клиента;
- *response_type* – тип ответа (принимает значение[16] “code”, “token”, “code token”, “code id_token”, “code id_token token”, “id_token token”, “id_token”);
- *response_mode* (необязательный параметр) – позволяет явно указать требуемый способ передачи кода авторизации. При обычном подключении приложения к Blitz Identity Provider данный параметр передаваться не должен, так как рекомендуется использовать стандартные способы передачи кода авторизации (query – для Authorization Code Flow и fragment – для Implicit/Hybrid Flow).

  Возможные значения параметра response_mode:

  - query – значение кода авторизации (code) возвращается на redirect_uri

[16] Использование иных режимов, кроме “code”, требует специального согласования при подаче заявки на регистрацию приложения в Blitz Identity Provider. Значение параметра response_type указывает выбранный приложением способ взаимодействия с Blitz Identity Provider:
- “code” – Authorization Code Flow;
- “code token”, “code id_token token”, “code id_token token” – Hybrid Flow;
- “id_token token”, “id_token” – OIDC Implicit Flow;
- “token” – OAuth 2.0 Implicit Flow.

приложения в форме query-параметра. Стандартный режим для Authorization Code Flow.

- fragment – значение кода авторизации (code) возвращается на redirect_uri приложения в форме fragment-параметра (#). Стандартный режим для Implicit Flow.

– *scope* – запрашиваемые разрешения, для проведения аутентификации должно быть передано разрешение *openid* и необходимые дополнительные scope для получения данных пользователя, например, *profile* (при запросе нескольких scope они передаются одной строкой и отделяются друг от друга пробелом);
– *redirect_uri* – ссылка для возврата пользователя в приложение, ссылка должна соответствовать одному из зарегистрированных значений;
– *state* – набор случайных символов, имеющий вид 128-битного идентификатора запроса (используется для защиты от перехвата), это же значение будет возвращено в ответе – опциональный параметр;
– *access_type* (необязательный параметр) – требуется ли приложению получать *refresh_token*, необходимый для получения сведений о пользователе в дальнейшем, когда пользователь будет оффлайн. Принимает значение *online* или *offline*, *refresh_token* предоставляется при *access_type=offline*. Если значение не задано, то поведение определяется настройкой, заданной для указанного приложения в Blitz Identity Provider;
– *prompt* (необязательный параметр) – указывает Blitz Identity Provider требуемый режим входа. Возможные значения параметра prompt:

- *none* – запрет на аутентификацию. Если при выполнении запроса у Blitz Identity Provider возникнет потребность отобразить пользователю экран запроса идентификации/аутентификации, то Blitz Identity Provider не будет этого делать, а вернет системе на ее *redirect_uri* ошибку *login_required*. Вызов с параметром *prompt=none* нужно делать в случае, если приложение хочет проверить наличие у пользователя сессии Blitz Identity Provider, но не хочет, чтобы при выполнении такой проверки пользователю отобразился экран входа Blitz Identity Provider.
- *select_account* – запрос смены текущего пользователя. Blitz Identity Provider отобразит пользователю экран выбора аккаунта, чтобы пользователь мог войти под другой учетной записью.
- *login* – запрет на SSO. Если при выполнении запроса Blitz Identity Provider выяснит, что пользователь уже проходит идентификацию/аутентификацию ранее, то Blitz Identity Provider явно потребует от пользователя пройти

идентификацию/аутентификацию заново. При этом Blitz Identity Provider дополнительно проверит, что вход будет осуществлен именно тем же самым пользователем, пользовательская сессия которого открыта. Если при повторной идентификации/аутентификации пользователь выполнит вход под другой учетной записью, то Blitz Identity Provider вернет системе на ее *redirect_uri* ошибку *login_required*. Вызов с параметром prompt=login нужно делать в случае, если приложение хочет явно запросить у пользователя идентификацию/аутентификацию, например, при доступе к требующей повышенной защиты функции приложения.

> **Примечание —** для *prompt=login* для приложения можно при необходимости включить иной сценарий обработки ситуации, что пользователь вошел под другой учетной записью, чем был ранее залогинен в сессии. А именно, можно включить, чтобы при вызове *prompt=login* осуществлялся принудительный логаут текущей сессии и создание сессии под новой учетной записью. Такое поведение не является рекомендуемым, но может быть включено для приложения по отдельному запросу.

- *nonce* (необязательный параметр) – строка, используемая для привязки сессии приложения с маркером идентификации. При стандартном подключении приложения к Blitz Identity Provider с использованием Authorization Code Flow параметр *nonce* использовать нет необходимости. При подключении по Implicit Flow или Hybrid Flow данный параметр должен передаваться. Значение *nonce* должно быть случайной текстовой строкой.
- *display* (необязательный параметр) – параметр в значении *script* передается только в случае запуска процесса входа через HTTP API (см. Приложение В).
- *bip_action_hint* (необязательный параметр) – указывает Blitz Identity Provider, что страница входа должна открыться в одном из специальных режимов:
  - *open_reg* – открыть в режиме регистрации пользователя; при использовании этого режима можно дополнительно указать параметр *login_hint* со значением email пользователя, и тогда поле «Адрес электронной почты» будет перезаполнено указанным значением email;
  - *open_recovery* – открыть в режиме восстановления пароля; при использовании этого режима можно дополнительно указать параметр *login_hint* со значением email пользователя, и тогда поле «Логин» будет перезаполнено указанным значением email;
  - *used_externalIdps:esia:esia_1* – открыть в режиме входа через ЕСИА;
  - *used_externalIdps:esiadp:esiadp_1* – вход с использованием учетной записи

ЕСИА (через получение согласия на доступ к цифровому профилю);

- *used_externalIdps:sbrf:sbrf_1* – открыть в режиме входа через Сбер ID;
- *used_externalIdps:tcs:tcs_1* – открыть в режиме входа через Тинькофф ID;
- *used_externalIdps:mos:mos_1* – открыть в режиме входа через Mos ID (СУДИР);
- *used_externalIdps:apple:apple_1* – открыть в режиме входа через Apple ID;
- *used_externalIdps:facebook:facebook_1* – открыть в режиме входа через Facebook;
- *used_externalIdps:google:google_1* – открыть в режиме входа через Google;
- *used_externalIdps:mail:mail_1* – открыть в режиме входа через Mail ID;
- *used_externalIdps:ok:ok_1* – открыть в режиме входа через Одноклассники;
- *used_externalIdps:vk:vk_1* – открыть в режиме входа через VK;
- *used_externalIdps:yandex:*yandex_1 – открыть в режиме входа через Яндекс;
- *used_password* – открыть в режиме входа по паролю (поведение по умолчанию);
- *used_webAuthn* – открыть в режиме входа с использованием FIDO2 ключа (Passkey);
- used_x509 – открыть в режиме входа по электронной подписи;
- *used_qrCode* – открыть в режиме входа по QR-коду;
- *used_spnego* – открыть в режиме входа по сеансу операционной системы;
- *used_sms* – открыть в режиме входа по коду в SMS;
- *used_outside_methodname* – открыть в режиме входа через внешний метод аутентификации с именем *methodname*.

– *code_challenge_method* (необязательный параметр) – передается значение “S256”, если подключенное приложение поддерживает спецификацию PKCE[17] для дополнительной защиты взаимодействия с Blitz Identity Provider. Для подключения веб-приложений применение PKCE не является обязательным. Для подключения мобильных приложений к Blitz Identity Provider должен использоваться PKCE.

– *code_challenge* (необязательный параметр) – при использовании PKCE в этот параметр передается значение, вычисленное от *code_verifier* по следующей формуле[18]:

```
code_challenge=BASE64URL-ENCODE(SHA256(ASCII(code_verifier)))
```

Примечание — запрещается открывать страницу входа во фрейме. Пользователь должен видеть URL страницы входа, а также иметь возможность убедиться в наличии HTTPS-соединения веб-порталом *login.company.com*.

[17] RFC 7636 «Proof Key for Code Exchange by OAuth Public Clients», см.: https://tools.ietf.org/html/rfc7636

[18] При отладке удобно использовать онлайн-калькулятор, см.: https://example-app.com/pkce

Пример запроса на получение кода авторизации (запрошена идентификация/аутентификация и маркер доступа с разрешениями openid и profile):

```
https://login.company.com/blitz/oauth/ae?client_id=ais
&response_type=code
&scope=openid+profile&access_type=offline
&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f
&redirect_uri=https%3A%2F%2Fapp.company.com%2Fre
```

Пример ответа со значением кода авторизации (*code*) и параметром *state*:

```
https://app.company.com/re?code=f954…nS0&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f
```

Возможные ошибки при вызове */oauth/ae* соответствуют RFC 6749 и описаны по ссылке[19].

Пример запроса на получение кода авторизации, при котором Blitz Identity Provider не должен открыть страницу входа в случае, если пользователь еще не проходил идентификацию/аутентификацию в текущем веб-браузере:

```
https://login.company.com/blitz/oauth/ae?client_id=ais
&response_type=code
&scope=openid+profile&access_type=offline
&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f&prompt=none
&redirect_uri=https%3A%2F%2Fapp.company.com%2Fre
```

Пример ответа с ошибкой, если для получения кода авторизации пользователь должен явно пройти идентификацию/аутентификацию на странице входа Blitz Identity Provider, а запрос был выполнен с параметром *prompt=none*:

```
https://app.company.com/re?error=login_required
&error_description=The+Authorization+Server+requires+End-User+authentication…
&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f
```

Пример запроса на получение кода авторизации, при котором Blitz Identity Provider должен осуществить вход в режиме входа через ЕСИА:

```
https://login.company.com/blitz/oauth/ae?client_id=ais
&response_type=code
&scope=openid+profile&access_type=offline
&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f
&bip_action_hint=used_externalIdps:esia:esia_1
&redirect_uri=https%3A%2F%2Fapp.company.com%2Fre
```

Пример запроса на получение маркера доступа и маркера идентификации с использованием OIDC Implicit Flow:

```
https://login.company.com/blitz/oauth/ae?client_id=ais
&response_type=id_token%20token
&scope=openid+profile
&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f
&nonce=n-0S6_WzA2Mj
&redirect_uri=https%3A%2F%2Fapp.company.com%2Fre
```

Пример ответа от Blitz Identity Provider с маркерами доступа и идентификации, полученными с использованием OIDC Implicit Flow:

```
https://app.company.com/re#access_token=SlAV32hkKG
&token_type=Bearer
&id_token=eyJ0…NiJ9.eyJ1c…I6IjIifX0.DeWt4Qu…ZXso
&expires_in=3600
&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f
```

[19] См.: https://tools.ietf.org/html/rfc6749#section-4.1.2.1

Пример запроса на получение кода авторизации и маркера идентификации с использованием OIDC Hybrid Flow:

```
https://login.company.com/blitz/oauth/ae?client_id=ais
&response_type=code%20id_token
&scope=openid+profile
&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f
&nonce=n-0S6_WzA2Mj
&redirect_uri=https%3A%2F%2Fapp.company.com%2Fre
```

Пример ответа от Blitz Identity Provider с маркерами доступа и идентификации, полученными с использованием OIDC Hybrid Flow:

```
https://app.company.com/re#code=f954…FxS0
&id_token=eyJ0…NiJ9.eyJ1c…I6IjIifX0.DeWt4Qu…ZXso
&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f
```

3.3.3. Получение маркеров

В целях проведения результата идентификации/аутентификации пользователя и получения его данных Blitz Identity Provider выпускает приложению различные маркеры. Перечень маркеров и их назначение приведены в таблице 6.

Таблица 6 – Используемые в Blitz Identity Provider маркеры

№	Название	Обозначение	Предназначение и срок действия
1.	Маркер доступа	*access_token*	Получение доступа к защищенному ресурсу, например, к данным пользователя. Маркер действителен 3600 секунд.
2.	Маркер обновления	*refresh_token*	Обновление маркера доступа. Маркер *refresh_token* предоставляется, только если для приложения при регистрации была указана необходимость получения *refresh_token*, или если в запросе на получение кода авторизации был указан параметр access_type=offline Маркер действителен до момента использования, но не дольше 365 дней.
3.	Маркер идентификации	*id_token*	Получение идентификационной информации, например, идентификатора пользователя. Маркер действителен 3 часа.

После получения кода авторизации приложение должно обменять его на маркеры. Для этого приложение должно сформировать запрос методом POST на URL для получения маркера. Запрос должен содержать заголовок *Authorization* со значением *Basic {secret}*, где *secret* – это *client_id:client_secret* (например, *app:topsecret*) в формате Base64.

> **Примечание** — сервис получения маркеров должен обязательно вызываться с серверов подключенного к Blitz Identity Provider приложения. Вызов сервиса из выполняемого на стороне веб-браузера программного кода (например, из JavaScript кода

> веб-страницы) **ЗАПРЕЩАЕТСЯ**. Полученный маркер доступа (access_token) должен обрабатываться серверной частью приложения и не должен передаваться через браузер пользователя.

Пример заголовка:

```
Authorization: Basic YWlzOmNsaWVudF9zZWNyZXQ=
```

Тело запроса должно содержать следующие параметры:

- *code* – значение кода авторизации, который был ранее получен;
- *grant_type* – принимает значение *authorization_code*, если код авторизации обменивается на маркер доступа;
- *redirect_uri* – ссылка, по которой должен быть направлен пользователь после того, как даст разрешение на доступ (то же самое значение, которое было указано в запросе на получение кода авторизации);
- *code_verifier* (только если используется PKCE) – значение проверочного кода, использованного при расчете *code_challenge* при получении кода авторизации.

Пример запроса:

```
POST /blitz/oauth/te HTTP/1.1
Authorization: Basic cG9ydGFsLmlhc2l1LmxvY2FsOnBvcnRhbC5pYXNpdS5sb2NhbA==
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

grant_type=authorization_code&code=FLZHS…GU
&redirect_uri=https%3A%2F%2Fapp.company.com%2Fre
```

В ответ возвращается маркер доступа, маркер обновления и маркер идентификации.

Пример ответа с успешным выполнением запроса:

```
{
    "id_token": "eyJhbGciOiJSUzI1NiJ9.eyJub…n0=.Ckt…sQ",
    "access_token": "dO-xym…BE",
    "expires_in": 3600,
    "refresh_token": "1lEWX…Iw",
    "token_type": "Bearer"
}
```

Используя полученный маркер доступа, приложение может запросить (пп. 3.6) актуальные данные пользователя из Blitz Identity Provider.

Если код авторизации был уже использован, не совпал *redirect_uri* с ранее использованным в вызове к */oauth/ae*, или истек срок действия кода, либо переданный *code_verifier* не соответствует *code_challenge*, то в качестве ответа будет возвращена ошибка.

Пример ответа с ошибкой:

```
{
    "error": "invalid_grant",
    "error_description": "The provided authorization grant … is invalid, expired, revoked…"
}
```

Возможные ошибки при вызове */oauth/te* соответствуют RFC 6749 и описаны по ссылке[20].

Для обновления маркера доступа приложение должно сформировать запрос методом POST на URL для получения и обновления маркера. Запрос должен содержать аналогичный заголовок *Authorization*, а также следующие параметры в теле запроса:

- *refresh_token* – маркер обновления;
- *grant_type* – принимает значение *refresh_token*, если маркер обновления обменивается на маркер доступа.

Пример запроса:

```
POST /blitz/oauth/te HTTP/1.1
Authorization: Basic cG9ydGFsLmlhc2l1LmxvY2FsOnBvcnRhbC5pYXNpdS5sb2NhbA==
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

grant_type=refresh_token&refresh_token=jj2DA…bQ
```

Приложение может обменять access_token с одним набором разрешений (scopes) и утверждений (claims) на access_token с другим набором разрешений и утверждений с использованием OAuth 2.0 Token Exchange[21]. Это может быть полезно перед передачей access_token от получившего его приложения другому приложению, чтобы приложение получило сокращенный набор разрешений и сведений о пользователе. Для использования обмена маркера доступа приложению должно быть предоставлено специальное разрешение на использование OAuth 2.0 Token Exchange (разрешен grant_type – *urn:ietf:params:oauth:grant-type:token-exchange*). Также должны быть заданы настройки правил обмена маркеров доступа (см. «Руководство администратора»). Для обмена маркера доступа приложение делает вызов методом POST на URL для получения и обновления маркера. Запрос должен содержать аналогичный заголовок *Authorization*, а также следующие параметры в теле запроса:

- *grant_type* – принимает значение *urn:ietf:params:oauth:grant-type:token-exchange*;
- resource – принимает имя ресурса, для передачи которому запрашивается обмен маркера доступа;
- *subject_token_type* – передается требуемый тип получаемого маркера – в текущей версии Blitz Identity Provider поддерживается только тип *urn:ietf:params:oauth:token-type:access_token*;
- *subject_token* – передается значение заменяемого маркера доступа (access_token).

Пример запроса:

```
POST /blitz/oauth/te HTTP/1.1
Authorization: Basic cG9…A==
```

[20] См.: https://tools.ietf.org/html/rfc6749#section-5.2
[21] См.: https://www.rfc-editor.org/rfc/rfc8693.txt

```
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

grant_type=urn:ietf:params:oauth:grant-type:token-
exchange&resource=…&subject_token_type=urn:ietf:params:oauth:token-
type:access_token&subject_token=eyJ…vA
```

Пример ответа с успешным выполнением запроса:

```
{
    "access_token": "eyJr...-g",
    "expires_in": 3600,
    "scope": "openid new_scope",
    "token_type": "Bearer",
    "issued_token_type": "urn:ietf:params:oauth:token_type:access_token"
}
```

Если приложению предоставлено специальное разрешение на использование OAuth 2.0 Resource Owner Password Credentials (ROPC) (разрешен grant_type – *password*), то приложение может запросить получение маркера доступа методом POST на URL для получения и обновления маркера. Запрос должен содержать аналогичный заголовок *Authorization*, а также следующие параметры в теле запроса:

- *grant_type* – принимает значение *password*;
- *username* – содержит логин пользователя;
- *password* – содержит пароль пользователя;
- *scope* – содержит список запрашиваемых разрешений.

Пример запроса:

```
POST /blitz/oauth/te HTTP/1.1
Authorization: Basic cG9…A==
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

grant_type=password&username=testuser&password=testpwd1&scope=profile
```

Пример ответа с успешным выполнением запроса:

```
{
    "access_token": "dO-xym…BE",
    "expires_in": 3600,
    "scope": "profile",
    "token_type": "Bearer"
}
```

3.3.4. Маркер идентификации. Получение данных идентификации

Для получения данных об идентификации и аутентификации приложение может самостоятельно анализировать содержание маркера идентификации (*id_token*). Но можно это не делать, а использовать полученный маркер доступа для запроса в Blitz Identity Provider через специальный сервис (пп. 3.6) актуальных данных пользователя.

Маркер идентификации состоит из трех частей:

- заголовок (*header*), в котором содержится общая информация о типе маркера, в том числе об использованных в ходе его формирования криптографических операциях;
- набор утверждений (*payload* / *claim set*) с содержательными сведениями о маркере;
- подпись (*signature*), которая удостоверяет, что маркер «выдан» Blitz Identity Provider и не был изменен при передаче.

Части маркера разделены точкой, так что он имеет вид:

```
HEADER.PAYLOAD.SIGNATURE
```

Маркер передается в виде строки в формате Base64url.

Заголовок (header) содержит:

- *alg* – описание алгоритма шифрования (параметр alg); в настоящее время в Blitz Identity Provider поддерживается алгоритм электронной подписи RSA SHA-256, рекомендуемый спецификацией (соответствует значению RS256);
- *kid* – идентификатор ключа, использованного для подписи маркера.

Набор утверждений включает следующие атрибуты:

- *exp* – время прекращения действия, указывается в секундах с 1 января 1970 г. 00:00:00 GMT;
- *iat* – время выдачи, указывается в секундах с 1 января 1970 г. 00:00:00 GMT;
- *sub* – идентификатор субъекта, в качестве значения указывается значение идентификатора пользователя;
- *ua_id* – идентификатор устройства пользователя;
- *aud* – адресат маркера, указывается *client_id* приложения, направившего запрос на аутентификацию;
- *iss* – организация, выпустившая маркер, указывается URL Blitz Identity Provider;
- *nonce* – строка безопасности, указывается значение *nonce*, которое было передано приложением к Blitz Identity Provider в исходном запросе к */oauth/ae*. Используется только при Implicit или Hybrid Flow. При получении приложением маркера с использованием Implicit или Hybrid Flow приложение должно сопоставить *nonce* из состава маркера идентификации с *nonce* из своего запроса;
- *at_hash* – половина хэша маркера доступа, передается только при использовании Implicit или Hybrid Flow. Представляет собой закодированную в Base64 левую половину значения функции SHA-256 от *access_token*; Приложение, получившее маркер доступа с использованием Implicit или Hybrid Flow должно извлечь из маркера идентификации значение *at_hash* и сравнить с маркером доступа.
- *c_hash* – половина хэша кода авторизации, передается только в случае использования Hybrid Flow. Представляет собой закодированную в Base64 левую половину (128 бит) значения функции SHA-256 от кода авторизации (*code*); Приложение, получившее код авторизации с использованием Hybrid Flow, должно извлечь из маркера идентификации значение *c_hash* и сравнить с кодом авторизации.
- *amr* – пройденные методы аутентификации, указывается список пройденных пользователем методов аутентификаций. Список может включать следующие идентификаторы методов:

- *password* – вход с использованием пароля;
- *cls:<метод>* (например, *cls:password*) – автоматический вход с запомненного устройства (в названии идентификатора после двоеточия указан метод аутентификации, первично пройденной пользователем, в результате чего произошло запоминание пользователя на данном устройстве);
- *css* – автоматический вход по результатам регистрации пользователя, восстановления пароля или перехода в веб-приложение из мобильного приложения, использующего вызов с использованием *scope=native*;
- *sms* – подтверждение входа с помощью кода в SMS-сообщении (второй фактор аутентификации);
- *push* – подтверждение входа с помощью кода в push-уведомлении в мобильное приложение (второй фактор аутентификации);
- *hotp* – подтверждение входа с помощью кода, сгенерированного HOTP-генератором кодов подтверждения (второй фактор аутентификации);
- *totp* – подтверждение входа с помощью кода, сгенерированного программным TOTP-генератором кодов подтверждения (второй фактор аутентификации);
- *spnego* – вход с использованием сеанса операционной системы;
- *userApp* – вход в мобильное приложение привязанной к устройству учетной записью пользователя (Touch ID/Face ID/ПИН-код);
- *webAuthn* – вход с использованием FIDO2 ключа (Passkey) или подтверждение входа с помощью U2F-ключа;
- *x509* – вход с использованием электронной подписи;
- *qrCode* – вход по QR-коду;
- *externalIdps:esia:esia_1* – вход с использованием учетной записи ЕСИА;
- *externalIdps:esiadp:esiadp_1* – вход с использованием учетной записи ЕСИА (через получение согласия на доступ к цифровому профилю);
- *externalIdps:sbrf:sbrf_1* – вход с использованием учетной записи Сбер ID;
- *externalIdps:tcs:tcs_1* – вход с использованием учетной записи Тинькофф ID;
- *externalIdps:mos:mos_1* – вход с использованием учетной записи Mos ID (СУДИР);
- *externalIdps:apple:apple_1* – вход с использованием учетной записи Apple ID;
- *externalIdps:facebook:facebook_1* – вход с использованием учетной записи в социальной сети Facebook;
- *externalIdps:google:google_1* – вход с использованием учетной записи Google;

- *externalIdps:mail:mail_1* – вход с использованием учетной записи Mail ID;
- *externalIdps:ok:ok_1* – вход с использованием учетной записи в социальной сети Одноклассники;
- *externalIdps:vk:vk_1* – вход с использованием учетной записи в социальной сети VK;
- *externalIdps:yandex:yandex_1* – вход с использованием учетной записи Яндекс;
- *outside_methodname* – признак, что в процессе входа пользователь использовал внешний метод аутентификации с именем *methodname*.

– *sid* – идентификатор сессии пользователя;

– дополнительные атрибуты в соответствии с заявкой на подключение приложения к Blitz Identity Provider (возможные атрибуты для включения в *id_token* приведены в таблице 4.

Пример набора утверждений:

```
{
  "exp": 1445004777,
  "iat": 1444994212,
  "ua_id": "f8a235ff-cb85-4c4b-b55d-544f9358a8d7",
  "sub": "3d10f626-ea77-481d-a50bd4a4d432d86b",
  "amr": [
    "externalIdps:esia:esia_1"
  ],
  "aud": [
    "ais"
  ],
  "iss": "https://login.company.com",
  "sid": "5a600d12-4b14-447e-ba21-2dc40344a44a"
}
```

Подпись (*signature*) маркера осуществляется по алгоритму, который указывается в параметре *alg* маркера. Подпись вычисляется от двух предыдущих частей маркера (HEADER.PAYLOAD). Сертификат открытого ключа Blitz Identity Provider, необходимый для проверки подписи, можно загрузить по следующим ссылкам (находится в атрибуте *x5c,* идентификатор ключа находится в атрибуте *kid*):

– *https://login-test.company.com/blitz/oauth/.well-known/jwks* (тестовая среда)

– *https://login.company.com/blitz/oauth/.well-known/jwks* (продуктивная среда)

После получения маркера идентификации приложению рекомендуется произвести валидацию маркера идентификации, которая включает в себя следующие проверки:

1. Получение идентификатора Blitz Identity Provider (*sub*), содержащегося в маркере идентификации, и получение иных необходимых приложению дополнительных атрибутов пользователя.
2. Проверка идентификатора приложения, т.е. именно приложение должно быть указано в качестве адресата маркера идентификации.
3. Проверка подписи маркера идентификации (с использованием указанного в маркере алгоритма).

4. Проверка, что текущее время должно быть не позднее, чем время прекращения срока действия маркера идентификации.

После валидации маркера идентификации приложение может считать пользователя аутентифицированным.

Для анализа содержания маркера идентификации, а также для упрощения разработки модулей по его проверке можно воспользоваться доступными онлайн-декодерами и библиотеками[22].

3.3.5. Проверка маркера доступа с использование сервиса интроспекции

Данные о маркере доступа (*access_token*) необходимо проверять в следующих случаях:

- приложению требуется отслеживать срок действия маркера, чтобы оперативно менять его на новый;
- к приложению предъявляются повышенные требования к безопасности, и приложение хочет через проверку маркера убедиться, что маркер не аннулирован досрочно. Аннулирование маркера доступам (*access_token*) или маркера идентификации (*id_token*) может произойти в целях безопасности в случае, если произошли сброс/изменение пароля учетной записи пользователя или если учетная запись пользователя была заблокирована;
- приложение является поставщиком ресурсов и предоставляет доступ к этим ресурсам по предъявлению маркера доступа, выданного Blitz Identity Provider приложению, запрашивающему ресурс.

Для запроса данных о маркере доступа необходимо выполнить запрос методом POST по адресу сервиса интроспекции маркера доступа[23]. В запрос для аутентификации системы-клиента должен быть добавлен описанный ранее заголовок *Authorization: Basic*. Также должен быть добавлен заголовок *Content-Type*, принимающий значение *application/x-www-form-urlencoded*.

Сервис интроспекции может быть вызван любой системой, зарегистрированной в Blitz Identity Provider, для проверки любого маркера доступа (система может проверить маркер, выданный другой системе). Проверять можно не только маркер доступа, но и маркер обновления.

В теле запроса могут быть указаны параметры:

---

[22] См.: http://jwt.io/ и http://kjur.github.io/jsjws/mobile/tool_jwt.html#verifier

[23] RFC 7662 «OAuth 2.0 Token Introspection», см.: https://tools.ietf.org/html/rfc7662

- *token* – маркер доступа, данные о котором требуется просмотреть (обязательный параметр);
- *token_type_hint* – тип маркера доступа (например, *access_token*), предназначен для ускорения поиска (опциональный параметр).

Пример запроса:

```
POST /blitz/oauth/introspect HTTP/1.1
Authorization: Basic cG9ydGFsLmlhc2l1LmxvY2FsOnBvcnRhbC5pYXNpdS5sb2NhbA==
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

token=MkvRf…No
```

В ответе будут переданы следующие данные о маркере доступа:

- *active* – признак действительности маркера доступа, принимает значения *true* или *false*. Маркер действителен, если он выдан сервисом авторизации Blitz Identity Provider, не был отозван и срок его действия не истек;
- *scope* – область доступа, на которую выдан маркер доступа. Передается в виде перечня разрешений;
- *client_id* – идентификатор системы-клиента, которая получила данный маркер доступа;
- *sbj* – идентификатор пользователя (владельца ресурса, предоставившего доступ к своим данным), определенный как базовый идентификатор в Blitz Identity Provider. Значение параметра возвращается только в том случае, если он может быть передан в рамках *scope* по предъявленному маркеру доступа;
- *jti* – идентификатор маркера доступа (в виде строки);
- *token_type* – тип предъявленного маркера доступа.

Пример ответа:

```
{
    "sbj": "3d10f626-ea77-481d-a50bd4a4d432d86b",
    "scope": "openid profile",
    "jti": "10jdlNohfHzuv3xoFurvWSPheEJEC7KHdHr-dcaVyYYvV3h0l2sh",
    "token_type": "Bearer",
    "client_id": "ais",
    "active": true
}
```

3.3.6. Самостоятельная проверка маркера доступа приложением

При регистрации приложения в Blitz Identity Provider можно указать, что приложение должно получать маркер доступа (*access_token*) в формате JWT. В этом случае приложение получает возможность самостоятельно проверить маркер доступа, выполнив его разбор. Структура первично полученного маркера доступа будет аналогична структуре маркера идентификации, описанной в пп. 3.3.4. Во вторичных маркерах доступа, полученных в результате обмена маркера обновления (refresh_token), не будет содержаться сессионная информация (будут отсутствовать *amr* и дополнительные атрибуты

пользователя).

Маркеры доступа в формате JWT следует использовать, только в случае если у приложения на это есть особые причины. В остальных случаях рекомендуется использовать обычные маркеры доступа в формате *opaque*.

3.3.7. Обеспечение логаута

Если приложение предоставляет пользователю возможность инициировать выход из приложения (логаут), то приложению для обеспечения выхода недостаточно завершить локальную сессию. Необходимо также вызвать в Blitz Identity Provider операцию логаута. Если этого не сделать, то может возникнуть ситуация, что пользователь в приложении нажал кнопку *Выход*, после чего сразу попробовал нажать кнопку *Вход*, и вместо ожидаемого запроса идентификации и аутентификации сработал *Single Sign-On*, и пользователь сразу автоматически оказался авторизованным.

Для инициирования логаута в Blitz Identity Provider приложение после закрытия своей локальной сессии должно направить пользователя в Blitz Identity Provider на URL для выполнения логаута[24], передав в качестве параметров:

- *id_token_hint (необязательный параметр)* – Blitz Identity Provider проверяет, что *id_token* из значения параметра выпущен им. Дизайн страницы выхода выводится пользователю в соответствии с настроенным для *client_id* из поля *aud* из *id_token*;
- *post_logout_redirect_uri (необязательный параметр)* – адрес возврата в приложение после логаута. Если параметр не задан, то перенаправление в приложение после логаута не осуществляется. Если задан, то проверяется, что значение соответствует хотя бы одному разрешенному префиксу возврата для приложения, соответствующего переданному в *id_token_hint* приложению (поле *aud* из *id_token*). При передаче параметра *post_logout_redirect_uri* обязательно также передать параметр *id_token_hint*;
- *state* – набор случайных символов, имеющий вид 128-битного идентификатора запроса. Это же значение будет возвращено в ответе при перенаправлении пользователя на *post_logout_redirect_uri*.

Пример запроса логаута:

```
https://login.company.com/blitz/oauth/logout?id_token_hint=eyJhbGciOiJSUzI1NiJ9.eyJub…n0=.Ckt…sQ
&post_logout_redirect_uri=https://app.company.com/redirect_uri
&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f
```

Если Blitz Identity Provider успешно завершит логаут, то он перенаправит пользователя по переданному URL обратно в приложение.

[24] Вызов логаута выполняется в соответствии со спецификацией «OpenID Connect RP-Initiated Logout 1.0», см.: https://openid.net/specs/openid-connect-rpinitiated-1_0.html

Допустимые префиксы страниц возврата должны быть зарегистрированы в настройках Blitz Identity Provider, иначе при логауте будет выдана ошибка.

Приложения, подключенные к Blitz Identity Provider по OIDC, могут подписаться на уведомление их о логауте пользователя из Blitz Identity Provider. Поддерживаются следующие возможности:

- Уведомление через веб-браузер (Front channel)[25].
- Уведомление через сервер (Back channel)[26].

Для уведомления через веб-браузер в настройках приложения в Blitz Identity Provider регистрируется обработчик «Ссылка для очистки сессии пользователя в браузере (Front channel)». Если обработчик зарегистрирован и в процессе сессии пользователь входил в приложение, то при вызове пользователем логаута Blitz Identity Provider через браузер на странице выхода пользователя через фрейм <*iframe src= "ссылка"*> вызовет через HTTP GET указанный в настройке обработчик приложения. В случае если была отмечена настройка «Добавлять идентификатор сессии и эмитента в ссылку очистки сессии в браузере (Front channel)», то дополнительно будут переданы следующие параметры в запросе:

- *iss* – идентификатор поставщика идентификации;
- *sid* – идентификатор сессии пользователя.

Пример вызова ссылки для очистки сессии пользователя в браузере (Front channel):

```
https://app.company.com/front_channel_logout?iss=https://login.company.com&sid=4ac78c75-b99d-44dc-9304-d2599c829440
```

В ответ на вызов приложение должно завершить локальную сессию и вернуть ответ *HTTP 200 OK*. Также в ответ должны быть включены заголовки:

```
Cache-Control: no-cache, no-store
Pragma: no-cache
```

> **Примечание** — при реализации на стороне приложения обработчика приема уведомления через веб-браузер следует учитывать особенности современных браузеров, которые противодействуют передаче cookies при вызове обработчиков в фрейме на URL-домены, отличные от URL-домена родительской страницы:
>
> – чтобы cookie стороннего сайта могла быть переданы из фрейма, у cookie должен быть установлен флаг *SameSite=None* и флаг *Secure*, в момент установки или перезаписи cookie не должен передаваться заголовок *X-Frame-Options*, а сам обработчик должен быть доступен по HTTPS;
>
> – вызов обработчика не будет производиться в некоторых браузерах в случае открытия страницы в режиме «инкогнито».

[25] OpenID Connect Front-Channel Logout 1.0, см.: https://openid.net/specs/openid-connect-frontchannel-1_0.html

[26] OpenID Connect Back-Channel Logout 1.0, см.: https://openid.net/specs/openid-connect-backchannel-1_0.html

Для уведомления через сервер в настройках приложения в Blitz Identity Provider регистрируется обработчик «Ссылка для очистки сессии пользователя в приложении (Back channel)». Если обработчик зарегистрирован и в процессе сессии пользователь входил в приложение, то при вызове пользователем логаута сервер Blitz Identity Provider вызовет сервер приложения через HTTP POST на указанный в настройке обработчик приложения. В вызов будет передан маркер логаута *logout_token*, представляющий собой JWT-токен, в теле которого содержатся следующие параметры:

- *iss* – идентификатор поставщика идентификации;
- *aud* – идентификаторы оповещаемых приложений;
- *iat* – время выпуска маркера обновления;
- *jti* – идентификатор маркера логаута;
- *events* – константное значение *http://schemas.openid.net/event/backchannel-logout* согласно спецификации OpenID Connect Back-Channel Logout 1.0;
- *sid* – идентификатор сессии пользователя;
- *sub* – идентификатор пользователя.

В маркере обновления присутствует либо *sub* (если не включена настройка «Добавлять идентификатор сессии и эмитента в ссылку очистки сессии в приложении (Back channel)», либо *sid* (если настройка «Добавлять идентификатор сессии и эмитента в ссылку очистки сессии в приложении (Back channel)» включена).

Пример вызова сервиса очистки сессии пользователя в приложении (Back channel):

```
POST /back_channel_logout HTTP/1.1
Host: app.company.com
Content-Type: application/x-www-form-urlencoded

logout_token=eyJ…J9.eyJ…J9.RV8…Nw
```

Пример разобранного тела маркера логаута при выключенной настройке «Добавлять идентификатор сессии и эмитента в ссылку очистки сессии в приложении (Back channel)»:

```
{
  "iss": "https://login.company.com",
  "aud": [
    "ais"
  ],
  "iat": 1646979918,
  "jti": "ee75ccd8-ad30-4175-9a61-3ae06c1a6730",
  "events": {
    "http://schemas.openid.net/event/backchannel-logout": {}
  },
  "sub": "3d10f626-ea77-481d-a50bd4a4d432d86b"
}
```

Пример разобранного тела маркера логаута при включенной настройке «Добавлять идентификатор сессии и эмитента в ссылку очистки сессии в приложении (Back channel)»:

```
{
  "iss": "https://login.company.com",
  "aud": [
    "ais"
  ],
  "iat": 1646979918,
  "jti": "ee75ccd8-ad30-4175-9a61-3ae06c1a6730",
```

```
  "events": {
    "http://schemas.openid.net/event/backchannel-logout": {}
  },
  "sid": "4ac78c75-b99d-44dc-9304-d2599c829440"
}
```

В ответ на вызов приложение должно:

1. Проверить подпись маркера логаута по аналогии с тем, как выполняется проверка подписи маркера идентификации (см. п. 3.3.4).
2. Проверить, что:
   - *iss* соответствует идентификатору развернутой системы Blitz Identity Provider;
   - *aud* включает идентификатор вызванного приложения;
   - маркер обновления выпущен (*iat*) не ранее 2 минут назад;
   - *sid* или *sub* соответствуют действующим сессиям пользователя.
3. Если какие-то проверки маркера логаута неуспешны, то вернуть код *HTTP 400 Bad Request*.
4. Если все проверки успешны, то завершить локальную сессию пользователя и вернуть *HTTP 200 OK* в случае успеха или *HTTP 501 Not Implemented* в случае, если сессию завершить не удалось. Рекомендуется включить в ответ заголовки:

```
Cache-Control: no-cache, no-store
Pragma: no-cache
```

3.4. Подключение мобильного приложения

3.4.1. Общие сведения

Взаимодействие мобильного приложения с Blitz Identity Provider дополнительно к штатным средствам протокола OIDC/OAuth 2.0 использует спецификации RFC 7591[27] и RFC 7592[28].

Взаимодействие мобильного приложения с Blitz Identity Provider включает следующие этапы:

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

[27] RFC 7591 «OAuth 2.0 Dynamic Client Registration Protocol», см.: https://tools.ietf.org/html/rfc7591

[28] RFC 7592 «OAuth 2.0 Dynamic Client Registration Management Protocol», см.: https://tools.ietf.org/html/rfc7592

Авторизация в Blitz Identity Provider с помощью шифрованной пары *client_id/client_secret*.

4. Удаление в Blitz Identity Provider полученной пары *client_id/client_secret* при логауте (смене учетной записи, выхода из учетной записи) пользователя из мобильного приложения.

Схематично последовательность действий этапов 1-2 представлена на рисунке 4, а этапа 3 – на рисунке 5.

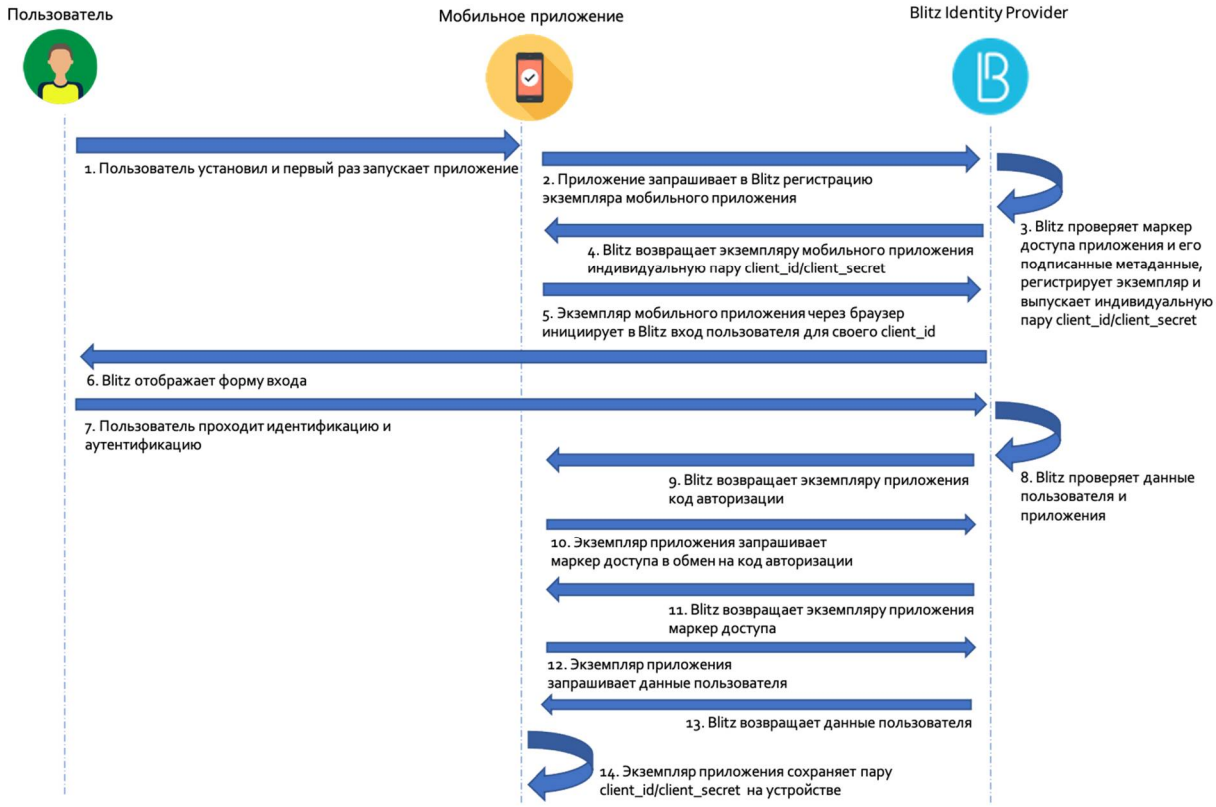

Рисунок 4 – Первый вход пользователя в мобильное приложение

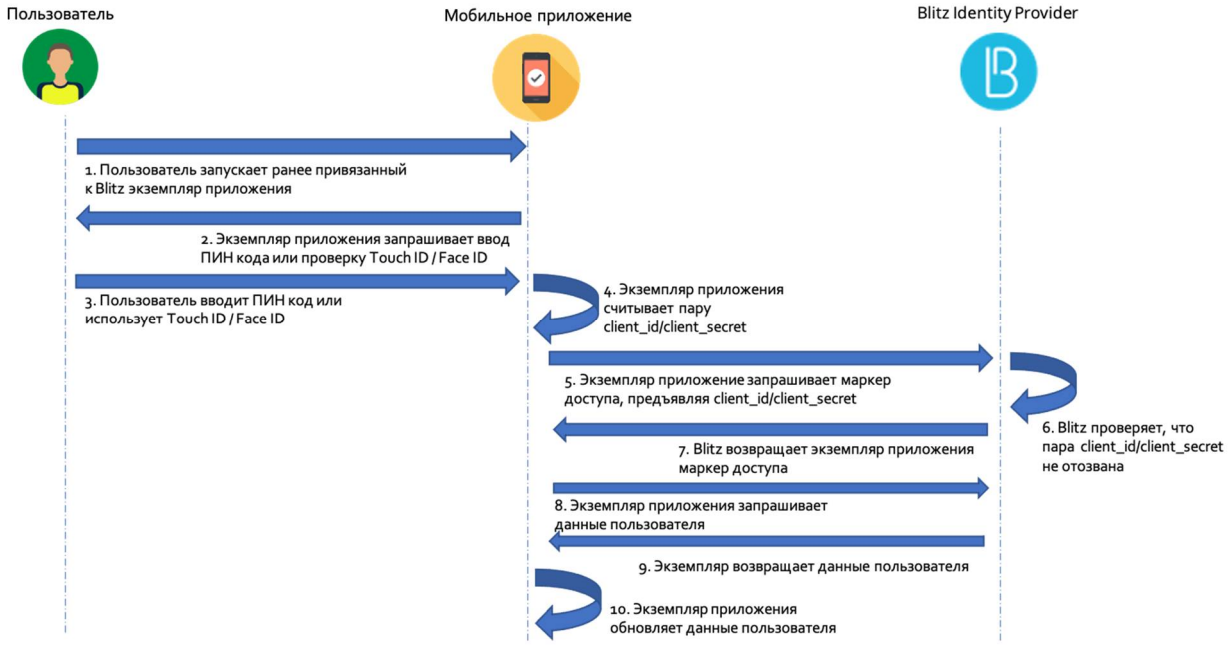

Рисунок 5 – Повторные входы пользователя в мобильное приложение

3.4.2. Динамическая регистрация в Blitz Identity Provider экземпляра мобильного приложения

Предварительные условия для динамической регистрации экземпляра мобильного приложения:

- пользователь должен установить мобильное приложение;
- мобильное приложение должно иметь следующие данные, полученные по результатам обработки заявки на подключение мобильного приложения к Blitz Identity Provider:
  - идентификатор мобильного приложения (*software_id*);
  - первичный маркер доступа (*Initial Access Token*);
  - метаданные мобильного приложения[29] (*software_statement*).

Мобильное приложение должно отправить HTTP-запрос методом POST в Blitz Identity Provider по адресу сервиса динамической регистрации */blitz/oauth/register*. Должны

[29] Метаданные мобильного приложения (*software_statement*) представляют собой JWS-токен, состоящий из трех частей, разделенных точками и закодированных Base64: заголовка, тела, подписи.

Заголовок содержит служебную информацию о типе токена и методе подписи.

Тело токена содержит утверждения, используемые для регистрации экземпляра приложения. В частности, среди утверждений указаны:

- разрешенные префиксы ссылок возврата (*redirect_uris*);
- перечень допустимых разрешений (*scope*).

Подписанный JWS-токен выпускает администратор Blitz Identity Provider в процессе обработки заявки на подключение мобильного приложения к Blitz Identity Provider.

быть переданы параметры:

- идентификатор мобильного приложения (*software_id*);
- метаданные мобильного приложения (*software_statement*);
- тип устройства, на котором работает мобильное приложение (*device_type*) – одно из возможных значений, представленных в таблице 7:

Таблица 7 – Используемые в Blitz Identity Provider маркеры

№	Тип устройства (device_type)	Описание
1.	iphone	Смартфоны семейства iPhone
2.	ipad	Планшеты семейства iPad
3.	android_phone	Смартфоны под управлением ОС Android
4.	android_tab	Планшеты под управлением ОС Android
5.	win_mobile	Устройства под управлением Windows 10 Mobile

Запрос на динамическую регистрацию должен содержать заголовок *Authorization* с первичным маркером доступа (тип – *Bearer*), выданным приложению.

Пример запроса:

```
POST /blitz/oauth/register HTTP/1.1
Content-Type: application/json
Authorization: Bearer NINxnizbgYYQg94vEd6MjkTPxR3r2s9IAHBO92AszgTIqItY
{
      "software_id": "CSI",
      "device_type": "iphone",
      "software_statement": "eyJ0e…xQ"
}
```

При успешном выполнении запроса Blitz Identity Provider возвращает экземпляру мобильного приложения перечень утверждений, среди которых для дальнейшей работы необходимы следующие (их нужно защищенным образом сохранить в устройстве пользователя):

- идентификатор экземпляра мобильного приложения (*client_id*);
- секрет экземпляра мобильного приложения (*client_secret*);
- маркер управления конфигурацией (*registration_access_token*);
- URL управления конфигурацией (*registration_client_uri*).

Пример ответа:

```
{
    "grant_types": [
        "authorization_code"
    ],
    "registration_client_uri": "https://login.company.com/blitz/oauth/register/dyn~CSI~4e6904c5-
ef29-4ae5-8d30-99c359b8270f",
    "scope": "openid profile",
    "registration_access_token": "eyJ0e…tw",
    "client_id": "dyn~CSI~4e6904c5-ef29-4ae5-8d30-99c359b8270f",
    "software_id": "CSI",
    "software_version": "1",
    "token_endpoint_auth_method": "client_secret_basic",
    "response_types": [
        "code"
    ],
    "redirect_uris": [
        "com.example.app:/oauth2redirect/example-provider"
```

```
    ],
    "client_secret": "3r0tt2OlyeGecWq",
    "client_secret_expires_at": 0
}
```

3.4.3. Первичный вход пользователя в мобильное приложение

Получив пару *client_id/client_secret* (пп. 3.4.2) экземпляр мобильного приложения должен провести идентификацию и аутентификацию пользователя согласно спецификациям OIDC/OAuth 2.0 и с учетом дополнительной спецификации RFC 7636[30] (мобильное приложение при взаимодействии с Blitz Identity Provider должно использовать PKCE).

Сценарий идентификация и аутентификации включает следующие шаги:

- запрос на получение кода авторизации;
- получение маркера доступа;
- получение данных пользователя в обмен на маркер доступа.

Первичный вход пользователя в мобильное приложение должен произойти в течение 1 часа с завершения динамической регистрации в Blitz Identity Provider экземпляра мобильного приложения. Иначе *client_id* будет аннулирован и потребуется повторная динамическая регистрация.

3.4.4. Запрос на получение кода авторизации

Для проведения аутентификации экземпляр мобильного приложения должен вызвать штатный браузер мобильной платформы и перенаправить в нем пользователя на URL Blitz Identity Provider сервиса проведения авторизации и аутентификации (*/blitz/oauth/ae*). При использовании браузера мобильным приложением следует учесть следующие особенности:

- для iOS необходимо использовать встроенный браузер: класс *SFSafariViewController* или класс *SFAuthenticationSession* (in-app browser tab pattern);
- для Android необходимо использовать встроенный браузер: функция «Android Custom Tab» (реализует in-app browser tab patter).

> **П р и м е ч а н и е** — использование Embedded-браузера не допускается.

В качестве параметров запроса следует указать:

- *client_id* – идентификатор экземпляра мобильного приложения;
- *response_type* – тип ответа (принимает значение “code”);

[30] RFC 7636 «Proof Key for Code Exchange by OAuth Public Clients», см.: https://tools.ietf.org/html/rfc7636

- *scope* – запрашиваемые разрешения, должно быть передано разрешение *openid* и необходимые дополнительные *scope* для получения данных пользователя (эти *scope* должны быть предусмотрены метаданными);
- *redirect_uri* – ссылка для возврата пользователя в приложение, ссылка должна соответствовать одному из указанных в метаданных значений. Чтобы после авторизации Blitz Identity Provider смог обратно вызвать мобильное приложение, следует использовать следующие схемы:
  - для iOS[31]:
    - вариант 1 – использовать private-use URI scheme («custom URL scheme»). Вид ссылок возврата: *com.example.app:/oauth2redirect/example-provider* (регистрируются в Info.plist ключи типа CFBundleURLTypes);
    - вариант 2 – использовать URI вида «https» («Universal links»). Вид ссылок возврата: *https://app.example.com/oauth2redirect/example-provider* (используется функция «Universal links», URL регистрируются в entitlement-файле в приложении и ассоциированы с доменом приложения). Этот способ предпочтительнее для iOS 9 и выше.
  - для Android[32]:
    - вариант 1 – использовать private-use URI scheme («custom URL scheme»). Вид ссылок возврата: *com.example.app:/oauth2redirect/example-provider* (поддержка ссылок с помощью Android Implicit Intents, ссылки регистрируются в manifest);
    - вариант 2 – использовать URI вида «https» («Universal links»). Вид ссылок возврата: *https://app.example.com/oauth2redirect/example-provider* (доступно начиная с Android 6.0, ссылки регистрируются в manifest). Этот способ предпочтительнее для Android 6.0 и выше.
- *state* – набор случайных символов, имеющий вид 128-битного идентификатора запроса (используется для защиты от перехвата), это же значение будет возвращено в ответе – опциональный параметр;
- *access_type* (необязательный параметр) – требуется ли приложению получать refresh_token, необходимый для получения сведений о пользователе в дальнейшем, когда пользователь будет оффлайн. Принимает значение “online”/“offline”, refresh_token предоставляется при access_type=offline. Если значение не задано, то

[31] Пример реализации – см.: https://github.com/openid/AppAuth-iOS

[32] Пример реализации – см.: https://github.com/openid/AppAuth-Android

поведение определяется настройкой, заданной для указанного приложения в Blitz Identity Provider;

- *code_challenge_method* – метод шифрования идентификатора запроса, следует указывать “S256”;
- *code_challenge* – зашифрованный идентификатор запроса. Идентификатор запроса (*code_verifier*) должен быть запомнен экземпляром мобильного приложения для последующей передачи в запрос на получение маркера доступа. Шифрованное значение вычисляется следующим образом:

```
code_challenge=BASE64URL-ENCODE(SHA256(ASCII(code_verifier)))
```

Пример запроса на получение кода авторизации (запрошена аутентификация и маркер доступа с разрешениями *openid* и *profile*, используется PKCE):

```
https://login.company.com/blitz/oauth/ae?scope=openid+profile
&access_type=online&response_type=code
&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f
&client_id=dyn~CSI~4e6904c5-ef29-4ae5-8d30-99c359b8270f
&code_challenge_method=S256&code_challenge=qjrzSW9gMiUgpUvqgEPE4
&redirect_uri=https%3A%2F%2Fapp.example.com%2Foauth2redirect%2Fexample-provider
```

Пример ответа со значением кода авторизации (*code*) и параметром *state*:

```
https://app.example.com/oauth2redirect/example-
provider?code=f954nEzQ08DXju4wxGbSSfCX7TkZlGvXUR7TzVus8fG
nu4AUl-YIosgax-BLXMeQQAlasD6CN2qG_0KXK5NIjARoKykhuR9IpbuzqeFxS0
&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f
```

Возможные ошибки при вызове */oauth/ae* соответствуют RFC 6749 и описаны по ссылке[33].

3.4.5. Получение маркеров

После получения кода авторизации экземпляр мобильного приложения должен обменять его на маркеры. Для этого экземпляр должен сформировать запрос методом POST на URL для получения маркера. Запрос должен содержать заголовок *Authorization* со значением *Basic {secret}*, где *secret* – это *client_id:client_secret* (например, *dyn~CSI~4e69…Wq*) в формате Base64.

Пример заголовка:

```
Authorization: Basic ZHluOkNTSTo…dx
```

Тело запроса должно содержать следующие параметры:

- *code* – значение кода авторизации, который был ранее получен экземпляром мобильного приложения от Blitz Identity Provider;
- *grant_type* – значение *authorization_code*;
- *redirect_uri* – должно быть то же самое значение, которое было указано в запросе на получение кода авторизации;

[33] См.: https://tools.ietf.org/html/rfc6749#section-4.1.2.1

– *code_verifier* – идентификатор запроса, сгенерированный экземпляром мобильного приложения при запросе на получение кода авторизации.

Пример запроса:

```
POST /blitz/oauth/te HTTP/1.1
Authorization: Basic ZHluOkNTSTo…dx
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

grant_type=authorization_code&code=FLZHS…GU
&redirect_uri=https%3A%2F%2Fapp.example.com%2Foauth2redirect%2Fexample-provider
&code_verifier=M25iVXpKU3puUjFaYWg3T1NDTDQtcW1ROUY5YXlwalNoc0hhakxifmZHag
```

В ответ возвращается маркер доступа и маркер идентификации.

Пример ответа с успешным выполнением запроса:

```
{
    "id_token": "eyJhb…J9. eyJub…0=.Ckt_dr…sQ",
    "access_token": "dO-xym…BE",
    "expires_in": 3600,
    "scope": "openid profile",
    "token_type": "Bearer"
}
```

После получения маркера доступа экземпляр мобильного приложения становится связанным с учетной записью пользователя. Рекомендуется, чтобы мобильное приложение предложило пользователю установить ПИН код или включить Touch ID/Face ID.

Также с помощью полученного маркера доступа приложение может запросить данные о пользователе (пп. 3.6).

Если код авторизации был уже использован, не совпал *redirect_uri* с ранее использованным в вызове к */oauth/ae*, или истек срок действия кода, либо переданный *code_verifier* не соответствует *code_challenge*, то в качестве ответа будет возвращена ошибка.

Пример ответа с ошибкой:

```
{
    "error": "invalid_grant",
    "error_description": "The provided authorization grant… is invalid, expired, revoked…"
}
```

Возможные ошибки при вызове */oauth/te* соответствуют RFC 6749 и описаны по ссылке[34].

3.4.6. Повторный вход пользователя в мобильное приложение

При каждом входе пользователя в экземпляр мобильного приложения, если с устройства доступен выход в сеть Интернет, следует производить аутентификацию пользователя посредством вызова сервиса Blitz Identity Provider. В частности, при каждом входе в экземпляр мобильного приложения необходимо проверить ПИН-код пользователя или Touch ID/Face ID, после чего извлечь защищенно хранимые на устройстве *client_id/client_secret* и сделать запрос в Blitz Identity Provider на проведение повторного

[34] См.: https://tools.ietf.org/html/rfc6749#section-5.2

входа пользователя. Использовать полученный в ответ от Blitz Identity Provider маркер доступа для получения актуальных данных пользователя.

Запрос в Blitz Identity Provider на проведение повторного входа должен быть выполнен методом POST на URL для получения маркера (*/oauth/te*). Запрос должен содержать заголовок *Authorization* со значением *Basic {secret}*, где *secret* – это *client_id:client_secret* экземпляра мобильного приложения в формате Base64.

Тело запроса должно содержать параметры:

- *grant_type* – значение *client_credentials*;
- *scope* – перечень запрашиваемых экземпляром мобильного приложения разрешений.

Пример запроса:

```
POST /blitz/oauth/te HTTP/1.1
Authorization: Basic cG9ydGFsLmlhc2l1LmxvY2FsOnBvcnRhbC5pYXNpdS5sb2NhbA==
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

grant_type=client_credentials&scope=profile
```

В ответ возвращается маркер доступа и информация об этом маркере.

Пример ответа с успешным выполнением запроса:

```
{
    "access_token": "dO-xym…BE",
    "expires_in": 3600,
    "scope": "openid profile",
    "token_type": "Bearer"
}
```

Используя полученный маркер доступа, экземпляр мобильного приложения может запросить (пп. 3.6) актуальные данные пользователя из Blitz Identity Provider, чтобы при необходимости визуализировать или обновить эти данные в устройстве.

Если пользователь в Blitz Identity Provider отозвал у экземпляра мобильного приложения право авторизации в Blitz Identity Provider, то в результате вызова Blitz Identity Provider экземпляр мобильного приложения получит ошибку.

Пример ответа с ошибкой:

```
{
    "error": "invalid_client",
    "error_description": " Client authentication failed…"
}
```

Возможные ошибки при вызове */oauth/te* соответствуют RFC 6749 и описаны по ссылке[35].

3.4.7. Переключение или выход пользователя из мобильного приложения

Если в мобильном приложении предусмотрена функция выхода или смены пользователя, то при вызове пользователем такой функции мобильное приложение должно также вызвать Blitz Identity Provider и удалить выпущенную для данного экземпляра

[35] См.: https://tools.ietf.org/html/rfc6749#section-5.2

мобильного приложения пару *client_id/client_secret*. Если это не будет сделано, то при выходе пользователя из мобильного приложения, пользователь в веб-приложении Blitz Identity Provider «Настройки безопасности[36]» все равно будет видеть, что мобильное приложение все еще привязано к его учетной записи.

Чтобы удалить из Blitz Identity Provider выпущенную для экземпляра мобильного приложения пару *client_id/client_secret*, мобильное приложение должно отправить в Blitz Identity Provider запрос методом DELETE на URL управления конфигурацией (*registration_client_uri*), полученный и запомненный мобильным приложением при вызове динамической регистрации в Blitz Identity Provider экземпляра мобильного приложения (пп. 3.4.2). Запрос должен содержать заголовок *Authorization* со значением *Bearer {registration_access_token}*, где *registration_access_token* – это маркер управления конфигурацией, также полученный и запомненный в процессе динамической регистрации. Запрос не требует указания параметров.

Пример запроса:

```
DELETE /blitz/oauth/register/dyn~CSI~4e6904c5-ef29-4ae5-8d30-99c359b8270f HTTP/1.1
Authorization: Bearer eyJ0e…tw
```

Если после удаления пары *client_id/client_secret* мобильное приложение сразу запросит получение новой пары *client_id/client_secret*, и запросит вход пользователя, то если предыдущий вход выполнялся в этой же браузерной сессии, то сработает SSO и пользователь автоматически войдет прежним аккаунтом. Обычно это нежелательное поведение для входа сразу после выхода, так как ожидается, что пользователь захочет войти под другим аккаунтом. Поэтому после выхода рекомендуется запрашивать новый вход одним из следующих способов:

- При запросе кода авторизации указывать в запросе дополнительный параметр *prompt=login*. Тогда Blitz Identity Provider предложит текущему пользователю пройти аутентификацию, даже если активна Blitz Identity Provider сессия. Также пользователь может на странице входа выбрать *«Сменить аккаунт»*, чтобы войти под другой учетной записью.
- При запросе кода авторизации указать в запросе дополнительный параметр *prompt=select_account*. Так Blitz Identity Provider сразу предложит пользователю выбрать аккаунт из числа запомненных или войти новым аккаунтом. Пользователю не придется дополнительно нажимать кнопку *«Сменить аккаунт»* на странице входа.

[36] Стандартный адрес имеет вид: https://login.company.com/blitz/profile

3.4.8. Открытие веб-ресурсов из мобильного приложения в режиме сквозной аутентификации

В некоторых мобильных приложениях разработчикам может потребоваться предусмотреть функцию открытия веб-ресурсов, также требующих идентификации/аутентификации пользователя, и использующих для этой цели Blitz Identity Provider. При доступе к веб-ресурсу пользователь, вошедший в мобильное приложение, может столкнуться с ситуацией, что Blitz Identity Provider повторно потребует у него пройти идентификацию/аутентификацию в веб-ресурсе в результате запроса соответствующим веб-приложением идентификации/аутентификации пользователя в Blitz Identity Provider. Чтобы такого не произошло, мобильное приложение может непосредственно перед вызовом веб-ресурса запросить в Blitz Identity Provider получение маркера доступа (*access_token*) на специальное разрешение (*scope*) с именем *native*. Получить маркер доступа можно способом, описанным в пп. 3.4.6 или в пп. 3.3.3 (при наличии у приложения *refresh_token*).

Пример запроса:

```
POST /blitz/oauth/te HTTP/1.1
Authorization: Basic cG9ydGFsLmlhc2l1LmxvY2FsOnBvcnRhbC5pYXNpdS5sb2NhbA==
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

grant_type=client_credentials&scope=native
```

В ответ возвращается не только маркер доступа и информация об этом маркере, но и специальный атрибут – маркер сквозного входа *css* (*cookie short session*).

Пример ответа с получением атрибута *css*:

```
{
    "access_token": "dO-xym…BE",
    "css": "nUngQ…LA",
    "expires_in": 3600,
    "scope": "native",
    "token_type": "Bearer"
}
```

После этого мобильное приложение может открывать веб-ресурс. При этом в запускаемом веб-браузере мобильное приложение должно предварительно установить cookie со следующими параметрами:

- имя cookie – *css*;
- домен cookie – *login.company.com*;
- путь (path) cookie – */blitz*;
- флаги *HTTPOnly=true* и *Secure=true*;
- значение cookie – значение, полученное в параметре *css* при получении от Blitz Identity Provider маркера доступа на scope с именем *native*.

Если запущенный веб-ресурс в течение 300 секунд с момента запуска инициирует в Blitz Identity Provider идентификацию/аутентификацию, и cookie была корректно

установлена, то Blitz Identity Provider по запросу веб-приложения проведет автоматическую сквозную идентификацию и аутентификацию пользователя под учетной записью, с которой пользователь ранее входил в экземпляр мобильного приложения, вызвавшего веб-ресурс.

3.4.9. Добавление в мобильное приложение функции входа по QR-коду

Вход по QR-коду может использоваться в Blitz Identity Provider как первый фактор аутентификации (альтернатива вводу логина/пароля). При выборе этого способа входа Blitz Identity Provider формирует и отображает пользователю QR-код, в котором закодирован запрос на вход (Рисунок 6). Срок действия QR-кода ограничен, а сформированный запрос является одноразовым. По истечение срока действия отображенного QR-кода пользователю предоставляется возможность запросить отображение нового QR-кода.

Закодированная в QR-коде ссылка имеет вид: *QR_URL?code=b0671081-cb73-4839-8bc1-8cf020457228*, например:

```
https://login.company.com/blitz/login/qr?code=b0671081-cb73-4839-8bc1-8cf020457228
```

Значение QR_URL может быть настроено таким образом, чтобы в случае наведения смартфона на QR-код с использованием стандартного приложения камеры пользователю могла быть отображена веб-страница с инструкцией по получению правильного мобильного приложения для загрузки QR-кодов или возможность вызова подходящего мобильного приложения через Universal Link.

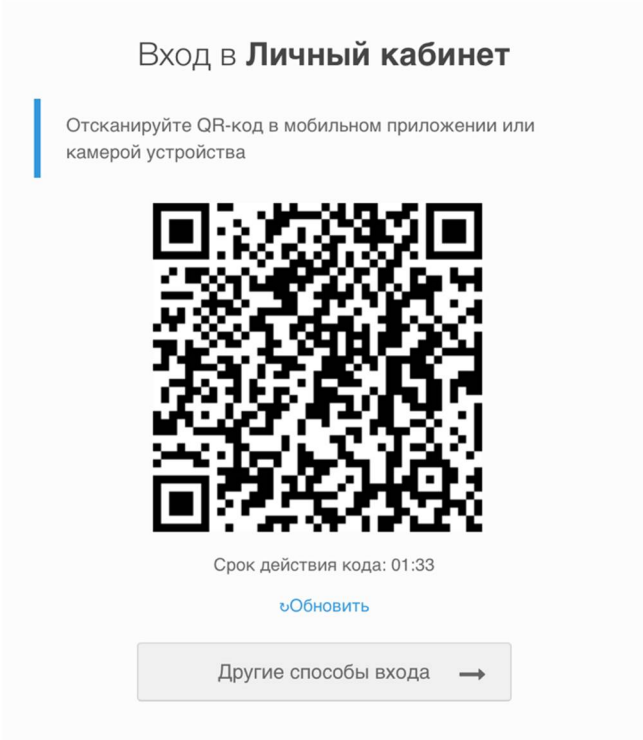

Рисунок 6 – Страница входа с отображением QR-кода

Процесс входа по QR-коду на стороне мобильного приложения состоит из следующих шагов:

1. Перед фотографированием QR-кода мобильным приложением пользователь должен

быть залогинен в мобильное приложение с использованием Blitz Identity Provider, и мобильное приложение должно получить в Blitz Identity Provider действующий маркер доступа со scope с именем *blitz_qr_auth* (разрешение на проведение входа с использованием QR-кода).

2. При фотографировании QR-кода мобильное приложение должно отбросить значение QR_URL (оно не нужно приложению и должно быть проигнорировано) и приложение должно считать значение переданного в ссылке параметра *code*.
3. После считывания QR-кода мобильное приложение должно вызвать в Blitz Identity Provider сервис получения сведений о запросе входа, передав в сервис значение полученного кода, а также заголовок с маркером доступа и заголовок текущего языка пользователя.

   Пример вызова:

```
curl --location --request GET 'https://login.company.com/blitz/api/v3/auth/qr/b0671081-cb73-4839-
8bc1-8cf020457228' \
--header 'Content-Type: application/json' \
--header 'Accept-Language: ru' \
--header 'Authorization: Bearer eyJhb…tA'
```

В ответ вернется JSON, содержащий информацию об IP-адресе, операционной системе и браузере устройства, на котором пользователь пытается войти с использованием входа по QR-коду, а также имя приложения, в которое пользователь пытается войти.

Пример успешного ответа:

```
{
    "ip": "83.220.238.103",
    "rp_name": "User profile",
    "ip_city": "Москва",
    "browser": "Chrome 109",
    "ip_state": "Москва",
    "os": "macOS 10.15.7",
    "ip_lng": "37.6171",
    "device_type": "pc",
    "ip_lat": "55.7483",
    "ip_country": "Россия",
    "rp_id": "_blitz_profile",
    "device_name": "macOS Big Sur (11)",
    "ip_radius": "20",
    "device": "PC"
}
```

Также пользователю в веб-странице будет показан экран, что ожидается подтверждение входа.

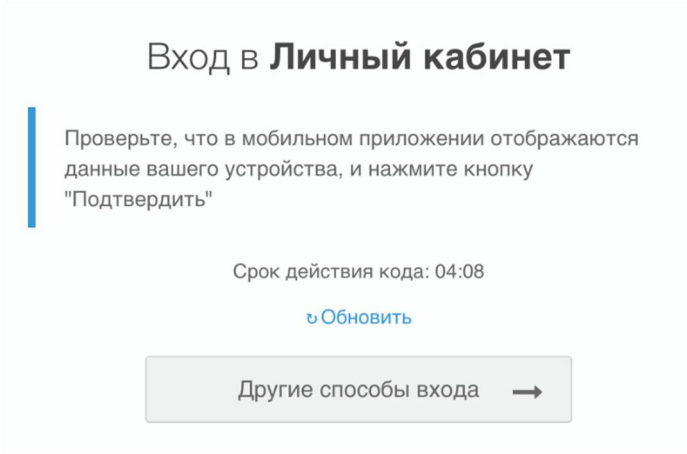

Рисунок 7 – Страница запроса подтверждения входа в мобильном приложении

Пользователю в мобильном приложении нужно отобразить имя приложения (*rp_name*), IP-адрес (*ip*), геоданные (*ip_country*, *ip_state*, *ip_city* – текстовое описание адреса или показать на карте по координатам *ip_lat*, *ip_lng*), используемое устройство (*device_name*), браузер (*browser*).

Возможные значения *device_type* сейчас: *kindle, mobile, tablet, iphone, windowsPhone, pc, ipad, playStation, unknown*. Можно их использовать в визуализации сообщения или можно просто вывести имя устройства текстовой строкой из device.

Пример ответа при недействительном маркере доступа:

```
{
    "type": "security_error",
    "error": "bad_access_token",
    "desc": "expired_access_token"
}
```

Пример ответа при просроченном QR-коде:

```
{
    "type": "process_error",
    "error": "qr_session_expired",
    "desc": "Error while getting QR authentication session"
}
```

Пример ответа при несуществующем коде:

```
{
    "params": {},
    "desc": "Error while getting QR authentication session",
    "error": "qr_session_not_found"
}
```

Пример ответа при вызове по уже использованной QR-сессии (когда уже подтвердили или уже отклонили вход):

```
{
    "type": "process_error",
    "error": "qr_session_already_completed",
    "desc": "Error while getting QR authentication session"
}
```

1. Мобильное приложение должно отобразить пользователю полученные из JSON от Blitz Identity Provider сведения о входе, а также выбор действия: «Разрешить» или «Отклонить». В случае «Отклонить» запросить причину отклонения («Вход вызван по ошибке» или «Я не запрашивал вход»).
2. В зависимости от решения пользователя мобильное приложение должно вызвать в

Blitz Identity Provider сервис подтверждения или отказа входа. При вызове должен использоваться маркер доступа со scope с именем *blitz_qr_auth*.

Пример вызова при подтверждении входа:

```
curl --location --request POST 'https://login.company.com/blitz/api/v3/auth/qr/5e20b01e-5c7c-
4101-8292-98e6865c7bfb/confirm' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer eyJhb…cQ'
```

Если успешно, то вернется HTTP 204 No Content без body. Также пользователь войдет в приложение.

Если код просрочен, то вернется:

```
{
    "type": "process_error",
    "error": "qr_session_expired",
    "desc": "Error while confirming QR authentication session"
}
```

Если код не существует, то вернется:

```
{
    "params": {},
    "desc": "Error while confirming QR authentication session",
    "error": "qr_session_not_found"
}
```

Пример ответа при недействительном маркере доступа:

```
{
    "type": "security_error",
    "error": "bad_access_token",
    "desc": "expired_access_token"
}
```

Пример ответа при вызове по уже использованной QR-сессии (когда уже подтвердили или уже отклонили вход):

```
{
    "type": "process_error",
    "error": "qr_session_already_completed",
    "desc": "Error while getting QR authentication session"
}
```

Пример вызова при отклонении входа:

```
curl --location --request POST 'https://login.company.com/blitz/api/v3/auth/qr/845f2334-fa6b-
40c0-9a71-f57997166e39/refuse' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer eyJhb…bQ' \
--data-raw '{
"cause_id": "mistake",
"desc": "Вход вызван по ошибке"
}'
```

При отклонении входа нужно обязательно передавать в теле запроса JSON с атрибутом *cause_id*. Рекомендуется при отклонении входа пользователем спросить причину. Если пользователь сообщит, что «передумал» (или «вызвал вход по ошибке»), то заполнить *cause_id=mistake*. Но если пользователь сообщит, что он не инициировал вход, то заполнить *cause_id=unauthorized*. Параметр *desc* опционален – можно указать любую текстовую строку.

В случае успешного вызова вернется *HTTP 204 No Content* без body. Также пользователю будет показан экран с ошибкой:

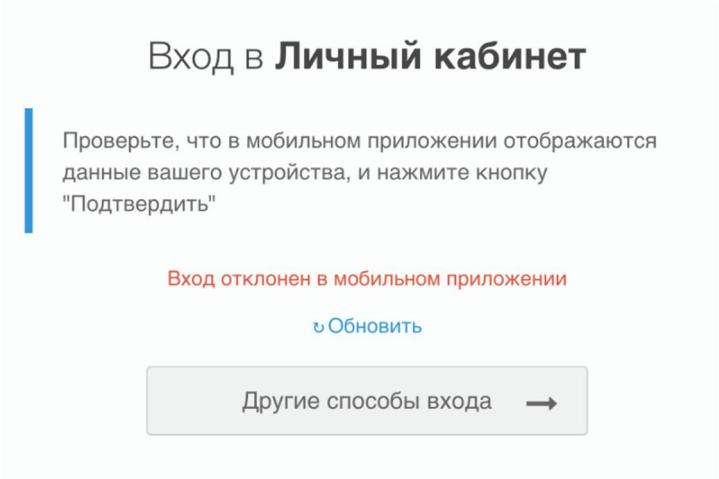

Рисунок 8 – Страница с ошибкой, что вход отклонен в мобильном приложении

В случае если код просрочен, то вернется ошибка:

```
{
    "type": "process_error",
    "error": "qr_session_expired",
    "desc": "Error while refusing QR authentication session"
}
```

Если код не существует, то вернется:

```
{
    "params": {},
    "desc": "Error while refusing QR authentication session",
    "error": "qr_session_not_found"
}
```

Пример ответа при недействительном маркере доступа:

```
{
    "type": "security_error",
    "error": "bad_access_token",
    "desc": "expired_access_token"
}
```

Пример ответа при вызове по уже использованной QR-сессии (когда уже подтвердили или уже отклонили вход):

```
{
    "type": "process_error",
    "error": "qr_session_already_completed",
    "desc": "Error while getting QR authentication session"
}
```

3.5. Подключение приложений умных устройств (IoT)

В Blitz Identity Provider реализована поддержка возможности авторизации приложений умных устройств (приложений голосовых помощников, смарт ТВ, чат-ботов) с использованием учетной записи пользователя на другом устройстве. Для такой авторизации используется спецификация RFC 8628 «OAuth 2.0 Device Authorization Grant»[37].

Для инициирования авторизации приложение умного устройства должно сделать запрос в адрес Blitz Identity Provider на сервис получения кода подтверждения авторизации

[37] См.: https://www.ietf.org/rfc/rfc8628.html

(*/oauth/da*). Запрос должен быть сделан методом POST. Запрос должен содержать заголовок *Authorization* со значением *Basic {secret}*, где *secret* – это *client_id:client_secret* (например, *app:topsecret*) в формате Base64.

Пример заголовка:

```
Authorization: Basic ZHluOkNTSTo…dx
```

Тело запроса должно содержать следующие параметры:

- *client_id* – идентификатор приложения;
- *scope* – запрашиваемые разрешения.

Пример запроса:

```
POST /blitz/oauth/da HTTP/1.1
Authorization: Basic ZHluOkNTSTo…dx
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

client_id=test-app&scope=profile
```

В ответ Blitz Identity Provider вернет данные, необходимые для подтверждения входа на другом устройстве:

- *device_code* – код устройства;
- *user_code* – отображаемый пользователю код подтверждения запроса авторизации;
- *verification_uri* – ссылка на страницу, на которой пользователь может ввести код подтверждения запроса авторизации;
- *verification_uri_complete* – ссылка на страницу, в которой в качестве параметра уже подставлен код подтверждения запроса авторизации;
- *expires_in* – время жизни пользовательского кода в секундах;
- *interval* – рекомендуемый период ожидания в секундах при опрашивании приложением ввода пользователем кода подтверждения запроса авторизации.

Пример ответа с успешным выполнением запроса:

```
{
    "device_code": "7Lz30lK57bWaKHBYxM8kW7KpOFvDg_4ujz3LpQxcleE",
    "user_code": "934-367-578",
    "verification_uri": "https://device.company.com",
    "verification_uri_complete": "https://device.company.com?uc=934-367-578",
    "expires_in": 300,
    "interval": 5
}
```

Получив ответ приложение умного устройства должно инструктировать пользователя, чтобы он перешел по ссылке *verification_uri*[38] и ввел код из *user_code*. В

[38] Ссылка в *verification_uri* выводится в соответствии с настройками, заданными в Blitz Identity Provider. Рекомендуется настроить, чтобы эта ссылка была короткой и удобной для ввода пользователям, а также хорошо воспринималась на слух или красиво отображалась на экране Смарт ТВ. С данной ссылки должна быть настроена переадресация на обработчик ввода пользователем кода подтверждения, расположенный на странице *https://login.company.com/blitz/oauth/device?ci=client_id*, где вместо *client_id* нужно задать идентификатор зарегистрированного в Blitz Identity Provider приложения, из настроек которого будут браться разрешенные способы входа и настройки внешнего вида страницы входа.

зависимости от типа умного устройства нужно выбрать наиболее удобный для пользователя способ. Например:

- При авторизации в Смарт ТВ приложение может отрисовать пользователю QR-код, в котором закодировать ссылку из *verification_uri_complete*. Тогда пользователю нужно будет навести камеру телефона на QR-код и пройти авторизацию на телефоне.
- При авторизации в чат-боте приложение может отрисовать пользователю кнопку, открывающую в браузере ссылку из *verification_uri_complete.* Тогда пользователю нужно будет пройти авторизацию в браузере своего устройства.
- При авторизации в приложении голосового помощника приложение может проинструктировать пользователя, на какой сайт он должен перейти, и озвучить код, который пользователь должен ввести, либо приложение может отправить пользователю SMS-сообщение или письмо по электронной почте с инструкцией.

После предоставления пользователю инструкций приложение умного устройства должно с интервалом из параметра *interval* начать осуществлять опрос Blitz Identity Provider для получения маркеров безопасности. Для этого приложение должно обращаться в Blitz Identity Provider методом POST на URL для получения маркера (*/oauth/te*). Запрос должен содержать заголовок *Authorization* со значением *Basic {secret}*, где *secret* – это *client_id:client_secret* экземпляра мобильного приложения в формате Base64.

Тело запроса должно содержать параметры:

- *grant_type* – значение *urn:ietf:params:oauth:grant-type:device_code*;
- *device_code* – ранее полученный код устройства.

Пример запроса:

```
POST /blitz/oauth/te HTTP/1.1
Authorization: Basic cG9…A==
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

grant_type=urn:ietf:params:oauth:grant-type:device_code&device_code=Yrn…_0
```

Если пользователь еще не подтвердил авторизацию, то Blitz Identity Provider вернет следующий ответ с ошибкой:

```
{
    "error": "authorization_pending",
    "error_description": "The authorization request is still pending"
}
```

Если срок действия пользовательского кода истек или код неправильный, то Blitz Identity Provider вернет следующий ответ с ошибкой:

```
{
    "error": "invalid_grant",
    "error_description": "The provided authorization grant (e.g., authorization code, resource
owner credentials) or refresh token is invalid, expired, revoked, does not match the redirection
URI used in the authorization request, or was issued to another client."
}
```

Если пользователь подтвердил авторизацию, то Blitz Identity Provider вернет приложению маркер доступа и информацию о нем, а также маркер обновления.

Пример ответа с успешным выполнением запроса:

```
{
    "access_token": "eyJ…tA",
    "refresh_token": "wVE…cw",
    "scope": "profile",
    "token_type": "Bearer",
    "expires_in": 3600
}
```

Используя полученный маркер доступа, приложение умного устройства может запросить (пп. 3.6) актуальные данные пользователя из Blitz Identity Provider.

3.6. Получение атрибутов пользователя

Для запроса данных о пользователе необходимо выполнить запрос методом GET по URL-адресу получения данных пользователя (*/oauth/me*). В запрос должен быть добавлен следующий заголовок:

```
Authorization: Bearer <access token>
```

В заголовке *<access token>* – это маркер доступа, полученный от Blitz Identity Provider (пп. 3.3.3 и 3.4.5).

Пример запроса:

```
GET /blitz/oauth/me HTTP/1.1
Authorization: Bearer NINxn…tY
Cache-Control: no-cache
```

В ответе будут отображены только те данные, которые определены в scope, на который получен маркер доступа (пп. 2.1.3).

Пример ответа:

```
{
    "family_name": "Иванов",
    "given_name": "Иван",
    "middle_name": "Иванович",
    "email": "iivanov@company.com",
    "phone_number": "79162628910",
    "sub": "3d10f626-ea77-481d-a50bd4a4d432d86b"
}
```

Учетная запись пользователя может быть включена в группы пользователей. Чтобы получить список групп, в которые включен пользователь, маркер доступа должен быть получен с scope с именем *usr_grps*.

Пример ответа по пользователю, включенному в группы доступа:

```
{
    "family_name": "Иванов",
    "given_name": "Иван",
    "middle_name": "Иванович",
    "email": "iivanov@company.com",
    "phone_number": "79162628910",
    "sub": "3d10f626-ea77-481d-a50bd4a4d432d86b",
    "groups": [
        {
            "id": "564486ff-af0a-3fb1-3f09-e7c5f7f9833e",
            "name": "Тестовая организация",
            "OGRN": "1234567890123",
            "INN": "9876543210"
        }
```

```
    ]
}
```

4. Подключение приложений по SAML

4.1. Данные для подключения по результатам рассмотрения заявки

По результатам исполнения заявки на подключение к Blitz Identity Provider будет предоставлена следующая информация:

– идентификатор, присвоенный приложению в Blitz Identity Provider (*entityID*);

– уточненный файл метаданных поставщика услуг[39].

Все URL в данном разделе необходимо скорректировать в зависимости от конкретных настроек, сделанных при внедрении Blitz Identity Provider. См. «Руководство администратора».

Приложение взаимодействует с сервисами Blitz Identity Provider, используя следующие адреса:

– метаданные Blitz Identity Provider:

- *https://login-test.company.com/blitz/saml/profile/Metadata/SAML* (тестовая среда)
- *https://login.company.com/blitz/saml/profile/Metadata/SAML* (продуктивная среда)

– URL для аутентификации:

- *https://login-test.company.com/blitz/saml/profile/SAML2/Redirect/SSO* (тестовая среда)
- *https://login.company.com/blitz/saml/profile/SAML2/Redirect/SSO* (продуктивная среда)

– URL для логаута:

- *https://login-test.company.com/blitz/saml/profile/SAML2/Redirect/SLO* (тестовая среда)
- *https://login.company.com/blitz/saml/profile/SAML2/Redirect/SLO* (продуктивная среда)

– URL издателя:

- *https://login-test.company.com/blitz/saml/* (тестовая среда)
- *https://login.company.com/blitz/saml/* (продуктивная среда)

Если приложение поддерживает протокол подключения SAML, то указанных данных должно быть достаточно для конфигурирования приложения. Если приложение не поддерживает протокол SAML, следует произвести его доработку согласно рекомендациям, изложенным в пп. 4.2 и пп. 4.3 настоящего документа.

[39] Как правило, этот файл будет совпадать с тем, что был приложен к заявке. Но Администратор Blitz Identity Provider может скорректировать файл метаданных и прислать его вместе с результатами рассмотрения заявки.

Типичные вопросы о том, как настроить приложение для подключения к Blitz Identity Provider по протоколу SAML, приведены в таблице ***8***.

Таблица 8 — Типичные вопросы при подключении приложения по протоколу SAML

Вопрос	Ответ
Где найти метаданные поставщика идентификации?	Чтобы загрузить метаданные, пройдите по ссылке *https://login.company.com/blitz/saml/profile/Metadata/SAML* и скопируйте открытый XML документ в приложение
Где найти сертификат SAML поставщика идентификации?	Откройте XML документ с метаданными поставщика идентификации. Найдите раздел `<ds:X509Certificate></ds:X509Certificate>` – в нем и располагается сертификат SAML поставщика идентификации. Пример:  <EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata" xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:shibmd="urn:mace:shibboleth:metadata:1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" entityID="https://sudir.mos.ru/blitz/saml"> <IDPSSODescriptor protocolSupportEnumeration="urn:mace:shibboleth:1.0 urn:oasis:names:tc:SAML:1.1:protocol urn:oasis:names:tc:SAML:2.0:protocol"> <Extensions> <shibmd:Scope regexp="false">0.1</shibmd:Scope> </Extensions> <KeyDescriptor> <ds:KeyInfo> <ds:X509Data> <ds:X509Certificate> MIIDDzCCAfegAwIBAgIJANjxtiKgDpaeMA0GCSqGSIb3DQEBBQUAMBcxFTATBgNV BAMTDHN1ZGlyLm1vcy5ydTAeFw0xODA2MjAxNjQ2MDZaFw0yODA2MTcxNjQ2MDZa MBcxFTATBgNVBAMTDHN1ZGlyLm1vcy5ydTCCASIwDQYJKoZIhvcNAQEBBQADggEP ADCCAQoCggEBANk5Ue/3dmNTLdTzKNrgKLM7lpdnBFNJNjDkkkBF2GodQ+r+ePLz thw5Gn9G4uLmwFol13fU6usbEdi2IDzg3M5slT8YbCxzvaw7ddNU9Jdh1YAqIrXT VvtRCajyZk3AwraXNj1Ai9Qq8XuXSlEtlymvdUAeY1SScKDpNYIM8cqdHmvSXKvx FggJn+S1l6MEDv/0quM2MvOhgLuP7i6J8wNXD4P4fz8+oNGPcqLwn90fIGgFyPBE nQ2vmEn0NRotwQCnYcIApEq9jMBGiMi2yQtIsjFYDjjdqBqau/cXuVybiYA8om3W cyMIDFdcJ2RAAHtzNdXN8xnnv8IMrqRqG/MCAwEAAaNeMFwwOwYDVR0RBDQwMoIw c3VkaXIubW9zLnJ1O1VSSTpodHRwczovL3N1ZGlyLm1vcy5ydS9ibGl0ei9zYW1s MB0GA1UdDgQWBBRw3ACqmoCP31aMlW/KtwFsQLZ7iDANBgkqhkiG9w0BAQUFAAOC AQEAJ72xDGx37QBdHIyDiOhwe1Kxibvwm5DZxQ6Sc6YTS6fnCWdJeUlLJ82yKOIW Hwfnre+nRRuAHLA9DhaZIYmBvUuqE1tBYadwqiKS0l518khE509jnMyizWMiwRPK IUz730BQUDl3zsT+WvO21Xced8PKR73Y2XZCnIybDbYNipy1ST9V0/bkB1S6VR8x OOiOr89rgY/1EWXRnQn+9Wm2tQZXbdCTHOBg7kCg4M4OnqyOi1rfUvoHboeVrLUA ap/b+fHRdL2p08qCJOSCRhPwETuyYolqt3DSYJqqTDui1Tyg8i61j65xL0lJER9J 48L3KzS5SY/DUHYmfLfddIRb/Q== </ds:X509Certificate> </ds:X509Data> </ds:KeyInfo> </KeyDescriptor> <ArtifactResolutionService Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding" Location="https://sudir.mos.ru/blitz/saml/profile/SAML1/SOAP/ArtifactResolution" index="1"/> <ArtifactResolutionService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP" Location="https://sudir.mos.ru/blitz/saml/profile/SAML2/SOAP/ArtifactResolution" index="2"/> <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect" Location="https://sudir.mos.ru/blitz/saml/profile/SAML2/Redirect/SLO" ResponseLocation="https://sudir.mos.ru/saml/profile/SAML2/Redirect/SLO"/> <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Plain-Redirect" Location="https://sudir.mos.ru/blitz/saml/profile/SAML2/Redirect/Plain/SLO" ResponseLocation="https://sudir.mos.ru/saml/profile/SAML2/Redirect/Plain/SLO"/> <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP" Location="https://sudir.mos.ru/blitz/saml/profile/SAML2/SOAP/SLO"/> <NameIDFormat>urn:mace:shibboleth:1.0:nameIdentifier</NameIDFormat> Иногда для корректной загрузки в приложение перед строкой с сертификатом нужно вставить строку `-----BEGIN CERTIFICATE-----` , а после – `-----END CERTIFICATE-----`
Где найти адреса SAML-обработчиков поставщика идентификации?	Запросы на идентификацию/аутентификацию приложение должно отправлять на следующие обработчики (*SingleSignOnService*) в ПРОД-среде: – *https://login.company.com/blitz/saml/profile/SAML2/Redirect/SSO* – для приема сжатых с помощью алгоритма Deflate запросов – стандартный SAML-обработчик.

Вопрос	Ответ
	– *https://login.company.com/blitz/saml/profile/SAML2/Redirect/Plain/SSO* – для приема несжатых запросов – следует использовать только в случае, если подключаемое приложение не использует deflate. Запросы на единый логаут приложение должно отправлять на следующие обработчики (*SingleLogoutService*) в ПРОД-среде: – *https://login.company.com/blitz/saml/profile/SAML2/Redirect/SLO* – для приема сжатых с помощью алгоритма Deflate запросов – стандартный SAML-обработчик. – *https://login.company.com/blitz/saml/profile/SAML2/Redirect/Plain/SLO* – для приема несжатых запросов – следует использовать только в случае, если подключаемое приложение не использует deflate. В ТЕСТ-среде аналогичные адреса начинаются с *https://login-test.company.com*.
Какой entityID у поставщика идентификации?	Blitz Identity Provider как поставщик идентификации имеет следующие entityID: – Для ПРОД-среды – *https://login.company.com/blitz/saml* – Для ТЕСТ-среды – *https://login-test.company.com/blitz/saml*

4.2. Готовые библиотеки

Так как самостоятельная разработка программного интерфейса клиента SAML является трудоемкой задачей, а ошибки в реализации чреваты угрозами безопасности, при интеграции приложения по SAML рекомендуется использовать существующие популярные библиотеки SAML-клиентов: OIOSAML[40] (Java, .NET), OpenSAML[41] (Java), Spring Security SAML[42] (Java), SimpleSAMLphp[43] (PHP), ruby-saml[44] (Ruby on Rails). Далее приводятся ключевые сведения, необходимые для понимания процесса аутентификации по протоколу SAML.

4.3. Разработка взаимодействия приложения с Blitz Identity Provider по SAML

4.3.1. Общие сведения

Для подключения к Blitz Identity Provider в целях идентификации и аутентификации пользователей приложение может использовать стандарт SAML версий 1.0, 1.1, 2.0[45]. При

[40] См.: https://digitaliser.dk/group/42063/resources

[41] См.: https://wiki.shibboleth.net/confluence/display/OS30/Home

[42] См.: https://spring.io/projects/spring-security-saml

[43] См.: https://simplesamlphp.org/

[44] См.: https://rubygems.org/gems/ruby-saml/

[45] См.: http://saml.xml.org/saml-specifications

этом процесс взаимодействия приложения и Blitz Identity Provider должен быть построен в соответствии с профилем SAML Web Browser SSO Profile[46].

Стандарт SAML основан на XML и определяет способы обмена информацией об аутентификации пользователей и их идентификационных данных (атрибуты, полномочия).

Для возможности осуществлять взаимодействия поставщик услуг и поставщик идентификации предварительно должны обменяться настройками взаимодействия, описанными в форме XML-документов и называемых метаданными. Поставщик услуг должен получить настройки Blitz Identity Provider, называемые метаданными поставщика идентификации (пп. 2.2.1).

4.3.2. Процесс проведения идентификации и аутентификации

В процессе взаимодействия приложение (поставщик услуг) посылает в Blitz Identity Provider SAML-запрос на идентификацию пользователя (SAML Request). Запрос представляет собой оформленный в соответствии со стандартом SAML XML-документ. В запросе присутствует идентификатор запрашивающего идентификацию приложения, называемый *entityID*, а также дополнительная служебная информация. Сам запрос передается подписанным электронной подписью приложения. В качестве транспортного протокола для передачи сообщения используется протокол HTTPS, вызов поставщика идентификации осуществляется через HTTP Redirect. Это означает, что запрос от приложения к Blitz Identity Provider осуществляется опосредованно, через браузер пользователя, и прямое сетевое взаимодействие между приложением и Blitz Identity Provider при использовании SAML не требуется.

Получив SAML-запрос на идентификацию, Blitz Identity Provider идентифицирует принадлежность запроса определенному приложению, после чего отображает пользователю веб-страницу единого входа для проведения идентификации и аутентификации пользователя. В случае успешной идентификации и аутентификации пользователя Blitz Identity Provider передает приложению (поставщику услуг) SAML-ответ (SAML Response). В зависимости от заданных настроек взаимодействия запрос может быть подписанным и зашифрованным. Для формирования подписи и для шифрования используются стандарты XML Signature и XML Encryption. В качестве транспортного протокола для передачи сообщения с результатами идентификации используется протокол HTTPS, вызов поставщика услуг осуществляется через HTTP POST.

[46] См.: https://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

Получив от Blitz Identity Provider SAML-ответ, приложение проверяет его подпись, выполняет расшифровку, после чего извлекает из SAML-утверждений (SAML Assertions) идентификационные данные пользователя (идентификаторы, атрибуты, полномочия).

Процесс взаимодействия приложения и Blitz Identity Provider с использованием SAML приведен на рисунке 9.

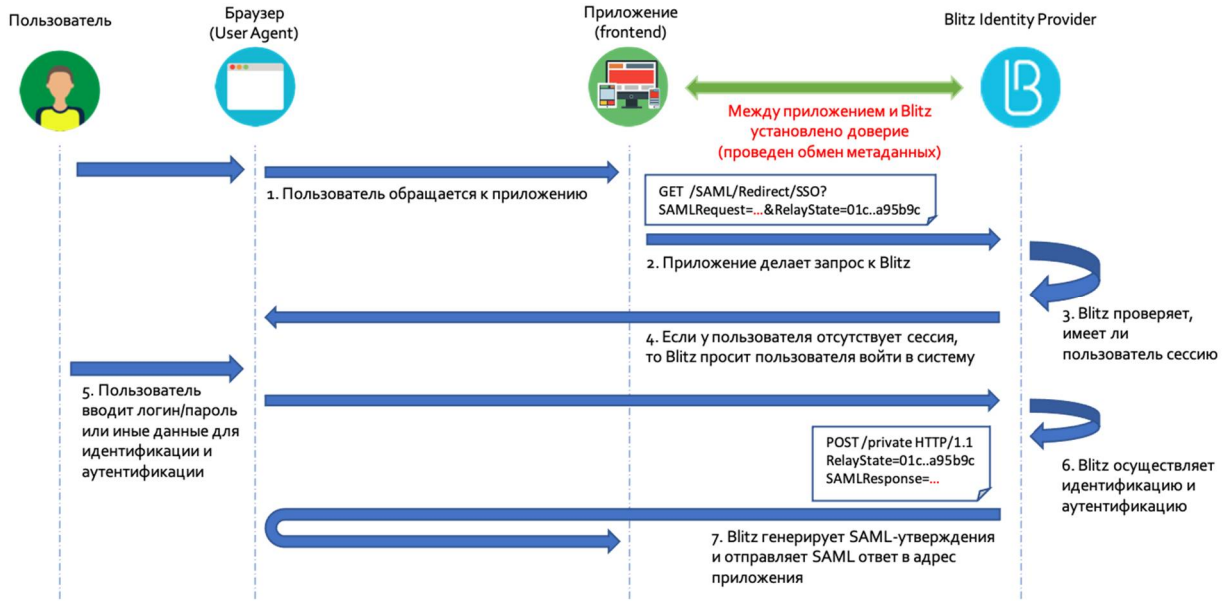

Рисунок 9 — Идентификация пользователя с использованием SAML

4.3.3. Обеспечение логаута при использовании SAML

Подключенное к Blitz Identity Provider по SAML приложение также может предусматривать возможность реализации единого выхода (логаута). Для этих целей Blitz Identity Provider поддерживает SAML Single Logout Profile[47]. Приложение может направить в Blitz Identity Provider SAML-запрос `<LogoutRequest>` и в случае успешного завершения единого логаута получить от Blitz Identity Provider SAML-ответ `<LogoutResponse>`. Если приложение должно быть задействовано в едином логауте, инициированным другим приложением, подключенным к Blitz Identity Provider, то оно также должно предусматривать возможность обработки запросов `<LogoutRequest>`, поступивших к приложению от Blitz Identity Provider. В случае успешного завершения локальной сессии приложение должно уведомлять Blitz Identity Provider путем отправки ему SAML-ответа `<LogoutResponse>`.

[47] См.: https://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

5. Требования по безопасности к приложению

Оператором приложения, подключенного к Blitz Identity Provider, должно обеспечиваться соблюдение следующих требований к безопасности:

1. Должна обеспечиваться конфиденциальность полученного для приложения при регистрации в Blitz Identity Provider значения *client_secret*:
   - Запрещается предавать значение *client_secret* лицам, не связанным с обеспечением эксплуатации приложения.
   - Запрещается использовать *client_secret* в клиентской части ПО (код, выполняемый на стороне браузера, мобильного приложения, десктопного приложения). Применяться *client_secret* должен только в серверных компонентах приложения. Исключение – *client_secret*, полученный мобильным или десктопным приложением с помощью операции динамической регистрации, такой *client_secret* можно хранить и обрабатывать в мобильном или десктопном приложении.
   - В случае если *client_secret* скомпрометирован, то должна быть подана заявка на замену *client_secret* приложения. В Blitz Identity Provider предусмотрена возможность «плавной замены» *client_secret*, а именно, приложению может быть присвоен дополнительный *client_secret* на время, пока будет выполняться перенастройка приложения с прежнего на новое значение *client_secret*.
2. Должна обеспечиваться конфиденциальность полученных приложением от Blitz Identity Provider маркеров доступа (*access_token*) и маркеров обновления (*refresh_token*).
   - Нужно избегать использования маркеров доступа в браузерной части приложения. Если все-таки это необходимо (SPA-приложение), то использующий маркер доступа JS-код должен предусматривать защиту от возможности получения значения маркера доступа из браузерной консоли.
   - Запрещено хранить/обрабатывать маркер обновления на стороне браузерной части приложения – маркер обновления должен использоваться исключительно в серверных компонентах приложения. При хранении маркеров обновления в приложении (в БД, файлах и т.д.) доступ к хранимым маркерам обновления должен быть ограничен.
3. Взаимодействие приложения с Blitz Identity Provider в продуктивном контуре должно осуществляться исключительно с использованием защищенного соединения (HTTPS). Запрещено использовать HTTP в обработчиках приложения (адреса возврата *redirect_uri*, *post_logout_redirect_uri*).

4. Приложению запрещено открывать страницу входа Blitz Identity Provider во фрейме.
5. При подключении мобильных приложений к Blitz Identity Provider:
   - использованием PKCE является обязательным;
   - запрещено использовать Embedded-браузер.

Приложение А. Описание REST API в Blitz Identity Provider

А.1. Версии REST API

В настоящий момент в Blitz Identity Provider доступны следующие версии REST API, различающиеся способом авторизации:

- 1-я – REST-сервисы, доступные по адресу *https://login.company.com/blitz/reg/api/v1/** и *https://login.company.com/blitz/api/v1/**. Для авторизации вызова этих сервисов используется HTTP Basic авторизация. Для приложения, которое будет вызывать REST-сервисы, необходимо в настройках приложения задать пароль на вкладке REST настроек протоколов приложения. Приложению будут доступны все REST-сервисы версии 1. Если какие-то из сервисов использовать не планируется, то рекомендуется запретить их вызов через настройки веб-сервера (nginx). Сервисы этой версий после появления аналогов в более новой 3-й версии будут помечены как устаревшие, и будет рекомендовано перейти с их использования на сервисы 3-й версии.
- 2-я версии – REST-сервисы, доступные по адресу *https://login.company.com/blitz/api/v2/**. Для авторизации вызова большинства этих сервисов используется HTTP Basic авторизация, а для части сервисов – OAuth 2.0. Сервисы этой версий после появления аналогов в более новой 3-й версии будут помечены как устаревшие, и будет рекомендовано перейти с их использования на сервисы 3-й версии.
- 3-я версия – REST-сервисы, доступные по адресу *https://login.company.com/blitz/api/v3/**. Для авторизации вызова этих сервисов используется OAuth 2.0 и полученные от Blitz Identity Provider маркеры безопасности. Доступ приложений к различным REST-сервисам регулируется через разрешения (*scope*).

Предоставляемые Blitz Identity Provider сервисы *https://login.company.com/blitz/api/v3/** можно вызывать в двух режимах:

- Пользовательский режим;
- Системный режим.

В пользовательском режиме сервис вызывается с правами в отношении учетной записи текущего авторизованного пользователя. При вызове сервиса должны передаваться следующие заголовки:

- Authorization: Bearer <*маркер доступа с пользовательскими разрешениями*> – заголовок авторизации, содержащий маркер доступа с разрешениями текущего пользователя. Разрешения описаны в таблице 9.
- X-Forwarded-For*:* <*IP-адрес пользователя*> – заголовок, в котором должно быть передано значение IP-адреса пользователя. Данное значение будет записано в событие безопасности Blitz Identity Provider.
- User-Agent: <*значение User-Agent пользователя*> – заголовок, в котором должно быть передано значение User-Agent устройства пользователя. Данное значение будет записано в событие безопасности Blitz Identity Provider.

A.2. Список разрешений для доступа к REST API

В данном разделе приведен перечень разрешений (*scope* в терминологии OIDC/OAuth 2.0), используемых для доступа к REST API, предоставляемым Blitz Identity Provider. Данные разрешения назначаются отдельным приложениям при необходимости предоставления им доступа к REST API в Blitz Identity Provider, описанным в данном приложении.

Таблица 9 – Разрешения (scope) для REST API в Blitz Identity Provider

№	Разрешение	Название	Описание
Пользовательские разрешения (разрешения, получаемые на пользователя)			
1.	blitz_change_password	Разрешение на изменение пароля	Для использования сервиса POST /blitz/api/v2/users/{subjectId}/password
2.	blitz_user_rights	Разрешение на управление правами учетной записи	Для использования сервиса POST /blitz/api/v2/users/rights/change
3.	blitz_api_user	Получение атрибутов	Для использования сервиса GET /blitz/api/v3/users/{subjectId}
4.	blitz_api_user_chg	Изменение атрибутов	Для использования сервиса POST /blitz/api/v3/users/{instanceId}
5.	blitz_api_usec	Получение настроек двухфакторной аутентификации, разрешений	Для использования сервисов: GET /blitz/api/v3/users/{subjectId}/auth GET /blitz/api/v3/users/{subjectId}/totps GET /blitz/api/v3/users/{subjectId}/acls

6.	blitz_api_usec_chg	Изменение пароля, настроек двухфакторной аутентификации, отзыв разрешений	Для использования сервисов: POST /blitz/api/v3/users/{instanceId}/pswd POST /blitz/api/v3/users/{subjectId}/auth GET /blitz/api/v3/users/{subjectId}/totps /attach/qr POST /blitz/api/v3/users/{subjectId /totps /attach/qr DELETE /blitz/api/v3/users/{subjectId} /totps/{id} DELETE /blitz/api/v3/users/{subjectId} /acls/{id}
7.	blitz_api_uapps	Получение запомненных устройств	Для использования сервиса GET /blitz/api/v3/users/{subjectId}/apps
8.	blitz_api_uapps_chg	Удаление запомненных устройств	Для использования сервиса DELETE /blitz/api/v3/users/{subjectId} /apps/{id}
9.	blitz_api_uaud	Получение событий безопасности	Для использования сервиса GET blitz/api/v3/users/{subjectId}/audit
10.	blitz_qr_auth	Вход с использованием QR-кода	Для использования сервисов: GET /blitz/api/v3/auth/qr/{QR_code} POST /blitz/api/v3/auth/qr/{QR_code}/confirm POST /blitz/api/v3/auth/qr/{QR_code}/refuse
Системные разрешения (разрешения, получаемые на приложение)			
11.	blitz_groups	Разрешение на доступ к сервисам работы с организациями	Для использования сервисов: GET /blitz/api/v2/grps/{id} POST /blitz/api/v2/grps POST /blitz/api/v2/grps/{id}?profile={profile} DELETE /blitz/api/v2/grps/{id}?profile={profile} GET /blitz/api/v2/grps/{id}/members POST /blitz/api/v2/grps/{id}/members/add?profile={ profile}

			POST /blitz/api/v2/grps/{id}/members/rm?profile={profile}
12.	blitz_rights_full_access	Разрешение на назначение и отзыв прав доступа	Для использования сервисов: PUT /blitz/api/v3/rights DELETE /blitz/api/v3/rights GET /blitz/api/v3/rights/on GET /blitz/api/v3/rights/of
13.	blitz_rm_rights	Разрешение на отзыв прав доступа ведомых учетных записей	Для использования сервиса POST /blitz/api/v2/users/rights/change
14.	blitz_api_sys_users	Получение атрибутов любого пользователя	Для использования сервиса GET /blitz/api/v3/users/{subjectId}
15.	blitz_api_sys_users_chg	Изменение атрибутов любого пользователя	Для использования сервиса POST /blitz/api/v3/users/{instanceId}
16.	blitz_api_sys_usec	Получение настроек двухфакторной аутентификации, разрешений любого пользователя	Для использования сервисов: GET /blitz/api/v3/users/{subjectId}/auth GET /blitz/api/v3/users/{subjectId}/totps GET /blitz/api/v3/users/{subjectId}/acls
17.	blitz_api_sys_usec_chg	Изменение пароля, настроек двухфакторной аутентификации, отзыв разрешений любого пользователя	Для использования сервисов: POST /blitz/api/v3/users/{instanceId}/pswd POST /blitz/api/v3/users/{subjectId}/auth GET /blitz/api/v3/users/{subjectId}/totps /attach/qr POST /blitz/api/v3/users/{subjectId}/totps /attach/qr DELETE /blitz/api/v3/users/{subjectId} /totps/{id} DELETE /blitz/api/v3/users/{subjectId} /acls/{id}
18.	blitz_api_sys_uapps	Получение запомненных	Для использования сервиса GET /blitz/api/v3/users/{subjectId}/apps

		устройств любого пользователя	
19.	blitz_api_sys_uapps_chg	Удаление запомненных устройств любого пользователя	Для использования сервиса DELETE /blitz/api/v3/users/{subjectId} /apps/{id}
20.	blitz_api_sys_uaud	Получение событий безопасности любого пользователя	Для использования сервиса GET blitz/api/v3/users/{subjectId}/audit

Маркер доступа на пользовательские разрешения приложение получает в момент идентификации и аутентификации пользователя, осуществляя взаимодействие в соответствии с описанием в пп. 3.3.2 и пп. 3.3.3.

Чтобы получить маркер доступа на системное разрешение, приложение должно выполнить запрос методом POST на URL для получения маркера (*https://login.company.com/blitz/oauth/te*). Запрос должен содержать заголовок *Authorization* со значением *Basic {secret}*, где *secret* – это *client_id:client_secret* (например, *app:topsecret*) в формате Base64.

Пример заголовка:

```
Authorization: Basic YWlzOm…XQ=
```

Тело запроса должно содержать следующие параметры:

- *grant_type* – принимает значение *client_credentials*;
- *scope* – запрашиваемое системное разрешение.

Пример запроса:

```
POST blitz/oauth/te HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Authorization: Basic ZG5ld…lg
Cache-Control: no-cache

grant_type=client_credentials&scope=blitz_groups
```

В ответ приложение получит маркер доступа (*access_token*), время его жизни (*expires_in*) и тип маркера (*token_type*). Возможные ошибки при вызове */oauth/te* соответствуют RFC 6749 и описаны по ссылке[48].

Пример ответа с успешным выполнением запроса:

```
{
    "access_token": "QFiJ9mPgERPuusd36mQvD4mfzYolH_CmuddAJ3YKTOI",
    "expires_in": 3600,
    "scope": "blitz_groups",
    "token_type": "Bearer"
}
```

[48] См.: https://tools.ietf.org/html/rfc6749#section-5.2

Рекомендуется, чтобы приложение кэшировало полученный маркер доступа для многократного использования на время, немного меньшее, чем параметр *expires_in*, после чего осуществляло получение нового маркера доступа для обновления в кэше.

Если приложение попытается вызвать с просроченным маркером доступа соответствующий ему REST-сервис, то получит ошибку *HTTP 401 Unauthorized*.

А.3. Сервисы для управления учетной записью пользователя

Blitz Identity Provider предоставляет сервисы для управления учетной записью пользователя, позволяющие осуществлять следующие операции для существующих функций Blitz Identity Provider:

- регистрация учетной записи пользователя;
- поиск учетной записи пользователя;
- получение атрибутов пользователя;
- изменение значения атрибута учетной записи пользователя;
- изменение номера мобильного телефона;
- изменение адреса электронной почты;
- изменение пароля пользователем;
- проверка состояния режима двухфакторной аутентификации;
- включение/отключение режима двухфакторной аутентификации;
- проверка наличия у пользователя привязки TOTP-генератора;
- привязка TOTP-генератора в качестве средства двухфакторной аутентификации;
- отвязка TOTP-генератора от учетной записи пользователя;
- получение списка привязанных к учетной записи пользователя социальных сетей;
- создание новой привязки социальной сети к пользователю;
- удаление привязки учетной записи социальной сети;
- получение списка событий аудита пользователя;
- получение списка известных устройств пользователя;
- удаления устройства пользователя из списка известных устройств;
- получение списка разрешений, выданных пользователем;
- отзыв выданного пользователем разрешения;
- получение списка привязанных к учетной записи пользователя мобильных приложений;
- отвязка от учетной записи пользователя привязанного мобильного приложения;
- удаление учетной записи пользователя.

A.3.1. Регистрация учетной записи пользователя

Для регистрации учетной записи пользователя приложение должно выполнить запрос методом PUT по адресу *https://login.company.com/blitz/reg/api/v1/users*.

В запрос должен быть добавлен следующий заголовок, где *secret* – это присвоенные приложению при регистрации в Blitz Identity Provider *client_id:rest_secret* в формате Base64:

```
Authorization: Basic <secret>
```

Список атрибутов приведен в качестве образца. Содержание списка необходимо скорректировать в зависимости от конкретных настроек, сделанных при внедрении Blitz Identity Provider. См. «Руководство администратора».

Тело запроса должно содержать атрибуты регистрируемой учетной записи:

- *first_name* – фамилия;
- *name* – имя;
- *middle_name* – отчество;
- *phone_number* – номер мобильного телефона в виде составного объекта с атрибутами:
  - *value* – номер телефона в формате 7XXXXXXXXXX;
  - *verified* – признак, что телефон подтвержден – *true* или *false*;
- *email* – адрес электронной почты в виде составного объекта с атрибутами:
  - *value* – адрес электронной почты;
  - *verified* – признак, что адрес подтвержден – *true* или *false*;
- *password* – пароль для создаваемой учетной записи пользователя (должен соответствовать настроенной парольной политике).

Пример запроса (регистрация с подтвержденными email и телефоном):

```
PUT /blitz/reg/api/v1/users HTTP/1.1
Authorization: Basic YXBwX2lkOmFwcF9zZWNyZXQ=
Content-Type: application/json
Cache-Control: no-cache

{
    "first_name":"Иванов",
    "name":"Иван",
    "middle_name":"Иванович",
    "phone_number": {
        "value": "79991234567",
        "verified": true
    },
    "email": {
        "value":"mail@example.com",
        "verified": true
    },
    "password":"QWErty$123"
}
```

Пример запроса (регистрация с неподтвержденными email и телефоном):

```
PUT /blitz/reg/api/v1/users HTTP/1.1
Authorization: Basic YXBwX2lkOmFwcF9zZWNyZXQ=
Content-Type: application/json
Cache-Control: no-cache

{
```

```
    "first_name":"Иванов",
    "name":"Иван",
    "middle_name":"Иванович",
    "phone_number": {
        "value": "79991234567",
        "verified": false
    },
    "email": {
        "value":"mail@example.com",
        "verified": false
    },
    "password":"QWErty$123"
}
```

Если регистрация вызвана с передачей неподтвержденных телефона и/или email, то сервис отправит пользователю проверочный SMS с кодом подтверждения и/или email с кодом подтверждения, и вернет сервисные атрибуты *instructions* и *context*.

Пример ответа сервиса (требуется ввод пользователем проверочных кодов):

```
{
    "instructions": [
        {
            "email":"mail@example.com",
            "exp":1654852044,
            "attemts":3,
            "name":"eml-enter-code"
        },
        {
            "mobile":"79991234567",
            "exp":1654849105,
            "attemts":3,
            "name":"mbl-enter-code"
        }
    ],
    "context":"YNx9…Dw"
}
```

Нужно запросить у пользователя ввод кода подтверждения, отправленного на email и на мобильный телефон. После ввода каждого кода вызвать сервис для подтверждения контакта, указанного при регистрации, передав в URL запроса значение из параметра *context*, а в теле запроса – введенный пользователем код подтверждения:

Пример запроса для подтверждения email:

```
POST /blitz/reg/api/v1/users/YNx9…Dw HTTP/1.1
Authorization: Basic YXBwX2lkOmFwcF9zZWNyZXQ=
Content-Type: application/json
Cache-Control: no-cache

{
    "email_code":"269302"
}
```

Пример ошибки, если введен неправильный код подтверждения email:

```
{
    "instructions": [
        {
            "email":"mail123@example.com",
            "exp":1655283696,
            "attemts":2,
            "name":"eml-try-again"},
        {
            "mobile":"79988984169",
            "exp":1655280756,
            "attemts":3,
            "name":"mbl-try-again"
        }
    ],
    "context":"kE6r…7g"
}
```

Пример ошибки, если истек срок действия или превышено число попыток (будет общая ошибка eml-expired):

```
{
    "instructions": [
        {
            "email":"mail123@example.com",
            "name":"eml-expired"
        },
        {
            "mobile":"79988984169",
            "exp":1655280756,
            "attemts":3,"name":"mbl-try-again"
        }
    ],
    "context":"kE6r…7g"
}
```

Пример запроса для инициирования повторной отправки кода по email (в качестве значения параметра указать любой код):

```
POST /blitz/reg/api/v1/users/YNx9…Dw HTTP/1.1
Authorization: Basic YXBwX2lkOmFwcF9zZWNyZXQ=
Content-Type: application/json
Cache-Control: no-cache

{
    "email_code_resend":"123456"
}
```

В случае если email успешно подтвержден, и осталось подтвердить телефон, то в ответе сервиса исчезнем инструкция про подтверждение email, и останется только инструкция про телефон:

```
{
    "instructions": [
        {
            "mobile":"79988984169",
            "exp":1655280756,
            "attemts":3,
            "name":"mbl-try-again"
        }
    ],
    "context":"kE6r…7g"
}
```

Пример запроса для подтверждения телефона:

```
POST /blitz/reg/api/v1/users/YNx9…Dw HTTP/1.1
Authorization: Basic YXBwX2lkOmFwcF9zZWNyZXQ=
Content-Type: application/json
Cache-Control: no-cache

{
    "sms_code":"953568"
}
```

Пример ошибки, если введен неправильный код подтверждения телефона:

```
{
    "instructions": [
        {
            "email":"mail123@example.com",
            "exp":1655283696,
            "attemts":2,
            "name":"eml-try-again"},
        {
            "mobile":"79988984169",
            "exp":1655280756,
            "attemts":3,
            "name":"mbl-try-again"
        }
    ],
    "context":"kE6r…7g"
}
```

Пример ошибки, если истек срок действия:

```
{
    "instructions": [
        {
            "mobile":"79988984169",
            "name":"mbl-expired"
        }
    ],
    "context":"kE6r…7g"
}
```

Пример ошибки, если превышено число попыток:

```
{
    "instructions": [
        {
            "mobile":"79988984169",
            "name":"mbl-no-attempts"
        }
    ],
    "context":"kE6r…7g"
}
```

Пример запроса для инициирования повторной отправки кода по SMS (в качестве значения параметра указать любой код):

```
POST /blitz/reg/api/v1/users/YNx9…Dw HTTP/1.1
Authorization: Basic YXBwX2lkOmFwcF9zZWNyZXQ=
Content-Type: application/json
Cache-Control: no-cache

{
    "sms_code_resend":"123456"
}
```

Если регистрация вызывалась с подтвержденными контактами или все контакты были подтверждены в процессе регистрации, то в результате вызова сервиса в Blitz Identity Provider будет зарегистрирована учетная запись пользователя с предоставленными атрибутами и паролем. Сервис вернет присвоенный учетной записи идентификатор пользователя (*subject*). Кроме того, вернется ряд сервисных атрибутов (*instructions* и *context*).

Пример ответа:

```
{
    "subject": "c629340d-4aa9-48b1-b890-bc030020b4eb",
    "instanceId": "dGFtOk…3M",
    "instructions": [],
    "context": "HDTR…UA"
}
```

Регистрация может завершиться ошибкой. Тогда в теле ответа будет пояснение проблемы. В частности, если в Blitz Identity Provider нарушена уникальность атрибута, то сообщение будет содержать перечень полей, по которым нарушена уникальность.

Пример ответа с сообщением об ошибке:

```
{
    "errors": [
        {
            "errMsg": "Такой пользователь уже зарегистрирован…",
            "field": "email"
        },
        {
            "errMsg": "Такой пользователь уже зарегистрирован…",
            "field": "phone_number"
        }
    ],
```

```
    "context": ""
}
```

A.3.2. Поиск учетной записи пользователя

Для поиска учетной записи необходимо методом GET вызвать сервис по адресу *https://login.company.com/blitz/api/v1/users*. В качестве URL параметра query передать поисковый запрос в форме Resource Query Language (RQL)[49].Поддерживаются следующие операции:

- *and* – одновременное выполнение поисковых условий;
- *or* – альтернативное выполнение поисковых условий (например, поиск по разным атрибутам);
- *eq* – проверка условия равенства.

При выполнении поиска по атрибуту, имеющему строковое значение, рекомендуется явно специфицировать тип значения. Например, string:02142527602.

Пример простого поискового запроса:

```
GET /blitz/api/v1/users?query=phone_number=string:79991234567 HTTP/1.1
Authorization: Basic YXBwX2lkOmFwcF9zZWNyZXQ=
Content-Type: application/json
```

Пример ответа:

```
[
   {
      "instanceId":"Mzg5…nU",
      "attrs":{
         "sub":"854436f6-af58-4a3f-8cb7-c2c441eb4a76",
         "family_name":"Иванов",
         "given_name":"Иван",
         "middle_name":"Иванович",
         "phone_number":"79991234567",
      }
   },
   …
]
```

Пример сложного поискового запроса:

```
GET /blitz/api/v1/users?
query=or(eq(phone_number,string:79991234567),eq(phone_number,string:79991112233)) HTTP/1.1
Authorization: Basic YXBwX2lkOmFwcF9zZWNyZXQ=
Content-Type: application/json
```

A.3.3. Получение атрибутов пользователя

Для получения атрибутов любого пользователя по его идентификатору необходимо методом GET вызвать сервис по адресу *https://login.company.com/blitz/api/v3/users/{subjectId}*.

Необходимые разрешения: *blitz_api_user* или *blitz_api_sys_users*.

В результате выполнения запроса Blitz Identity Provider вернет JSON, содержащий атрибуты пользователя.

Пример запроса:

```
GET /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a HTTP/1.1
Authorization: Bearer cNw…Nz
```

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

```
Content-Type: application/json
```

Пример ответа:

```
{
    "family_name": "Иванов",
    "sub": "d2580c98-e584-4aad-a591-97a8cf45cd2a",
    "given_name": "Иван",
    "locked": false,
    "meta": {
        "instanceId": "Mzg…J1",
        "unmodifiable": [
            "sub"
        ]
    }
}
```

В блоке meta передаются метаданные учетной записи. Атрибут *instanceId* нужен для возможности вызова в последующем сервисов изменения атрибутов учетной записи и изменения пароля.

A.3.4. Изменение атрибута учетной записи

Для изменения секретного вопроса и ответа на секретный вопрос необходимо методом POST вызвать сервис по адресу *https://login.company.com/blitz/api/v3/users/{instanceId}*.

Необходимые разрешения: *blitz_api_user_chg* или *blitz_api_sys_users_chg*.

Тело запроса должно содержать значения изменяемых атрибутов пользователя.

Пример запроса:

```
POST /blitz/api/v3/users/Mzg…J1 HTTP/1.1
Authorization: Bearer cNw…Nz
Content-Type: application/json
Cache-Control: no-cache

{
   "family_name":"Петров"
}
```

Пример ответа:

```
{
    "family_name": "Петров",
    "given_name": "Иван",
    "locked": false,
    "sub": "5cffd68f-2cb8-4f7a-b0f3-9fa69a1fbbcd",
    "meta": {
        "instanceId": "Mzg…J1",
        "unmodifiable": [
            "sub"
        ]
    }
}
```

При ошибке выполнения запроса, что переданные значения атрибутов не прошли проверку, вернется ошибка HTTP 400 Bad Request. Вложенный JSON в теле ответа будет включать:

- тип ошибки (type) – имеет значение “input_error” для случаев, когда запрос содержит некорректное или недопустимое значение;
- код ошибки (error);
- текстовое описание ошибки.

Коды ошибок и тексты ошибок могут быть определены специфично для различных атрибутов свои и определяться реализованной для атрибутов логикой валидаторов.

Пример ошибки:

```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "contact_use_violation",
            "desc": "Validation mobile:79988887812 is failed.",
            "pos": "mobile"
        }
    ]
}
```

A.3.5. Изменение номера мобильного телефона

Для изменения номера мобильного телефона необходимо методом POST вызвать сервис по адресу *https://login.company.com/blitz/api/v3/users/{instanceId}*.

Необходимые разрешения: *blitz_api_user_chg* или *blitz_api_sys_users_chg*.

Предусмотрены следующие режимы изменения:

– Изменение телефона сразу на подтвержденный

– Изменение телефона с прохождением подтверждения

Тело запроса должно содержать следующие параметры:

– *phone_number* – мобильный телефон, передается в виде составного объекта с атрибутами:

- *value* – номер телефона в формате 7XXXXXXXXXX;
- *vrf* – признак, что телефон подтвержден – true.

> **Примечание** — в отличие от сервиса регистрации признак подтвержденности называется vrf, а не verified.

Пример запроса с изменением сразу на подтвержденный:

```
POST /blitz/api/v3/users/Mzg…J1 HTTP/1.1
Authorization: Bearer wzb…Tw
Content-Type: application/json
Cache-Control: no-cache

{
    "phone_number":
        {
            "value":"79991234567",
            "vrf":true
        }
}
```

Пример ответа:

```
{
    "given_name": "Иван",
    "family_name": "Иванов",
    "meta": {
        "instanceId": "Mzg5L…2M",
        "unmodifiable": [
            "uid"
        ]
    },
    "email": {
```

```
        "value": "aivanov+2@gmail.com",
        "vrf": true
    },
    "sub": "BIP-LIR6BO33XBBDHANE6DZPUTYVME",
    "phone_number": {
        "value": "+7(999)1234567",
        "vrf": true
    }
}
```

Пример запроса с изменением атрибута с прохождением подтверждения:

```
POST /blitz/api/v3/users/Mzg…J1 HTTP/1.1
Content-Type: application/json
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/83.0.4103.106 Safari/537.36
Authorization: Bearer wzb…Tw
{
    "phone_number":{"value":"+79999999998","vrf":false}
}
```

Пример ответа:

```
{
    "given_name": "Иван",
    "family_name": "Иванов",
    "meta": {
        "instanceId": "Mzg5L…2M",
        "unmodifiable": [
            "sub"
        ]
    },
    "email": {
        "value": "aivanov+2@gmail.com",
        "vrf": true
    },
    "sub": "BIP-LIR6BO33XBBDHANE6DZPUTYVME",
    "notes": {
        "actions": {
            "state": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
            "exp": 300,
            "status": "code_waiting",
            "from": "+7(964)1234567",
            "attr": "mobile",
            "attempts_left": 3,
            "value": "+7(999)9999998",
            "action": "validate_mobile",
            "created": 1598446512
        }
    },
    "phone_number": {
        "value": "+7(964)1234567",
        "vrf": true
    }
}
```

Пример запроса на подтверждение смены (в URL используется *action* и *state* из ответа сервиса):

```
POST /blitz/api/v3/users/notes/validate_mobile/ch_El…yQ HTTP/1.1
Content-Type: application/json
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)…
Authorization: Bearer wzb…Tw
{
    "cmd": "code",
    "value": "123456"
}
```

Пример ответа (успешное изменение):

```
{
    "given_name": "Иван",
    "family_name": "Иванов",
    "meta": {
        "instanceId": "Mzg5L…2M",
        "unmodifiable": [
            "sub"
        ]
    },
```

```
    "email": {
        "value": "aivanov+2@gmail.com",
        "vrf": true
    },
    "sub": "BIP-LIR6BO33XBBDHANE6DZPUTYVME",
    "phone_number": {
        "value": "+7(999)9999998",
        "vrf": true
    }
}
```


Пример ответа (неправильный код):


```
{
    "state": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
    "exp": 2592000,
    "from": "+7(964)1234567",
    "attr": "phone_number",
    "msg": "wrong_code",
    "attempts_left": 2,
    "created": 1649695409,
    "value": "+7(999)9999998",
    "action": "validate_mobile"
}
```


Пример ответа (неправильный код после последней попытки):


```
{
    "state": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
    "id": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
    "attr": "phone_number",
    "cause": "no_attempts_left",
    "from": "+7(964)1234567",
    "value": "+7(999)9999998",
    "action": "validate_mobile"
}
```


Пример ответа (код просрочен):


```
{
    "state": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
    "id": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
    "attr": "phone_number",
    "cause": "code_expired",
    "from": "+7(964)1234567",
    "value": "+7(999)9999998",
    "action": "validate_mobile"
}
```


A.3.6. Изменение адреса электронной почты

Для изменения адреса электронной почты необходимо методом POST вызвать сервис по адресу *https://login.company.com/blitz/api/v3/users/{instanceId}*.

Необходимые разрешения: *blitz_api_user_chg* или *blitz_api_sys_users_chg*.

Предусмотрены следующие режимы изменения:

– Изменение email сразу на подтвержденный

– Изменение email с прохождением подтверждения

Тело запроса должно содержать следующие параметры:

– *email* – адрес электронной почты:

- *value* – адрес электронной почты;
- *vrf* – признак, что адрес подтвержден – true;

> **Примечание** — в отличие от сервиса регистрации признак подтвержденности называется vrf, а не verified.

Пример запроса с изменением сразу на подтвержденный:

```
POST /blitz/api/v3/users/Mzg…J1 HTTP/1.1
Authorization: Bearer wzb…Tw
Content-Type: application/json
Cache-Control: no-cache

{
    "email":
        {
            "value":"mail@example.com",
            "vrf":true
        }
}
```

Пример ответа:

```
{
    "given_name": "Иван",
    "family_name": "Иванов",
    "meta": {
        "instanceId": "Mzg5LW…2M",
        "unmodifiable": [
            "sub"
        ]
    },
    "mail": {
        "value": "mail@example.com",
        "vrf": true
    },
    "sub": "BIP-LIR6BO33XBBDHANE6DZPUTYVME",
    "phone_number": {
        "value": "+7(999)1234567",
        "vrf": true
    }
}
```

Пример запроса с изменением атрибута с прохождением подтверждения:

```
POST /blitz/api/v3/users/Mzg…J1 HTTP/1.1
Content-Type: application/json
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/83.0.4103.106 Safari/537.36
Authorization: Bearer wzb…Tw
{
    "email":{"value":"mail@example.com","vrf":false}
}
```

Пример ответа:

```
{
    "given_name": "Иван",
    "family_name": "Иванов",
    "meta": {
        "instanceId": "Mzg5L…2M",
        "unmodifiable": [
            "sub"
        ]
    },
    "email": {
        "value": "aivanov+2@gmail.com",
        "vrf": true
    },
    "sub": "BIP-LIR6BO33XBBDHANE6DZPUTYVME",
    "notes": {
        "actions": {
            "state": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
            "exp": 86400,
            "status": "code_waiting",
            "from": "aivanov+2@gmail.com",
            "attr": "mail",
            "attempts_left": 3,
            "value": "mail@example.com",
            "action": "validate_mail",
            "created": 1598446512
        }
    },
    "phone_number": {
        "value": "+7(964)1234567",
        "vrf": true
    }
```

```
}
```

Пример запроса на подтверждение смены (в URL используется *action* и *state* из ответа сервиса):

```
POST /blitz/api/v3/users/notes/validate_email/ch_El…yQ HTTP/1.1
Content-Type: application/json
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)…
Authorization: Bearer wzb…Tw
{
    "cmd": "code",
    "value": "123456"
}
```

Пример ответа (успешное изменение):

```
{
    "given_name": "Иван",
    "family_name": "Иванов",
    "meta": {
        "instanceId": "Mzg5L…2M",
        "unmodifiable": [
            "sub"
        ]
    },
    "email": {
        "value": "mail@example.com",
        "vrf": true
    },
    "sub": "BIP-LIR6BO33XBBDHANE6DZPUTYVME",
    "phone_number": {
        "value": "+7(999)9999998",
        "vrf": true
    }
}
```

Пример ответа (неправильный код):

```
{
    "state": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
    "exp": 2592000,
    "from": "aivanov+2@gmail.com",
    "attr": "email",
    "msg": "wrong_code",
    "attempts_left": 2,
    "created": 1649695409,
    "value": "mail@example.com",
    "action": "validate_email"
}
```

Пример ответа (неправильный код после последней попытки):

```
{
    "state": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
    "id": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
    "attr": "email",
    "cause": "no_attempts_left",
    "from": "aivanov+2@gmail.com",
    "value": "mail@example.com",
    "action": "validate_email"
}
```

Пример ответа (код просрочен):

```
{
    "state": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
    "id": "ch_EludIw5fEDouy8wpT_GVOJ7rLxKfZUi-G3blijf34yQ",
    "attr": "email",
    "cause": "code_expired",
    "from": "aivanov+2@gmail.com",
    "value": "mail@example.com",
    "action": "validate_email"
}
```

A.3.7. Изменение пароля пользователем

Для изменения пароля необходимо вызвать методом POST сервис по адресу *https://login.company.com/blitz/api/v3/users/{instanceId}/pswd*.

Чтобы узнать значение *instanceId* для пользователя, необходимо предварительно вызвать методом GET сервис получения атрибутов пользователя (А.3.3).

Необходимые разрешения: *blitz_api_usec_chg* или *blitz_api_sys_usec_chg*.

При смене пароля в пользовательском режиме необходимо передать заголовки с IP-адресом пользователя и User-Agent.

В Blitz Identity Provider есть функция безопасности, что при смене пользователем пароля происходит аннулирование всех сессий пользователя и удаление всех запомненных устройств, с которых пользователь ранее осуществлял вход, и на которых был возможен повторный вход без ввода пароля. В сценарии самостоятельной смены пользователем пароля в Личном кабинете может быть нежелательно, чтобы произошел выход пользователя в том числе с текущего устройства/браузера. Для того, чтобы указать Blitz Identity Provider, что определенное устройство необходимо сохранить по результатам успешной смены пароля (не делать с него логаут), необходимо в вызов сервиса смены пароля передать от приложения заголовок *IB-CI-UA-ID* с идентификатором текущего устройства пользователя. Идентификатор текущего устройства пользователя можно получить из маркера идентификации (см. п. 3.3.4).

Тело запроса должно содержать следующие параметры:

- *current* – текущий пароль пользователя (только при смене пароля в пользовательском режиме – в этом случае параметр должен быть обязательно передан);
- *password* – новый пароль пользователя (необязательный параметр). Если параметр не задан, то Blitz Identity Provider самостоятельно сгенерирует новый пароль;
- *sendPswdToAttr* – имя атрибута с телефонным номером для отправки пользователю пароля (необязательный параметр). Если параметр задан, то пользователю на телефон из указанного атрибута будет отправлена SMS с паролем.

Пример запроса (в пользовательском режиме смены пароля):

```
POST /blitz/api/v3/users/Mzg…J1/pswd HTTP/1.1
Content-Type: application/json
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)…
Authorization: Bearer wzb…Tw
IB-CI-UA-ID: {SHA256}rVWFmwgRKWeW_flH4CA4yuW7OhKZ32Da94m0kzwWsVs
{
    "current": "QWErty123",
    "password": "P@$$w0rd"
}
```

Пример запроса (в режиме смены пароля системой):

```
POST /blitz/api/v3/users/Mzg…J1/pswd HTTP/1.1
Content-Type: application/json
Authorization: Bearer qwa…Ez
{
    "password": "P@$$w0rd"
}
```

Пример запроса (отправка нового пароля по SMS с автоматической генерацией пароля):

```
POST /blitz/api/v3/users/Mzg…J1/pswd HTTP/1.1
Content-Type: application/json
Authorization: Bearer qwa…Ez
{
    "sendPswdToAttr": "phone_number"
}
```

В случае успешного вызова Blitz Identity Provider вернет код *HTTP 204 No Content*. Смена пароля также приведет к аннулированию ранее полученных маркеров доступа и маркеров обновления текущего пользователя.

Если смена пароля завершилась ошибкой, то Blitz Identity Provider вернет сообщение об ошибке. Возможны коды ошибок *HTTP 401 Unauthorized* в случае ошибки контроля доступа (неправильный маркер доступа или неправильный текущий пароль пользователя) или *HTTP 400 Bad Request* (новый пароль не удовлетворяет требованиям парольной политики).

Пример ошибки с неправильным текущим паролем:

```
{
    "type": "security_error",
    "error": "invalid_credential",
    "desc": "Wrong subject identifier or current password"
}
```

Пример ошибки с неправильным маркером доступа:

```
{
    "type": "security_error",
    "error": "bad_access_token",
    "desc": "BEARER_AUTH: CRID does not match"
}
```

Пример ошибки, что новый пароль не соответствует парольной политике (слишком короткий):

```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "password_policy_violated",
            "desc": "Password's length must be greater than 6",
            "pos": "password",
            "params": {
                "rule": "to_short",
                "low": 6
            }
        }
    ]
}
```

Пример ошибки, что новый пароль не соответствует парольной политике, установленной в LDAP-каталоге:

```
{
    "type": "input_error",
    "error": "password_policy_violated",
    "desc": "Failed to update password\n",
    "pos": "password",
    "params": {
        "rule": "id_store"
    }
}
```

Пример ошибки, что новый пароль не соответствует парольной политике (не содержит требуемых групп символов):


```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "password_policy_violated",
            "desc": "Password doesn't match enough symbols groups",
            "pos": "password",
            "params": {
                "rule": "not_enough_groups",
                "no_matched_groups": [
                    {
                        "desc": "password.policy.desc.digits",
                        "min_number_symbols": 1
                    },
                    {
                        "desc": "password.policy.desc.capital",
                        "min_number_symbols": 1
                    },
                    {
                        "desc": "password.policy.desc.special",
                        "min_number_symbols": 1
                    }
                ]
            }
        }
    ]
}
```


Пример ошибки, что новый пароль не соответствует парольной политике (пароль ранее использовался):


```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "password_policy_violated",
            "desc": "Password found in previous used ones",
            "pos": "password",
            "params": {
                "rule": "in_password_history"
            }
        }
    ]
}
```


Пример ошибки, что новый пароль не соответствует парольной политике (новый пароль совпадает с текущим):


```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "password_policy_violated",
            "desc": "A new password can't be the same as the current",
            "pos": "password",
            "params": {
                "rule": "eq_current"
            }
        }
    ]
}
```


Пример ошибки, что новый пароль не соответствует парольной политике (в новом пароле недостаточное число символов отличается от прежнего):


```
{
    "type": "input_error",
    "error": "wrong_values",
```

```
    "errors": [
        {
            "type": "input_error",
            "error": "password_policy_violated",
            "desc": "There are not enough new characters in a new password",
            "pos": "password",
            "params": {
                "rule": "not_enough_new_chars",
                "minNew": 5
            }
        }
    ]
}
```

Пример ошибки, что новый пароль не соответствует парольной политике (пароль включает вхождение из словаря запрещенных паролей):

```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "password_policy_violated",
            "desc": "Password contains a word from the stop dictionary",
            "pos": "password",
            "params": {
                "rule": "in_stop_dic",
                "stop_word": "qwerty"
            }
        }
    ]
}
```

Пример ошибки, что новый пароль не соответствует парольной политике (пароль совпадает со словарным):

```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "password_policy_violated",
            "desc": "Password found in a password dictionary",
            "pos": "password",
            "params": {
                "rule": "in_password_dic"
            }
        }
    ]
}
```

Пример ошибки, что новый пароль не соответствует парольной политике (пароль изменен ранее разрешенного срока):

```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "password_policy_violated",
            "desc": "Password is too young",
            "pos": "password",
            "params": {
                "rule": "too_young",
                "minAgeInSec": 86400
            }
        }
    ]
}
```

Пример ошибки, что переданный атрибут для отправки пароля не существует:

```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
```

```
        {
            "type": "input_error",
            "error": "wrong_value",
            "desc": "Wrong mobile attribute 'phone_number_wrong'",
            "pos": "sendPswdToAttr"
        }
    ]
}
```

Пример ошибки, что у пользователя не задан атрибут с телефоном для отправки пароля на телефон:

```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "wrong_value",
            "desc": "User not contains mobile attribute 'phone_number'",
            "pos": "sendPswdToAttr"
        }
    ]
}
```

A.3.8. Изменение пароля ведомой учетной записи пользователя

Изменение пароля ведомой учетной записи пользователя с помощью ведущей учетной записи пользователя осуществляется методом POST по адресу *https://login.company.com/blitz/api/v2/users/{subject}/password*, где subject – это идентификатор (*sub*) ведомой учетной записи. В запрос должен быть добавлен заголовок с маркером доступа на разрешение с именем *blitz_change_password*, полученным ведущей учетной записью. Ведущий пользователь может вызвать смену пароля ведомого только если ранее ведущему пользователю было дано право на изменение пароля change_password (см. А.5.4).

Тело запроса должно содержать атрибут *value* со значением нового пароля. Переданный пароль должен соответствовать требованиям настроенной парольной политики.

Пример запроса:

```
POST /blitz/api/v2/users/c574a512-3704-4576-bc3a-3fe28b636e85/password HTTP/1.1
Authorization: Bearer cNwIX…Tg
Content-Type: application/json
Cache-Control: no-cache
{"value":"QWErty1234"}
```

При успешной смене пароля сервис возвращает статус *HTTP 200 (OK)*.

При наличии ошибки сервис возвращает описание полученной ошибки.

Пример ответа с обнаруженными ошибками:

```
{
    "errors": [
        {
            "code": "access_denied",
            "desc": "Not enough rights: change_password",
            "params": {}
        }
    ]
}
```

A.3.9. Проверка состояния режимов аутентификации

Можно проверить состояния следующих режимов аутентификации учетной записи:

- включена ли у пользователя двухфакторная аутентификация;
- установлен ли у пользователя признак необходимости смены пароля;
- установлен ли у пользователя временный запрет по входу с использованием определенного метода входа.

Для проверки состояния режимов аутентификации необходимо вызвать методом GET сервис по адресу *https://login.company.com/blitz/api/v3/users/{subjectId}/auth*.

Необходимые разрешения: *blitz_api_usec* или *blitz_api_sys_usec*.

В ответ будут получены следующие режимы:

- *requiredFactor* – признак включенной двухфакторной аутентификации. Может принимать следующие значения:
  - *отсутствует, 0 или 1* – выключен
  - *2* – включен (требуется 2-й фактор аутентификации)
- *needPasswordChange* – признак необходимости смены пароля при входе;
- *methodsLocked* – список заблокированных методов аутентификации – пользователь не может использовать именно эти методы входа, но может использовать остальные.

Пример запроса:

```
GET /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/auth HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
Cache-Control: no-cache
```

Пример ответа:

```
{
    "requiredFactor": 2,
    "needPasswordChange": true,
    "methodsLocked": ["password"]
}
```

A.3.10. Изменение режимов аутентификации

Для изменения режимов аутентификации пользователя необходимо вызвать методом POST сервис по адресу *https://login.company.com/blitz/api/v3/users/{subjectId}/auth*.

Необходимые разрешения: *blitz_api_usec_chg* или *blitz_api_sys_usec_chg*.

При изменении настройки в пользовательском режиме необходимо передать заголовки с IP-адресом пользователя и User-Agent.

Тело запроса может содержать следующие параметры:

- *requiredFactor* – признак включенной двухфакторной аутентификации. Может принимать следующие значения:
  - *null* – выключен
  - *2* – включен (требуется 2-й фактор аутентификации)

- *needPasswordChange* – признак необходимости смены пароля при входе – допустима только передача значения *true*;
- *methodsLocked* – список заблокированных методов аутентификации – пользователь не может использовать именно эти методы входа, но может использовать остальные. В настоящий момент Blitz Identity Provider поддерживает только блокирование использования парольного входа (password).

Пример запроса:

```
POST /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/auth HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)…
Content-Type: application/json
Cache-Control: no-cache

{
    "requiredFactor": 2,
    "needPasswordChange": true,
    "methodsLocked": ["password"]
}
```

Пример ответа:

```
{
    "requiredFactor": 2,
    "needPasswordChange": true,
    "methodsLocked": ["password"]
}
```

Пример ошибки *HTTP 400 Bad Request* (если у пользователя не настроен ни один метод для второго фактора аутентификации):

```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "has_not_sf_methods",
            "desc": "User 'd2580c98-e584-4aad-a591-97a8cf45cd2a' has not any second factor
method",
            "pos": "requiredFactor"
        }
    ]
}
```

A.3.11. Проверка наличия у пользователя привязки TOTP-генератора

Для проверки того, настроен ли у пользователя TOTP[50]-генератор кодов подтверждения, необходимо вызвать методом GET сервис по адресу *https://login.company.com/blitz/api/v3/users/{subjectId}/totps*. Если метод настроен, то в ответ будут получены его настройки.

Необходимые разрешения: *blitz_api_usec* или *blitz_api_sys_usec*.

Пример запроса:

```
GET /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/totps HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
Cache-Control: no-cache
```

[50] RFC 6238 «TOTP: Time-Based One-Time Password Algorithm», см.: https://tools.ietf.org/html/rfc6238

Пример ответа:

```
[
    {
        "id": "SW_TOTP_1_d2580c98-e584-4aad-a591-97a8cf45cd2a",
        "len": 6,
        "name": "Google Authenticator"
    }
]
```

A.3.12. Привязка TOTP-генератора

Привязка к учетной записи пользователя TOTP-генератора осуществляется в два этапа:

- Запрос в Blitz Identity Provider QR-кода и строки привязки
- Подтверждение регистрации привязки

Необходимые разрешения: *blitz_api_usec_chg* или *blitz_api_sys_usec_chg*.

При изменении настройки в пользовательском режиме необходимо передать заголовки с IP-адресом пользователя и User-Agent.

На первом этапе приложение должно вызвать методом GET сервис по адресу *https://login.company.com/blitz/api/v3/users/{subjectId}/totps/attach/qr*. В ответ от Blitz Identity Provider будут получены атрибуты:

- *base64QRCode* – QR-код привязки генератора, который нужно отобразить пользователю;
- *base32Secret* – секретная строка привязки генератора, которую нужно отобразить пользователю, если ему неудобно будет фотографировать QR-код, и он предпочтет ввести код привязки в генератор вручную.

Пример запроса:

```
GET /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/totps/attach/qr HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
Cache-Control: no-cache
```

Пример ответа:

```
{
    "base64QRCode": "iVB…g==",
    "base32Secret": "W247OHVTPPTIAOXMGKK6Z7BZ3DEYWO74"
}
```

На втором этапе приложение должно вызвать методом POST сервис по адресу *https://login.company.com/blitz/api/v3/users/{subjectId}/totps/attach/qr*.

Тело запроса должно содержать следующие параметры:

- *base32Secret* – секретная строка инициализации TOTP-генератора;
- *otpCode* – код подтверждения, выработанный генератором по алгоритму TOTP от строки secret и текущего временного слота;
- *name* – отображаемое имя TOTP-генератора (необязательно).

Пример запроса:

```
POST /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/totps/attach/qr HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
```

```
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)…

{
    "base32Secret": "W247OHVTPPTIAOXMGKK6Z7BZ3DEYWO74",
    "name": "Google Authenticator",
    "otpCode": "123456"
}
```

Пример ответа:

```
{
    "base64QRCode": "iVB…g==",
    "base32Secret": "W247OHVTPPTIAOXMGKK6Z7BZ3DEYWO74"
}
```

В случае успешного выполнения сервис вернет *HTTP 204 No Content*.

В случае ошибки сервис вернет *HTTP 400 Bad Request*.

Пример ошибки, если передан неправильный код:

```
{
    "type": "process_error",
    "error": "wrong_otp_code"
}
```

A.3.13. Удаление привязки TOTP-генератора

Чтобы удалить привязку TOTP-генератора к учетной записи пользователя, необходимо вызвать методом DELETE сервис по адресу *https://login.company.com/blitz/api/v3/users/{subjectId}/totps/{id}*, где в качестве *id* указать идентификатор привязки, полученный в пп. A.3.11.

Необходимые разрешения: *blitz_api_usec_chg* или *blitz_api_sys_usec_chg*.

При вызове в пользовательском режиме необходимо передать заголовки с IP-адресом пользователя и User-Agent.

Пример запроса:

```
DELETE /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/totps/SW_TOTP_1_d2580c98-e584-
4aad-a591-97a8cf45cd2a HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)…
```

При успешном выполнении сервис вернет *HTTP 204 No Content*.

A.3.14. Получение списка привязанных социальных сетей

Чтобы получить список привязок социальных сетей к учетной записи пользователя, необходимо вызвать методом GET сервис по адресу *https://login.company.com/blitz/api/v2/users/{subjectId}/fa*. В ответ будут получены тип привязки (*fpKey*) и идентификатор привязки (*sid*).

Пример запроса:

```
GET /blitz/api/v2/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/fa HTTP/1.1
Authorization: Basic ZG5ldm5pay10ZXN0Lm1vcy5ydTphUU56S0JuY2VBQVQwelg
Cache-Control: no-cache
```

Пример ответа:

```
[
    {
        "sid": "1169460959792489",
        "fpKey": "yandex:yandex_1",
        "sbjId": "d2580c98-e584-4aad-a591-97a8cf45cd2a"
```

```
    },
    {
        "sid": "385956543",
        "fpKey": "vk:vk_1",
        "sbjId": "d2580c98-e584-4aad-a591-97a8cf45cd2a"
    }
]
```

A.3.15. Создание новой привязки социальной сети к пользователю

Привязка к учетной записи социальной сети осуществляется в два этапа:

- Запрос инструкции привязки
- Выполнение привязки

На первом этапе приложение должно вызвать методом POST сервис по адресу *https://login.company.com/blitz/api/v2/users/current/fa/bind*.

Тело запроса должно содержать следующие параметры:

- *fp* – идентификатор поставщика, связь с профилем которого должна быть установлена;
- *callback* – адрес, на который должен быть возвращен пользователь после успешной привязки аккаунта соцсети;
- *isPopup* – требуется ли открытие страницы поставщика идентификации в popup-окне (опционально).

Пример запроса:

```
POST /blitz/api/v2/users/current/fa/bind HTTP/1.1
Authorization: Basic ZG5ldm5pay10ZXN0Lm1vcy5ydTphUU56S0JuY2VBQVQwelg
Content-Type: application/json
Cache-Control: no-cache

{
    "fp": "vk:vk_1",
    "callback": "https://app.company.com/callback"
}
```

В ответ от Blitz Identity Provider будет получен параметр *redirectTo* с ссылкой, на которую необходимо направить пользователя в браузере для выполнения второго этапа и создания привязки учетной записи пользователя к социальной сети.

Пример ответа:

```
{
    "redirectTo":
"https://oauth.vk.com/authorize?state=5c415063-a153-424c-af9c-023a6bbf1892&scope=email&redirect_u
ri=https%3A%2F%2Flogin.company.com%2Fblitz%2Fapi%2Fusers%2Fcurrent%2Ffps%2Fbind%2Fcb%2Fvk%2Fvk_1&
client_id=5566286&v=5.52&response_type=code"
}
```

A.3.16. Удаление привязки социальной сети

Чтобы удалить привязку социальной сети к пользователю, необходимо вызвать методом DELETE сервис по адресу *https://login.company.com/blitz/api/v2/users/{subjectId}/fa/{sid}*, где в качестве параметров передается *guid* пользователя (*subjectId*) и идентификатор привязки (*sid*), полученный в пп. A.3.14.

Пример запроса:

```
DELETE /blitz/api/v2/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/fa/385956543 HTTP/1.1
Authorization: Basic ZG5ldm5pay10ZXN0Lm1vcy5ydTphUU56S0JuY2VBQVQwelg
Cache-Control: no-cache
```

A.3.17. Получение списка событий аудита пользователя

Для получения списка событий безопасности, зарегистрированных на учетную запись пользователя, необходимо методом GET вызвать сервис по адресу *https://login.company.com/blitz/api/v3/users/{subjectId}/audit*.

Необходимые разрешения: *blitz_api_uaud* или *blitz_api_sys_uaud*.

Запрос должен включать следующие URL-параметры:

- *rql* – указывается запрос фильтрации выводимых сведений. Поддерживается фильтрация по атрибуту *ts* (время события). Запрос имеет формат Resource Query Language (RQL)[51].Поддерживаются следующие операции:
  - *and* – одновременное выполнение поисковых условий;
  - *le* – проверка условия «меньше или равно»;
  - *ge* – проверка условия «больше или равно»;
  - *limit* – ограничение числа возвращаемых записей.
- *ua* – указывает требуемый вид вывода сведений о UserAgent (атрибут *ua*). Поддерживаются варианты:
  - *none* – не возвращать UserAgent;
  - *parsed* – возвращать UserAgent в разобранном виде (отдельно браузер и операционная система с указанием их версий);

Если параметр *ua* не указывать, то UserAgent (атрибут *ua*) вернется просто в виде строки.

В результате выполнения запроса Blitz Identity Provider вернет JSON, содержащий перечень событий аудита учетной записи за указанный период времени.

Пример запроса (без парсинга *ua*):

```
GET /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/audit?
rql=and(ge(ts,1637230238),le(ts,1637250238),limit(2)) HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
Cache-Control: no-cache
```

Пример ответа:

```
[
    {
        "sbj": "af583e70-fe39-407d-a87e-06cd0ec1830c",
        "ua": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) …",
        "ts": 1637250238015,
        "cAthM": "Basic",
        "ipCt": "Москва",
        "ipRad": 20,
        "cId": "test_app",
        "ip": 1406987879,
        "obj": "af583e70-fe39-407d-a87e-06cd0ec1830c",
```

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

```
        "ipSt": "Москва",
        "lpId": "test_app",
        "pid": "ddeebaba-2dc3-41bb-b539-7f0e472414a3",
        "ipLat": 55.7483,
        "prms": {
            "used_login": "test@yandex.ru",
            "auth_methods": "password",
            "authnDone": "true",
            "id_store": "389-ds"
        },
        "type": "login",
        "ipCtr": "Россия",
        "proc": "profile",
        "ipLng": 37.6171,
        "sid": "54914ac3-0d39-40d3-9617-92e0e7fe07ab"
    },
    …
]
```

Пример запроса (с парсингом *ua*):

```
GET /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/audit?
rql=and(ge(ts,1637230238),le(ts,1637250238),limit(2))&ua=parsed HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
Cache-Control: no-cache
```

Пример ответа:

```
[
    {
        "sbj": "af583e70-fe39-407d-a87e-06cd0ec1830c",
        "ua": {
            "broName": "Chrome",
            "broVer": "109",
            "deviceType": "pc",
            "raw": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) …",
            "osName": "macOS",
            "osVer": "10.15.7"
        },
        "ts": 1637250238015,
        "cAthM": "Basic",
        "ipCt": "Москва",
        "ipRad": 20,
        "cId": "test_app",
        "ip": 1406987879,
        "obj": "af583e70-fe39-407d-a87e-06cd0ec1830c",
        "ipSt": "Москва",
        "lpId": "test_app",
        "pid": "ddeebaba-2dc3-41bb-b539-7f0e472414a3",
        "ipLat": 55.7483,
        "prms": {
            "used_login": "test@yandex.ru",
            "auth_methods": "password",
            "authnDone": "true",
            "id_store": "389-ds"
        },
        "type": "login",
        "ipCtr": "Россия",
        "proc": "profile",
        "ipLng": 37.6171,
        "sid": "54914ac3-0d39-40d3-9617-92e0e7fe07ab"
    },
    …
]
```

A.3.18. Получение списка известных устройств пользователя

Для получения списка устройств пользователя необходимо методом GET вызвать сервис по адресу *https://login.company.com/blitz/api/v3/users/{subjectId}/uas*.

Необходимые разрешения: *blitz_api_uapps* или *blitz_api_sys_uapps*.

В результате выполнения запроса Blitz Identity Provider вернет JSON, содержащий перечень устройств пользователя.

Пример запроса:

```
GET /blitz/api/v3/users/af583e70-fe39-407d-a87e-06cd0ec1830c/uas HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
Cache-Control: no-cache
```

Пример ответа:

```
[
    {
        "name": "Chrome 96",
        "lastUsed": 1637249978,
        "tp": "Browser",
        "os": "macOS 10.15.7",
        "newlyCreated": false,
        "deviceType": "pc",
        "latestIp": "172.25.0.1",
        "subjectId": "af583e70-fe39-407d-a87e-06cd0ec1830c",
        "id": "SHA256_Z0x284K3qv313WViRuPfV5rglhDuYqSn4ztdxVKMBec",
        "trusted": false,
        "cls": true,
        "deviceId": "738f5ce91f912ddd4a0cc5fefa9e8c63",
        "device": "PC"
    },
    …
]
```

A.3.19. Удаления устройства пользователя из списка известных устройств

Для удаления устройства из числа запомненных необходимо методом DELETE вызвать сервис *https://login.company.com/blitz/api/v3/users/{subjectId}/uas/{id}*, передав в качестве *id* полученный в пп. A.3.18 идентификатор устройства.

Необходимые разрешения: *blitz_api_uapps_chg* или *blitz_api_sys_uapps_chg*.

При вызове в пользовательском режиме необходимо передать заголовки с IP-адресом пользователя и User-Agent.

Пример запроса:

```
DELETE /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/uas/SHA256_Z0x284K3qv313WViRuPfV5
rglhDuYqSn4ztdxVKMBec HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
Cache-Control: no-cache
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)…
```

A.3.20. Получение списка разрешений, выданных пользователем

Для получения списка выданных пользователем разрешений необходимо методом GET вызвать сервис по адресу *https://login.company.com/blitz/api/v3/users/{subjectId}/acls*.

Необходимые разрешения: *blitz_api_usec* или *blitz_api_sys_usec*.

В результате выполнения запроса Blitz Identity Provider вернет JSON, содержащий перечень устройств пользователя.

Пример запроса:

```
GET /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/acls HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
Cache-Control: no-cache
```

Пример ответа:

```
[
    {
        "id": "d2580c98-e584-4aad-a591-97a8cf45cd2a_app1",
```

```
        "updated": 1552896932780,
        "client_id": "app1",
        "scopes": [
            "openid",
            "profile",
        ]
    }
]
```

A.3.21. Отзыв выданного пользователем разрешения

Для отзыва выданного разрешения необходимо методом DELETE вызвать сервис *https://login.company.com/blitz/api/v3/users/{subjectId}/acls/{acl_id}*, передав в качестве *acl_id* полученный в пп. A.3.20 идентификатор (*id*) разрешения.

Необходимые разрешения: *blitz_api_usec_chg* или *blitz_api_sys_usec_chg*.

При вызове в пользовательском режиме необходимо передать заголовки с IP-адресом пользователя и User-Agent.

Пример запроса:

```
DELETE /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/acls/
d2580c98-e584-4aad-a591-97a8cf45cd2a_app1 HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
Cache-Control: no-cache
```

A.3.22. Получение списка привязанных мобильных приложений

Для получения списка привязанных мобильных приложений необходимо методом GET вызвать сервис по адресу *https://login.company.com/blitz/api/v3/users/{subjectId}/apps*.

Необходимые разрешения: *blitz_api_uapps* или *blitz_api_sys_uapps*.

В результате выполнения запроса Blitz Identity Provider вернет JSON, содержащий перечень устройств пользователя.

Пример запроса:

```
GET /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/apps HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
Cache-Control: no-cache
```

Пример ответа:

```
[
    {
        "id": "dyn~test_app~afae0cab-2649-482d-9832-5f73816afb59",
        "name": {
            "_default_": "Тестовое приложение (test_app)"
        },
        "availableScopes": [
            "openid",
            "profile"
        ],
        "softwareId": "test_app"
    }
]
```

A.3.23. Отвязка от учетной записи мобильного приложения

Для отзыва выданного разрешения необходимо методом DELETE вызвать сервис *https://login.company.com/blitz/api/v3/users/{subjectId}/apps/{app_id}*, передав в качестве *app_id* полученный в пп. A.3.22 идентификатор (*id*) привязки приложения.

Необходимые разрешения: *blitz_api_uapps_chg* или *blitz_api_sys_uapps_chg*.

При вызове в пользовательском режиме необходимо передать заголовки с IP-адресом пользователя и User-Agent.

Пример запроса:

```
DELETE /blitz/api/v3/users/d2580c98-e584-4aad-a591-97a8cf45cd2a/apps/
d2580c98-e584-4aad-a591-97a8cf45cd2a_app1 HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
X-Forwarded-For: 200.200.100.100
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)…
Cache-Control: no-cache
```

A.3.24. Удаление учетной записи пользователя

Для удаления учетной записи пользователя необходимо методом DELETE вызвать сервис *https://login.company.com/blitz/api/v2/users/{subjectId}?instanceId={instanceId}*, передав в subjectId идентификатор удаляемой учетной записи, и в параметре instanceId ссылку на удаляемую учетную запись.

Чтобы узнать значение *instanceId* для пользователя, необходимо предварительно вызвать методом GET сервис получения атрибутов пользователя (A.3.3).

Пример запроса:

```
DELETE /blitz/api/v2/users/d2580c98-e584-4aad-a591-97a8cf45cd2a?instanceId=Mzg…nU HTTP/1.1
Authorization: Basic YXBwX2lkOmFwcF9zZWNyZXQ=
```

A.4. Сервисы для работы с группами пользователей

Blitz Identity Provider предоставляет системе сервисы для выполнения действий с группами пользователей:

- получение атрибутов группы.
- создание группы.
- изменение атрибутов группы.
- удаление группы.
- получение списка пользователей группы.
- добавление пользователей в группу.
- исключение пользователей из группы.

Для вызова сервисов система должна получить маркер доступа на системное разрешение *blitz_groups* (инструкцию в пп. A.2) и включать его во все вызываемые сервисы, описанные в следующих подразделах.

Список атрибутов групп пользователей приведен в качестве образца. Содержание списка необходимо скорректировать в зависимости от конкретных настроек, сделанных при внедрении Blitz Identity Provider. См. «Руководство администратора».

Группы в Blitz Identity Provider описываются следующими атрибутами:

- *id* – идентификатор группы в Blitz Identity Provider;

– *name* – наименование группы пользователей.

A.4.1. Получение атрибутов группы

Если известен id группы, то получение атрибутов группы выполняется с помощью вызова методом GET сервиса по адресу *https://login.company.com/blitz/api/v2/grps/{id}*. Запрос должен включать следующие URL-параметры:

– *profile* – имя профиля групп пользователей (например, *orgs*);

– *expand* – значение true, указывающее, что необходимо вернуть все атрибуты группы.

Пример запроса получения атрибутов группы:

```
GET /blitz/api/v2/grps/14339e8e-a665-4556-92f1-5c348eff6696?
profile=orgs HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
Cache-Control: no-cache
```

Пример ответа:

```
{
    "instanceId": "Mzg…nU",
    "id": "14339e8e-a665-4556-92f1-5c348eff6696",
    "OGRN": "1234567890329",
    "INN": "7743151614",
    "name": "ООО Тестовая компания",
    "profile": "orgs"
}
```

Если не известен id группы, то поиск группы и получение ее атрибутов выполняется с помощью вызова методом GET сервиса по адресу *https://login.company.com/blitz/api/v2/grps*. Запрос должен включать следующие URL-параметры:

– *profile* – имя профиля групп пользователей;

– *rql* – указывается поисковый запрос по атрибутам группы. Запрос имеет формат Resource Query Language (RQL)[52].Поддерживаются следующие операции:

- *and* – одновременное выполнение поисковых условий;
- *or* – альтернативное выполнение поисковых условий (например, поиск по разным атрибутам);
- *eq* – проверка условия равенства;
- *limit* – ограничение числа возвращаемых записей.

– *expand* (необязательный параметр) – требуется ли включать в полученный ответ атрибуты групп (true) или достаточно только вернуть идентификаторы найденных группы (false).

В результате выполнения запроса Blitz Identity Provider вернет JSON, содержащий перечень групп, удовлетворяющих заданным поисковым условиям. Группы будут

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

возвращены с указанием их идентификатора (*id*), а также значения остальных атрибутов групп (в случае expand=true).

Пример запроса с поиском групп по атрибутам (ОГРН или ИНН в примере):

```
GET /blitz/api/v2/grps?profile=orgs&expand=true&rql=
or(eq(OGRN,string:1230123456789),eq(INN,string:7743151614)) HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
Cache-Control: no-cache
```

Пример ответа:

```
[
    {
        "instanceId": "Mzg5L…nU",
        "id": "14339e8e-a665-4556-92f1-5c348eff6696",
        "OGRN": "1234567890329",
        "INN": "7743151614",
        "name": "ООО Тестовая компания",
        "profile": "orgs"
    }
]
```

A.4.2. Создание группы пользователей

Для создания группы пользователей необходимо вызвать методом POST сервис по адресу *https://login.company.com/blitz/api/v2/grps*.

Тело запроса должно содержать следующие параметры:

- *profile* – имя профиля групп пользователей;
- *id* – уникальный идентификатор группы;
- {остальные атрибуты группы и их значения}

Пример запроса:

```
POST /blitz/api/v2/grps HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
Cache-Control: no-cache

{
    "id":"95339e8e-a665-4556-92f1-5c348eff6696",
    "OGRN":"9876543210321",
    "INN":"5012345678",
    "name":"ООО Тестовая компания 2",
    "profile":"orgs"
}
```

Пример ответа:

```
{
    "instanceId": "b3Jnc…dQ",
    "name": "ООО Тестовая компания 2",
    "OGRN": "9876543210321",
    "id": "95339e8e-a665-4556-92f1-5c348eff6696",
    "profile": "orgs",
    "INN": "5012345678"
}
```

A.4.3. Изменение атрибутов группы пользователей

Для изменения атрибутов группы необходимо вызвать методом POST сервис по адресу *https://login.company.com/blitz/api/v2/grps/{id}?profile=orgs* с новым набором атрибутов.

Тело запроса должно содержать следующие параметры:

- *profile* – имя профиля групп (должно быть передано и в составе URL, и в теле запроса);
- *id* – идентификатор группы;
- {остальные атрибуты группы и их значения}

Пример запроса:

```
POST /blitz/api/v2/grps/5f7b0580-cd2e-4146-8fc5-6eb5a95c7b42? profile=orgs HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
Cache-Control: no-cache

{
    "id": "5f7b0580-cd2e-4146-8fc5-6eb5a95c7b42",
    "OGRN": "1147746651733",
    "INN": "7715434658",
    "name": "Новое название",
    "profile": "orgs"
}
```

Пример ответа:

```
{
    "instanceId": "Mzg5L…nU",
    "id": "5f7b0580-cd2e-4146-8fc5-6eb5a95c7b42",
    "OGRN": "1147746651733",
    "INN": "7715434658",
    "name": "Новое название",
    "profile": "orgs"
}
```

Привет ответа с ошибкой (организация не существует):

```
{
    "errors": [
        {
            "code": "group_not_found",
            "desc": "Group with '95339e8e-…97' id not found in '389-ds' LDAP group store",
            "params": {}
        }
    ]
}
```

A.4.4. Удаление группы пользователей

Для удаления группы необходимо вызвать методом DELETE сервис по адресу *https://login.company.com/blitz/api/v2/grps/{id}?profile=orgs* без передачи тела запроса.

Пример запроса:

```
DELETE /blitz/api/v2/grps/5f7b0580-cd2e-4146-8fc5-6eb5a95c7b42? profile=orgs HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
Cache-Control: no-cache
```

A.4.5. Получение списка пользователей в группе пользователей

Для получения списка пользователей из группы пользователей необходимо вызвать методом GET сервис по адресу *https://login.company.com/blitz/api/v2/grps/{id}/members*. Запрос должен включать следующие URL-параметры:

- *profile* – имя профиля групп пользователей;
- *expand* (необязательный параметр) – требуется ли включать в полученный ответ ФИО пользователя (true) или достаточно только вернуть их идентификаторы (false).

Пример запроса:

```
GET /blitz/api/v2/grps/14339e8e-a665-4556-92f1-5c348eff6696/members? profile=orgs&expand=false
HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
Cache-Control: no-cache
```

Пример ответа (при expand=false):

```
[
    {
        "instanceId": "Mzg5L…J1",
        "subjectId": "d434b7d4-9816-460a-83aa-0a994226cbe7"
    },
    {
        "instanceId": "Mzg5L…J1",
        "subjectId": "2cafa5f4-bc84-4f6f-91aa-080da47975f0"
    }
]
```

Пример ответа (при expand=true):

```
[
    {
        "instanceId": "Mzg5L…J1",
        "family_name": "Иванов",
        "middle_name": "Иванович",
        "given_name": "Иван",
        "subjectId": "d434b7d4-9816-460a-83aa-0a994226cbe7"
    },
    {
        "instanceId": "Mzg5L…J1",
        "family_name": "Сергеев",
        "middle_name": "Сергеевич",
        "given_name": "Сергей",
        "subjectId": "2cafa5f4-bc84-4f6f-91aa-080da47975f0"
    }
]
```

A.4.6. Включение пользователей в группу

Для добавления пользователей в группу необходимо вызвать методом POST сервис по адресу *https://login.company.com/blitz/api/v2/grps/{id}/members/add?profile=orgs*.

Тело запроса должно содержать список добавляемых в группу пользователей с указанием их идентификаторов (*sub)* в атрибуте *subjectId*.

Пример запроса:

```
POST /blitz/api/v2/grps/5f7b0580-cd2e-4146-8fc5-6eb5a95c7b42/members/add? profile=orgs HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
Cache-Control: no-cache

[
    {
        "subjectId": "45ff69f2-6c40-418f-a21d-cbe6f07b88c9"
    },
    {
        "subjectId": "cc8c4589-b2f8-40b8-b351-36d643808943"
    }
]
```

Пример ответа (при успешном исполнении):

```
[
    {
        "instanceId": "Mzg5L…J1",
        "storeId": "tam",
        "subjectId": "45ff69f2-6c40-418f-a21d-cbe6f07b88c9"
    },
    {
        "instanceId": "Nzg5L…J1",
        "storeId": "tam",
        "subjectId": "cc8c4589-b2f8-40b8-b351-36d643808943"
```

```
    }
]
```

Пример ответа при ошибке (попытка добавить несуществующего пользователя):

```
{
    "errors": [
        {
            "code": "user_not_found",
            "desc": "User with subjectId 'd2580c98-e584-4aad-a591-97a8cf45cd2q' not found",
            "params": {}
        }
    ]
}
```

Пример ответа при ошибке (попытка добавить того, кто уже есть в группе):

```
{
    "errors": [
        {
            "code": "some_members_already_in_group",
            "desc": "Some of adding members are already included in group",
            "params": {}
        }
    ]
}
```

A.4.7. Исключение пользователей из группы

Для исключения пользователей из группы необходимо вызвать методом POST сервис по адресу *https://login.company.com/blitz/api/v2/grps/{id}/members/rm?profile=orgs*.

Тело запроса должно содержать список исключаемых из организации доверенных лиц с указанием их идентификаторов (*sub*) в атрибуте *subjectId*.

Пример запроса:

```
POST /blitz/api/v2/grps/5f7b0580-cd2e-4146-8fc5-6eb5a95c7b42/members/rm? profile=orgs HTTP/1.1
Authorization: Bearer cNwIXatB0wk5ZHO0xG5kxuuLubesWcb_yPPqLOFWDuwzMDc0Nz
Content-Type: application/json
Cache-Control: no-cache

[
    {
        "subjectId": "d2580c98-e584-4aad-a591-97a8cf45cd2a"
    }
]
```

Пример ответа (при успехе):

```
[
    {
        "instanceId": "Mzg5L…J1",
        "storeId": "389-ds",
        "subjectId": "d2580c98-e584-4aad-a591-97a8cf45cd2a"
    }
]
```

Пример ответа (при попытке удалить из группы того, кого там уже нет):

```
{
    "errors": [
        {
            "code": "some_members_not_in_group",
            "desc": "Some of removing members are not included in group",
            "params": {}
        }
    ]
}
```

Пример ответа (при попытке удалить несуществующего пользователя):

```
{
    "errors": [
        {
            "code": "user_not_found",
            "desc": "User with subjectId 'd2580c98-e584-4aad-a591-97a8cf45cd2b' not found",
            "params": {}
```

```
        }
    ]
}
```

A.5. Сервисы для работы с правами доступа

Для выполнения запросов по просмотру, назначению, отзыву прав доступа приложение должно получить маркер доступа с системным разрешением *blitz_rights_full_access*.

Право доступа назначается от субъекта доступа к объекту доступа. В качестве субъектов доступа могут выступать пользователи и приложения (префикс *its*). В качестве объектов доступа могут выступать пользователи, группы пользователей (префикс *grps*), приложения (префикс *its*).

Доступны следующие сервисы:

- получить по субъекту доступа перечень его прав доступа к различным объектам доступа;
- получить по объекту доступа перечень прав доступа к нему от субъектов доступа;
- назначить субъекту доступа право доступа к объекту доступа;
- отозвать у субъекта доступа право доступа к объекту доступа.

A.5.1. Получение прав доступа

Получение прав доступа по субъекту доступа, являющемуся пользователем, выполняется с помощью GET сервиса *https://login.company.com/blitz/api/v3/rights/of/<sub>*.

Пример запроса:

```
GET /blitz/api/v3/rights/of/BIP-1SEQ41A HTTP/1.1
Content-Type: application/json
Authorization: Bearer cNwIX…Nz
```

Пример ответа сервиса (пользователь *BIP-1SEQ41A* имеет право *ORG_ADMIN* к группе пользователей *1147746651733*, право *APP_ADMIN* к приложению *test_app2*, право *change_password* к учетной записи пользователя *BIP-3SGR7TA*):

```
{
    "grps|1147746651733|orgs": {
        "ORG_ADMIN": [
            "set_from_api",
            "another_one_tag"
        ]
    },
    "its|test_app2": {
        "APP_ADMIN": [
            "set_from_api"
        ]
    },
    "BIP-3SGR7TA": {
        "change_password": [
            "parent"
        ]
    }
}
```

Получение прав доступа по субъекту доступа, являющемуся приложением, выполняется с помощью GET сервиса *https://login.company.com/blitz/api/v3/rights/of/its/<app_id>*.

Пример запроса:

```
GET /blitz/api/v3/rights/of/its/test_app HTTP/1.1
Content-Type: application/json
Authorization: Bearer cNwIX…Nz
```

Пример ответа сервиса (приложение *test_app* имеет право *SYS_MON* к приложению *test_app2*, право *change_password* к учетной записи пользователя *BIP-3SGR7TA*, право *ORG_ADMIN* к группе пользователей *1147746651733*):

```
{
    "its|test_app2": {
        "SYS_MON": [
            "set_from_api"
        ]
    },
    "BIP-3SGR7TA": {
        "change_password": [
            "set_from_api"
        ]
    },
    "grps|1147746651733|orgs": {
        "ORG_ADMIN": [
            "set_from_api"
        ]
    }
}
```

Получение прав доступа по объекту доступа, являющемуся пользователем, выполняется с помощью GET сервиса *https://login.company.com/blitz/api/v3/rights/on/<sub>*.

Пример запроса:

```
GET /blitz/api/v3/rights/on/BIP-3SGR7TA HTTP/1.1
Content-Type: application/json
Authorization: Bearer cNwIX…Nz
```

Пример ответа сервиса (на учетную запись *BIP-3SGR7TA* у пользователя *BIP-1SEQ41A,* и у приложения *test_app* есть право *change_password*):

```
{
    "BIP-1SEQ41A": [
        "change_password"
    ],
    "its|test_app": [
        "change_password"
    ]
}
```

Получение прав доступа по объекту доступа, являющемуся группой, выполняется с помощью GET сервиса *https://login.company.com/blitz/api/v3/rights/on/grps/<grp_id>?objectExt=<profile>*.

Пример запроса:

```
GET /blitz/api/v3/rights/on/grps/1147746651733?objectExt=orgs HTTP/1.1
Content-Type: application/json
Authorization: Bearer cNwIX…Nz
```

Пример ответа сервиса (на учетную запись группы *1147746651733* из профиля *orgs* у пользователя *BIP-1SEQ41A,* и у приложения *test_app* есть право *ORG_ADMIN*):

```
{
    "BIP-1SEQ41A": [
        "ORG_ADMIN"
    ],
    "its|test_app": [
```

```
        "ORG_ADMIN"
    ]
}
```

Получение прав доступа по объекту доступа, являющемуся приложением, выполняется с помощью GET сервиса *https://login.company.com/blitz/api/v3/rights/on/its/<app_id>*.

Пример запроса:

```
GET /blitz/api/v3/rights/on/its/test_app2 HTTP/1.1
Content-Type: application/json
Authorization: Bearer cNwIX…Nz
```

Пример ответа сервиса (на учетную запись приложения *test_app2* у пользователя *BIP-1SEQ41A* есть право *APP_ADMIN,* и у приложения *test_app* есть право *SYS_MON*):

```
{
    "BIP-1SEQ41A": [
        "APP_ADMIN"
    ],
    "its|test_app": [
        "SYS_MON"
    ]
}
```

В случае если маркер доступа просрочен, то сервис вернет ошибку *HTTP 401 Unauthorized*:

```
{
    "type": "security_error",
    "error": "bad_access_token",
    "desc": "expired_access_token"
}
```

A.5.2. Назначение прав доступа

Назначение права доступа выполняется через PUT вызов сервиса *https://login.company.com/blitz/api/v3/rights*.

Тело запроса должно содержать следующие параметры:

- *subject* – идентификатор субъекта, которому назначается право (идентификатор пользователя или приложения);
- *subjectType* – тип субъекта. Параметр указывается только в случае назначения права приложению. В этом случае используется значение *its*;
- *object* – идентификатор объекта, на который назначается право (идентификатор пользователя, группы пользователей или приложения);
- *objectType* – тип объекта. Параметр указывается только в случае назначения права на группу пользователей (значение *grps*) или на приложение (значение *its*);
- *rights* – массив со списком назначаемых прав субъекту на объект;
- *tags* (не обязательный атрибут) – массив со списком тэгов назначаемых прав.

Пример запроса на назначение права доступа пользователю на другого пользователя:

```
PUT /blitz/api/v3/rights
Authorization: Bearer cNwIX…Nz
Content-Type: application/json
Cache-Control: no-cache

[{
    "subject": "BIP-1SEQ41A",
```

```
    "object": "BIP-3SGR7TA",
    "rights": ["change_password"],
    "tags": ["set_from_api"]
}]
```

Пример запроса на назначение права доступа пользователю на группу:

```
PUT /blitz/api/v3/rights
Authorization: Bearer cNwIX…Nz
Content-Type: application/json
Cache-Control: no-cache

[{
    "subject": "BIP-1SEQ41A",
    "object": "1147746651733",
    "objectType": "grps",
    "rights": ["ORG_ADMIN"],
    "tags": ["set_from_api"]
}]
```

Пример запроса на назначение права доступа пользователю на приложение:

```
PUT /blitz/api/v3/rights
Authorization: Bearer cNwIX…Nz
Content-Type: application/json
Cache-Control: no-cache

[{
    "subject": "BIP-1SEQ41A",
    "object": "test_app2",
    "objectType": "its",
    "rights": ["APP_ADMIN"],
    "tags": ["set_from_api"]
}]
```

Пример запроса на назначение права доступа приложению на пользователя:

```
PUT /blitz/api/v3/rights
Authorization: Bearer cNwIX…Nz
Content-Type: application/json
Cache-Control: no-cache

[{
    "subject": "test_app",
    "subjectType": "its",
    "object": "BIP-3SGR7TA",
    "rights": ["change_password"],
    "tags": ["set_from_api"]
}]
```

Пример запроса на назначение права доступа приложению на группу:

```
PUT /blitz/api/v3/rights
Authorization: Bearer cNwIX…Nz
Content-Type: application/json
Cache-Control: no-cache

[{
    "subject": "test_app",
    "subjectType": "its",
    "object": "1147746651733",
    "objectType": "grps",
    "rights": ["ORG_ADMIN"],
    "tags": ["set_from_api"]
}]
```

Пример запроса на назначение права доступа приложению на другое приложение:

```
PUT /blitz/api/v3/rights
Authorization: Bearer cNwIX…Nz
Content-Type: application/json
Cache-Control: no-cache

[{
    "subject": "test_app",
    "subjectType": "its",
    "object": "test_app2",
    "objectType": "its",
    "rights": ["SYS_MON"],
    "tags": ["set_from_api"]
}]
```

В случае успешного назначения права доступа сервис вернет *HTTP 204 No Content*.

В случае если маркер доступа просрочен, то сервис вернет ошибку *HTTP 401 Unauthorized*:

```
{
    "type": "security_error",
    "error": "bad_access_token",
    "desc": "expired_access_token"
}
```

В случае если назначаемого права не существует, то сервис вернет ошибку *HTTP 400 Bad Request*:

```
{
    "type": "process_error",
    "error": "unknown_right",
    "desc": "The specified right is unknown",
    "params": {
        "right": "change_password1"
    }
}
```

В случае если указанного в качестве субъекта или объекта пользователя не существует, то сервис вернет ошибку *HTTP 400 Bad Request*:

```
{
    "type": "process_error",
    "error": "unknown_user",
    "desc": "The specified user is unknown",
    "params": {
        "userId": "ivanov1"
    }
}
```

В случае если указанной в качестве объекта группы не существует, то сервис вернет ошибку *HTTP 400 Bad Request*:

```
{
    "type": "process_error",
    "error": "unknown_group",
    "desc": "The specified group is unknown",
    "params": {
        "grpId": "1147746651734"
    }
}
```

В случае если указанной в качестве субъекта или объекта приложения не существует, то сервис вернет ошибку HTTP *400 Bad Request*:

```
{
    "type": "process_error",
    "error": "unknown_rp",
    "desc": "The specified relying party is unknown",
    "params": {
        "rpId": "test_app3"
    }
}
```

A.5.3. Отзыв прав доступа

Отзыв права доступа выполняется через DELETE вызов сервиса *https://login.company.com/blitz/api/v3/rights*.

Тело запроса должно содержать следующие параметры:

- *subject* – идентификатор субъекта, у которого отзывается право (идентификатор пользователя или приложения);
- *subjectType* – тип субъекта. Параметр указывается только в случае отзыва права у приложения. В этом случае используется значение *its*;

- *object* – идентификатор объекта, на который отзывается право (идентификатор пользователя, группы пользователей или приложения);
- *objectType* – тип объекта. Параметр указывается только в случае отзыва права на группу пользователей (значение *grps*) или на приложение (значение *its*);
- *rights* – массив со списком отзываемых прав субъекта на объект;
- *tags* (не обязательный атрибут) – массив со списком тэгов отзываемых прав.

Если право доступа было назначено субъекту доступа на объект доступа с указанием нескольких тэгов, то для отзыва права доступа тоже необходимо указать все тэги. Если отзыв права доступа вызывается не с полным указанием тэгов, то при отзыве будут удалены только отзываемые тэги, а право доступа у субъекта доступа к объекту доступа останется, пока остается хотя бы один из тэгов.

Пример запроса на отзыв права доступа пользователю на другого пользователя:

```
PUT /blitz/api/v3/rights
Authorization: Bearer cNwIX…Nz
Content-Type: application/json
Cache-Control: no-cache

[{
    "subject": "BIP-1SEQ41A",
    "object": "BIP-3SGR7TA",
    "rights": ["change_password"],
    "tags": ["set_from_api"]
}]
```

Пример запроса на отзыв права доступа пользователю на группу:

```
PUT /blitz/api/v3/rights
Authorization: Bearer cNwIX…Nz
Content-Type: application/json
Cache-Control: no-cache

[{
    "subject": "BIP-1SEQ41A",
    "object": "1147746651733",
    "objectType": "grps",
    "rights": ["ORG_ADMIN"],
    "tags": ["set_from_api"]
}]
```

Пример запроса на отзыв права доступа пользователю на приложение:

```
PUT /blitz/api/v3/rights
Authorization: Bearer cNwIX…Nz
Content-Type: application/json
Cache-Control: no-cache

[{
    "subject": "BIP-1SEQ41A",
    "object": "test_app2",
    "objectType": "its",
    "rights": ["APP_ADMIN"],
    "tags": ["set_from_api"]
}]
```

Пример запроса на отзыв права доступа приложению на пользователя:

```
PUT /blitz/api/v3/rights
Authorization: Bearer cNwIX…Nz
Content-Type: application/json
Cache-Control: no-cache

[{
    "subject": "test_app",
    "subjectType": "its",
    "object": "BIP-3SGR7TA",
    "rights": ["change_password"],
```

```
    "tags": ["set_from_api"]
}]
```

Пример запроса на отзыв права доступа приложению на группу:

```
PUT /blitz/api/v3/rights
Authorization: Bearer cNwIX…Nz
Content-Type: application/json
Cache-Control: no-cache

[{
    "subject": "test_app",
    "subjectType": "its",
    "object": "1147746651733",
    "objectType": "grps",
    "rights": ["ORG_ADMIN"],
    "tags": ["set_from_api"]
}]
```

Пример запроса на отзыв права доступа приложению на другое приложение:

```
PUT /blitz/api/v3/rights
Authorization: Bearer cNwIX…Nz
Content-Type: application/json
Cache-Control: no-cache

[{
    "subject": "test_app",
    "subjectType": "its",
    "object": "test_app2",
    "objectType": "its",
    "rights": ["SYS_MON"],
    "tags": ["set_from_api"]
}]
```

В случае успешного отзыва права доступа сервис вернет *HTTP 204 No Content*.

В случае если маркер доступа просрочен, то сервис вернет ошибку *HTTP 401 Unauthorized*:

```
{
    "type": "security_error",
    "error": "bad_access_token",
    "desc": "expired_access_token"
}
```

В случае если отзываемого права не существует, то сервис вернет ошибку *HTTP 400 Bad Request*:

```
{
    "type": "process_error",
    "error": "unknown_right",
    "desc": "The specified right is unknown",
    "params": {
        "right": "change_password1"
    }
}
```

В случае если указанного в качестве субъекта или объекта пользователя не существует, то сервис вернет ошибку *HTTP 400 Bad Request*:

```
{
    "type": "process_error",
    "error": "unknown_user",
    "desc": "The specified user is unknown",
    "params": {
        "userId": "ivanov1"
    }
}
```

В случае если указанной в качестве объекта группы не существует, то сервис вернет ошибку *HTTP 400 Bad Request*:

```
{
    "type": "process_error",
    "error": "unknown_group",
    "desc": "The specified group is unknown",
    "params": {
        "grpId": "1147746651734"
```

```
    }
}
```

В случае если указанной в качестве субъекта или объекта приложения не существует, то сервис вернет ошибку *HTTP 400 Bad Request*:

```
{
    "type": "process_error",
    "error": "unknown_rp",
    "desc": "The specified relying party is unknown",
    "params": {
        "rpId": "test_app3"
    }
}
```

A.5.4. Управление правами между ведущим и ведомым пользователями

Назначение права ведущему пользователю в отношении ведомого пользователя осуществляется методом POST по адресу *https://login.company.com/blitz/api/v2/users/rights/change*.

В запрос должен быть добавлен заголовок с маркером доступа на разрешение с именем *blitz_user_rights*, полученным учетной записью ведущего пользователя.

Если назначаются права, то тело запроса должно содержать заполненный блок update с перечнем прав, которые должны быть добавлены в результате выполнения операции. Каждое право описывается следующими параметрами:

- *subject* – идентификатор (*sub*) учетной записи ведущего пользователя;
- *object* – идентификатор (*sub*) учетной записи ведомого пользователя;
- *rights* – перечень прав в виде массива, который получает учетная запись ведущего пользователя в отношении учетной записи ведомого пользователя. Например, для права менять пароль от учетной записи должно быть указано право *change_password* (смена пароля);
- *tags* – перечень тегов, указывающих на основания, по которым данный пользователь получил права.

Если отзываются права, то тело запроса должно содержать заполненный блок delete с перечнем прав, которые должны быть отозваны в результате выполнения операции. Каждое право описывается следующими параметрами:

- *subject* – идентификатор (*sub*) учетной записи ведущего пользователя;
- *object* – идентификатор (*sub*) учетной записи ведомого пользователя;
- *rights* – перечень прав в виде массива, которые отзываются у ведущей учетной записи в отношении ведомой учетной записи;
- *tags* – перечень тегов, указывающих на основания, по которым данный пользователь получил права.

Если при выполнении запроса права не назначаются или не отзываются, то в теле запроса должен соответственно присутствовать или пустой блок *update*, или пустой блок

delete.

В одном запросе может быть указано сразу несколько назначаемых/отзываемых прав, но в качестве субъекта (*subject*) должен быть указан только тот пользователь, на которого был получен маркер доступа, используемый для вызова сервиса.

Пример запроса на назначение прав:

```
POST /blitz/api/v2/users/rights/change HTTP/1.1
Authorization: Bearer cNwIX…Tg
Content-Type: application/json
Cache-Control: no-cache
{
   "update":[
      {
         "subject":"6561d0d9-5583-4bb5-a681-b591358e5fcd",
         "object":"5cffd68f-2cb8-4f7a-b0f3-9fa69a1fbbcd",
         "rights":[
            "change_password"
         ],
         "tags":[
            "parent"
         ]
      },
      {
         "subject":"6561d0d9-5583-4bb5-a681-b591358e5fcd",
         "object":"b855957d-bf24-48d4-bb63-cce4f5064590d",
         "rights":[
            "change_password"
         ],
         "tags":[
            "parent"
         ]
      }
   ],
   "delete":[
   ]
}
```

Пример запроса на отзыв прав:

```
POST /blitz/api/v2/users/rights/change HTTP/1.1
Authorization: Bearer cNwIX…Tg
Content-Type: application/json
Cache-Control: no-cache
{
   "update":[
   ],
   "delete":[
      {
         "subject":"b855957d-bf24-48d4-bb63-cce4f5064590d",
         "object":"5cffd68f-2cb8-4f7a-b0f3-9fa69a1fbbcd",
         "rights":[
            "change_password"
         ],
         "tags":[
            "parent"
         ]
      }
   ]
}
```

Запрос может завершиться ошибкой. В этом случае Blitz Identity Provider отклонит запрос целиком и вернет перечень возникших ошибок. Пример ответа с ошибками:

```
{
   "errors" : [
      {
         "code" : "validation_error",
         "params" : {},
         "desc" : "(For subject 'dea75b73-a2ba-4b60-a41c-bb640968826b') Incorrect right '' to
object '5cffd68f-2cb8-4f7a-b0f3-9fa69a1fbbcd'"
      },
      {
         "params" : {},
```

```
         "code" : "validation_error",
         "desc" : "(For subject 'dea75b73-a2ba-4b60-a41c-bb640968826b') Incorrect tag '' for
right 'write' to object '5cffd68f-2cb8-4f7a-b0f3-9fa69a1fbbcd'"
      },
      {
         "desc" : "(For subject 'dea75b73-a2ba-4b60-a41c-bb640968826b') Incorrect object ''",
         "code" : "validation_error",
         "params" : {}
      },
      {
         "desc" : "Incorrect subject ''",
         "code" : "validation_error",
         "params" : {}
      }
   ]
}
```


Запрос на отзыв прав может быть выполнен приложением не только с использованием пользовательского маркера доступа, полученного на разрешение с именем *blitz_user_rights*, но и с использованием системного маркера доступа, полученного на разрешение с именем *blitz_rm_rights*. В этом случае запрос на отзыв может включать *subject* любых пользователей (для отзыва у пользователя права не потребуется, чтобы именно этот пользователь осуществлял вход в систему и получал маркер доступа – система может отзывать права любого пользователя).

Приложение Б. Добавление дополнительного метода аутентификации

Blitz Identity Provider позволяет подключить собственный разработанный метод аутентификации. Для этого необходимо система, выступающая в качестве «поставщика» такого метода аутентификации, должна:

- Предоставить обработчик запроса на аутентификацию.
- Передать в Blitz Identity Provider результат аутентификации.

Также система может опционально предоставить сервис проверки применимости метода аутентификации.

В Blitz Identity Provider разработанный метод аутентификации нужно зарегистрировать как «Внешний метод аутентификации» (см. «Руководство администратора»).

Б.1. Сервис обработчика запроса на аутентификацию

Представляет собой URL для приема HTTP-запросов от Blitz Identity Provider. При запросе на аутентификацию Blitz Identity Provider будет делать запрос методом POST по данному адресу. В теле данного запроса Blitz Identity Provider в формате JSON передаст следующие данные:

- идентификатор запроса (id);
- утверждения, характеризующие пользователя (claims) – опционально, только при вызове в качестве второго фактора;
- идентификатор системы, запросивший вход (rpId);
- идентификатор контекста аутентификации (loginContextId);
- данные о запросе (request), включающий в себя заголовки (headers), IP-адрес пользователя (remoteAddress), адрес метода (uri), перечень cookie (cookies) и User Agent пользователя (userAgent).

Пример тела запроса:

```
{
    "id": "a9692091-4613-41aa-91d2-9a71a3fc2e07",
    "claims": {},
    "rpId": "_blitz_profile",
    "loginContextId": "4502aa51-f28c-4a64-951c-5ab1e77b1294",
    "request": {
      "headers": {},
      "remoteAddress": "172.25.0.1",
      "uri": "/blitz/login/methods2/outside_test",
      "cookies": {},
      "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:80.0)…"
    }
}
```

На стороне «поставщика» внешнего метода необходимо предусмотреть обработку данного запроса. В результате внешний метод должен вернуть либо HTTP-ответ для выполнения в браузере пользователя (например, вернуть код HTML-страницы или

инициировать редирект браузера на необходимую страницу внешнего метода), либо сообщение об ошибке.

Требования к HTTP-ответу на запрос:

- ответ должен включать в себя установку cookie (на общий домен Blitz Identity Provider и внешнего метода);
- название cookie должно быть предварительно зарегистрировано в Blitz Identity Provider;
- в качестве значения cookie должен быть использован идентификатор сессии, сгенерированный внешним методом.

Пример HTTP-ответа с редиректом и установкой cookie:

```
HTTP/1.1 302 Found
Location: https://login.company.com/blitz/begin?id=a9692091-4613-41aa-91d2
Set-Cookie: Bmr=YTk2OTIwOTEtNDYxMy00MWFhLTkxZDItOWE3MWEzZmMyZTA3; Domain=company.com;
path=/blitz; Secure; HttpOnly
```

При прохождении внешнего метода «поставщик» должен проверить, что значение cookie для данного запроса не было изменено.

В случае невозможности провести аутентификацию пользователя, внешний метод должен вернуть ошибку. Пример рекомендуемых к возврату ошибок приведен в таблице ниже.

Таблица 10 – Рекомендуемые кода возврата

№	Код ответа HTTP	Значение ответа	Описание ответа
1	200	OK	Инициирование внешнего метода посредством отображения контента страницы
2	302	Found	Инициирование внешнего метода посредством редиректа
3	400	Bad Request	Отсутствуют обязательные параметры запроса
4	500	Internal Server Error	Внутренняя ошибка обработки входящего запроса

Б.2. Передача результата аутентификации

После прохождения внешнего метода «поставщик» должен выполнить следующие действия:

а) серверная часть «поставщика» должна вызвать Blitz Identity Provider методом POST по следующему адресу:

```
https://login.company.com/blitz/login/methods/outside/save?methodName=outside_{name}
```

В данном запросе *name* – это имя внешнего метода, присвоенное ему в Blitz Identity

Provider при регистрации.

В случае успеха в теле запроса должны быть указаны:

- идентификатор запроса (*id*);
- *extSessionId* – идентификатор сессии, сгенерированный внешним методом. Идентификатор должен совпадать со значением, переданным в исходном запросе в cookie;
- *claims* – перечень утверждений, которыми нужно обогатить сессию пользователя. Перечень может быть пустым;
- *subjectId* – идентификатор пользователя (только для первого фактора; при вызове внешнего метода в качестве второго фактора нельзя передавать идентификатор пользователя);
- *loginContextId* – идентификатор контекста аутентификации, соответствующий исходному запросу.

Пример запроса:

```
POST /blitz/login/methods/outside/save?methodName=outside_test HTTP/1.1
Content-Type: application/json

{
    "id": "426b5139-e4f7-41e6-a206-9503de6f34dd",
    "extSessionId": "YTk2OTIwOTEtNDYxMy00MWFhLTkxZDItOWE3MWEzZmMyZTA3",
    "claims": {},
    "loginContextId": "3ca4d1f0-654a-4665-be98-d105ab6ec35d",
    "subjectId": "2db787c7-6e37-4018-abe9-2bea1011c047"
}
```

В случае ошибки в теле запроса должны быть указаны:

- *id* – идентификатор запроса;
- *extSessionId* – идентификатор сессии, сгенерированный внешним методом. Идентификатор должен совпадать со значением, переданным в исходном запросе в cookie;
- *error* – код ошибки;
- *msg* – текстовое описание ошибки (опционально).

```
POST /blitz/login/methods/outside/save?methodName=outside_test HTTP/1.1
Content-Type: application/json

{
    "id": "426b5139-e4f7-41e6-a206-9503de6f34dd",
    "extSessionId": "YTk2OTIwOTEtNDYxMy00MWFhLTkxZDItOWE3MWEzZmMyZTA3",
    "error": "not_found",
    "msg": "User not found"
}
```

В случае сохранения результатов аутентификации (как успешной, так и неуспешной) Blitz Identity Provider возвращает ответ *HTTP 200 OK*.

b) браузерная часть «поставщика» должна обеспечить перенаправление пользователя обратно в Blitz Identity Provider. Для этого необходимо перенаправить браузер по адресу:

```
https://login.company.com/blitz/login/methods/outside/callback?methodName=outside_{name}
```

В данном запросе *name* – это имя внешнего метода, присвоенное ему в Blitz Identity Provider при регистрации.

Б.3. Сервис проверки применимости метода аутентификации

Представляет собой URL для приема HTTP-запросов от Blitz Identity Provider. До запроса на аутентификацию Blitz Identity Provider будет делать запрос методом POST по данному адресу, передав в теле в формате JSON те же данные, что и в запросе на аутентификацию.

В качестве ответа внешний метод должен вернуть JSON со следующими атрибутами:

- идентификатор запроса (*id*);
- результат проверки применимости (*result*), принимающий значение либо *true* (метод применим) или *false* (метод не применим);
- идентификатор контекста аутентификации (*loginContextId*), соответствующий запросу.

Если сервис вернет *false* в качестве результата проверки применимости, то далее Blitz Identity Provider не будет выполнять запрос на аутентификацию для данного пользователя.

Приложение B. API аутентификации пользователя

Стандартно при необходимости провести идентификацию и аутентификацию пользователя веб-сайт или мобильное приложение взаимодействует с Blitz Identity Provider по любому из доступных протоколов (см. п. 1). При этом непосредственно аутентификацией приложение не занимается. Приложение перенаправляет пользователя в Blitz Identity Provider на страницу входа. Далее Blitz Identity Provider самостоятельно предлагает пользователю различные методы аутентификации, осуществляет взаимодействие с пользователем в процессе входа.

В некоторых случаях может быть желательно предоставить пользователю возможность пройти идентификацию и аутентификацию без перенаправления на страницу входа Blitz Identity Provider. Такие возможности ограничены (не все методы входа и подтверждения входа доступны без перенаправления), требуют большого объема доработок на стороне приложения (так как в приложении необходимо поддержать обработку различных сценариев, связанных с аутентификацией).

Blitz Identity Provider предоставляет HTTP API, позволяющее встроить в веб-страницу приложения идентификацию и аутентификацию пользователей без перенаправления пользователя на отдельную страницу входа. Данное HTTP API создано для веб-приложений. При использовании API обеспечивается Web Single Sign-On, а именно при последующем входе в той же веб-сессии пользователя в другое подключенное к Blitz Identity Provider приложение, у него не будет повторно запрашиваться вход.

B.1. Настройки для использования API

Приложение должно быть зарегистрировано в Blitz Identity Provider. Приложению в Blitz Identity Provider должны быть присвоены *client_id* и *client_secret*, и в Blitz Identity Provider должны быть зарегистрированы URL возврата приложения.

Взаимодействие страницы приложения и Blitz Identity Provider основано на выполнении серии AJAX-взаимодействий. Для возможности такого взаимодействия на веб-сервере приложения и на веб-сервере Blitz Identity Provider должны быть сделаны следующие настройки CORS (Cross-origin resource sharing):

На сервере Blitz Identity Provider для обработчика */blitz/oauth/ae* нужно настроить CORS-разрешение, добавив следующие HTTP Headers (нужно указать *origin* для ПРОД-сайта и необходимые *origin* для нужных тестовых сред):

```
"Access-Control-Allow-Origin" -> "https://{app-domain}",
"Access-Control-Allow-Credentials" -> "true"
```

В этом заголовке *{app-domain}* – это домен приложения.

На сервере портала для callback-обработчика (см. следующий раздел) ответа от Blitz Identity Provider нужно настроить следующее CORS-разрешение (разрешение на *null*, так как после редиректа браузер сбрасывает *origin*):

```
"Access-Control-Allow-Origin" -> null,
"Access-Control-Allow-Credentials" -> "true"
```

B.2. Схема взаимодействия

HTTP API аутентификации позволяет:

- Проверить наличие SSO-сессии. В случае отсутствия SSO-сессии получить список доступных пользователю методов аутентификации.
- Провести идентификацию и аутентификацию с использованием логина и пароля.
- Провести идентификацию и аутентификацию с использованием логина (телефона) и кода подтверждения, отправляемого по SMS.
- Провести идентификацию и аутентификацию по QR-коду;
- Провести подтверждение входа с использованием кода подтверждения, отправляемого по SMS.

На рисунке 10 приведена схема взаимодействия при входе по логину и паролю с последующим подтверждением входа с использованием кода подтверждения, отправляемого по SMS.

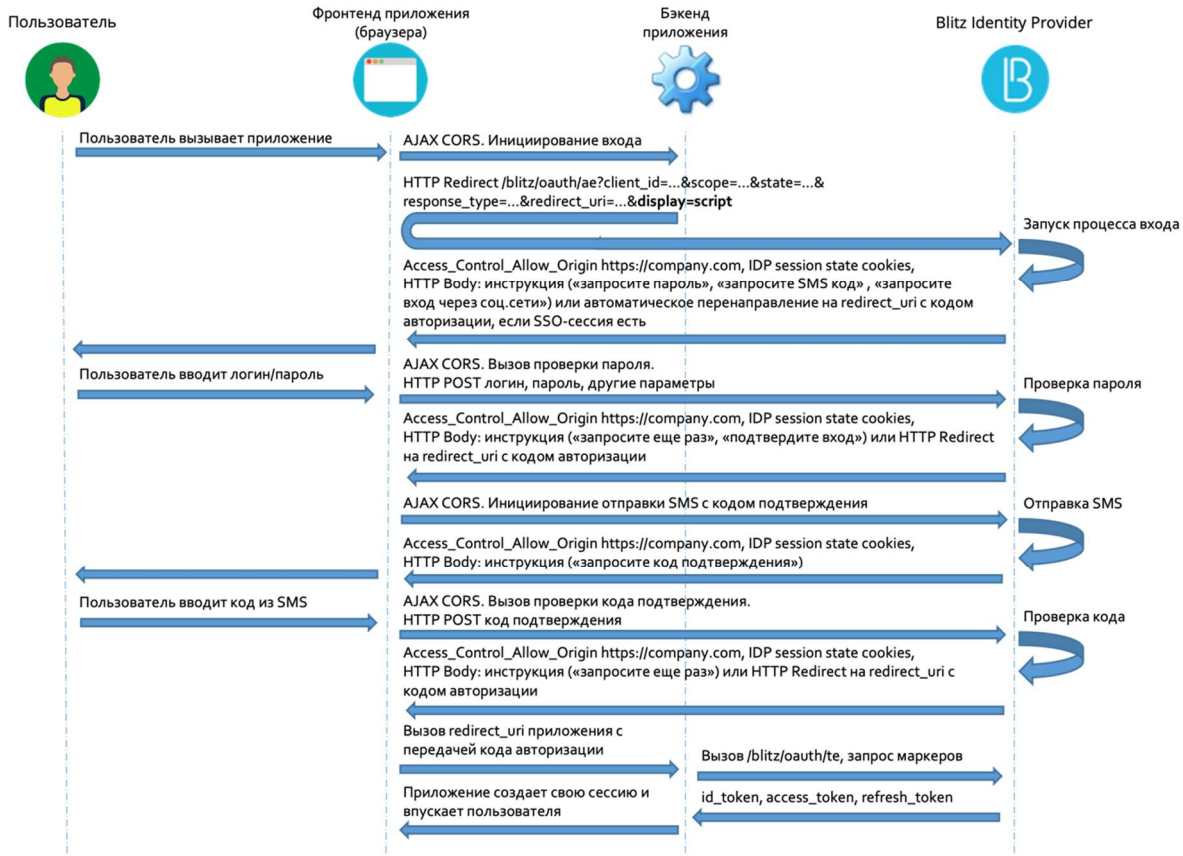

Рисунок 10 — Схема взаимодействия при входе по логину и паролю

На рисунке 11 приведена схема взаимодействия при входе по телефону и коду подтверждения, отправляемому по SMS.

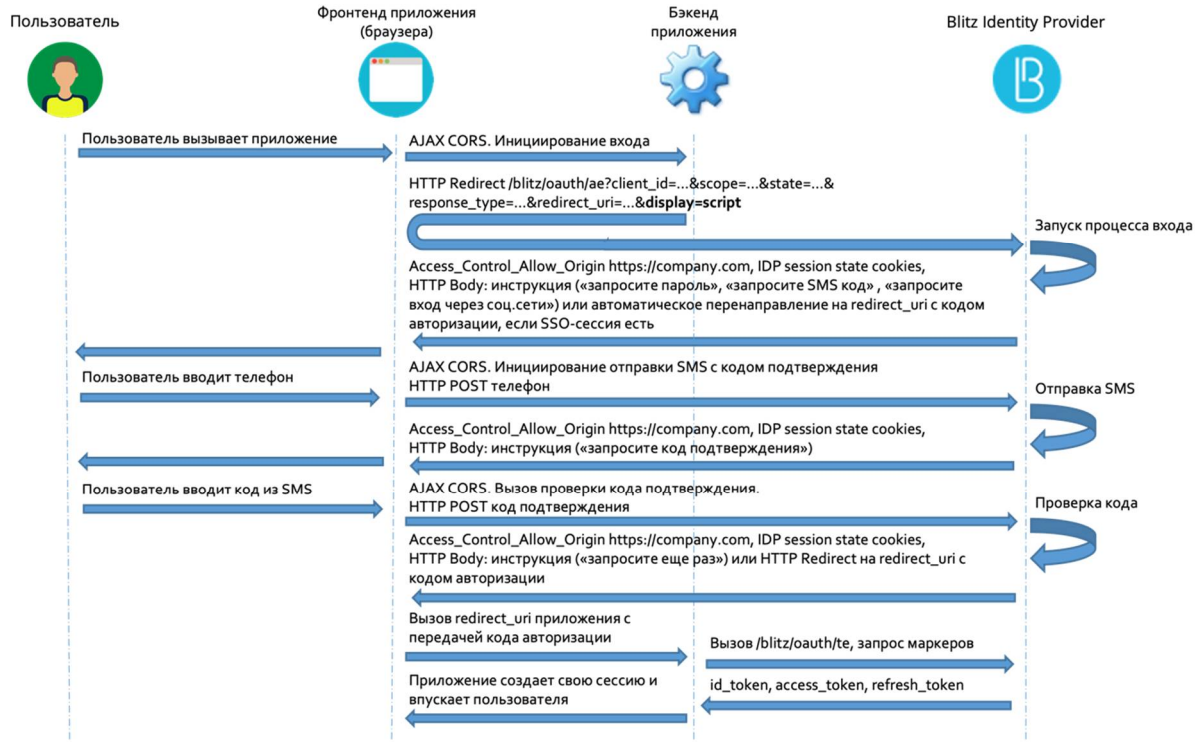

Рисунок 11 — Схема взаимодействия при входе по телефону и коду подтверждения

Веб-приложение взаимодействует с Blitz Identity Provider, выполняя серию из AJAX-запросов.

> **Примечание** — Запросы должны делаться обязательно с сохранением и передачей cookie – необходимо использовать *withCredentials: true*

В последующих разделах приводятся описания вызываемых запросов, возможных ответов и рекомендаций по их обработке. Примеры запросов и ответов приводятся в виде вызовов cURL.

B.3. Запуск процесса входа

Чтобы запустить процесс входа, приложение должно направить в AJAX к Blitz Identity Provider запрос HTTP GET (обязательно с *withCredentials: true)* на обычный обработчик Authorization Endpoint (*/blitz/oauth/ae*, см. п. 3.3.2), добавив к запросу специальный параметр *display=script.*

Пример запроса:

```
curl -v -b cookies.txt -c cookies.txt \
--request GET
'https://login.company.com/blitz/oauth/ae?response_type=code&client_id=ais&scope=openid&state=…&display=script&redirect_uri=https%3A%2F%2Fapp.company.com%2Fre'
```

Если SSO-сессия уже существует, то Blitz Identity Provider автоматически перенаправит пользователя на адрес обработчика *redirect_uri*, добавив к запросу код

авторизации и параметр *state*. Используя полученный код авторизации приложение продолжит стандартное взаимодействие по OpenID Connect для получения маркеров безопасности и данных учетной записи.

Пример ответа с перенаправлением, если сессия SSO-сессия уже существует:

```
…
< HTTP/2 302
…
< location: https://…?code=…&state=…
…
```

Пример ответа, если требуется аутентификация:

```
{
    "inquire":"choose_one",
    "items":[
        {
            "inquire":"login_with_password"
        },
        {
            "inquire":"request_auth_with_fed_point",
            "fp":"esia:esia_1"
        },
        …
        {
            "inquire":"request_auth_with_fed_point",
            "fp":"yandex:yandex_1"
        },
        {
            "inquire":"login_to_send_sms"
        },
        {
            "inquire":"show_qr_code",
            "link":"https://…?code=dde087f0-8f4a-478e-886b-5354b0283362",
            "expires":1660905165,
            "logo":"https:/…"
        }
    ]
}
```

Если требуется аутентификация, то Blitz Identity Provider возвращает приложению одну из возможных инструкций:

- *login_with_password* – войти по логину и паролю;
- *request_auth_with_fed_point* – войти с помощью внешнего поставщика идентификации (социальной сети);
- *login_to_send_sms* – войти с помощью логина и кода подтверждения, отправленного по SMS;
- *show_qr_code* – отобразить QR-код, позволяющий осуществить вход.

Если какие-то из методов аутентификации не сконфигурированы в Blitz Identity Provider или являются недоступными для входа в запрашивающее приложение (например, в результате настроек «процедуры входа» для соответствующего приложения), то и инструкции по ним будут отсутствовать в ответе сервиса.

В зависимости от включенных в Blitz Identity Provider режимов защиты инструкция *login_with_password* может содержать дополнительные параметры:

– Если в Blitz Identity Provider настроен режим необходимости использования CAPTCHA при входе, то в инструкции будет параметр *captchaId*, который необходимо использовать приложению для теста CAPTCHA:

```
{
    "inquire": "choose_one",
    "items": [
        {
            "inquire": "login_with_password",
            "captchaId": "9cf48a75-6be1-4008-b34e-8906220c472f"
        },
        …
    ]
}
```

– Если в Blitz Identity Provider настроен режим защиты от подбора пароля, требующий решения от приложения длительной вычислительной задачи (Proof of Work), то в инструкции будет параметр *proofOfWork*:

```
{
    "inquire": "choose_one",
    "items": [
        {
            "inquire": "login_with_password",
            "captchaId": "9cf48a75-6be1-4008-b34e-8906220c472f",
            "proofOfWork": "1:15:220313184752:abe…539::Ekf…w==:"
        }
    ]
}
```

– В случае получения параметра *proofOfWork* рекомендуется асинхронно сразу запустить алгоритм нахождения решения, не дожидаясь, пока пользователь выберет режим входа по логину и паролю и введет данные. Это позволит скрыть от пользователя время задержки на решение задачи (может составлять несколько секунд в зависимости от сложности задачи). В настоящий момент используется алгоритм Hashcash[53]. Необходимо дополнить параметр proofOfWork таким значением[54], чтобы вычисленный от него по алгоритму SHA-1 хэш содержал в начале столько нулевых бит, сколько задано условием задачи (число после первого символа : в параметре proofOfWork).

В зависимости от выбранного способа аутентификации приложение вызывает в Blitz Identity Provider вход одним из следующих способов:

– Вход по логину и паролю – описан далее в п. B.4.
– Вход по телефону и коду подтверждения в SMS – описан далее в п. B.5.
– Вход по QR-коду – описан далее в п. B.6.
– Вход через внешний поставщик идентификации – такой способ входа возможен только через браузер с перенаправлением пользователя на страницу входа внешнего поставщика идентификации. Нужно повторить вызов *Authorization Endpoint*

[53] См.: http://www.hashcash.org
[54] Например, решением для 1:15:yyyy03Su212003:BlitzIdp::McMybZIhxKXu57jd:0 будет строка 1:15:yyyy03Su212003:BlitzIdp::McMybZIhxKXu57jd:**3/g**

(см. п. п. 3.3.2), использовать в вызове необходимое значение параметра *bip_action_hint*, соответствующее выбранному пользователем внешнему поставщику входа (например, *bip_action_hint=externalIdps:esia:esia_1*).

Пример запроса:

```
https://login.company.com/blitz/oauth/ae?response_type=code&client_id=portal.ru&scope=openid+prof
ile&redirect_uri=https://apitest.company.com/success&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f&
bip_action_hint=used_externalIdps:esia:esia_1
```

Завершение процесса входа в этом случае будет происходить стандартным образом в соответствии с OpenID Connect – Blitz Identity Provider вернет код авторизации на *redirect_uri* обработчик приложения.

B.4. Вход по логину и паролю

Если в Blitz Identity Provider настроено использование CAPTCHA, то до вызова проверки логина и пароля приложение должно выполнить вызовы по получению и проверке CAPTCHA. Запросы на проверку должны формироваться через специализированные proxy-сервисы Blitz Identity Provider, а не напрямую к сервисам CAPTCHA.

При использовании reCAPTCHA v3 необходимо выполнить инициализацию reCAPTCHA v3 согласно документации[55]:

– Загрузить на странице приложения скрипт, используя такой же reCAPTCHA v3 *sitekey* как зарегистрирован в Blitz Identity Provider:

```
<script src="https://www.google.com/recaptcha/api.js?render=reCAPTCHA_site_key"></script>
```

– Вызвать grecaptcha.execute на нажатие кнопки входа:

```
<script>
    function onClick(e) {
        e.preventDefault();
        grecaptcha.ready(function() {
            grecaptcha.execute('reCAPTCHA_site_key', {action: 'submit'}).then(function(token) {
                // Add your logic to submit to your backend server here.
            });
        });
    }
</script>
```

Сразу после вызова со страницы входа сервисов reCAPTCHA необходимо вызвать с сервера приложений операцию проверки (verify). Вызов должен быть произведен не напрямую на сервера Google, а через специальный proxy-сервис в Blitz Identity Provider.

Пример запроса на проверки (операция *verify*):

```
POST /blitz/login/captcha/verify
Content-Type: 'text/json'
{
    "ctx": {
         // captchaId, полученный от Blitz Identity Provider
        "id": "9cf48a75-6be1-4008-b34e-8906220c472f",
        "method": "password"
    },
    "params": {
        // token для проверки капчи, полученный при регистрации в Google
        "response": "03…sA"
    }
}
```

[55] См.: https://developers.google.com/recaptcha/docs/v3#programmatically_invoke_the_challenge

Ответ *HTTP 200 OK*:

```
{
    "action": "submit",
    "challenge_ts": "2021-03-16T11:18:41Z",
    "success": true,
    "hostname": "company.com",
    "score": 0.9
}
```

Также если в Blitz Identity Provider включена защита Proof of Work, то нужно вычислить значение параметра *proofOfWork* как описано в п. B.3.

Для проверки логина и пароля приложение должно направить в AJAX к Blitz Identity Provider запрос HTTP POST (обязательно с *withCredentials: true*) на URL *https://login.company.com/blitz/login/methods/headless/password* с *Content-Type x-www-form-urlencoded* и Body, содержащим параметры *login* и *password*, а также вычисленный *proofOfWork* (если этот параметр был получен от Blitz Identity Provider при запуске процесса входа).

Пример запроса:

```
curl -v -b cookies.txt -c cookies.txt \
--request POST 'https://login.company.com/blitz/login/methods/headless/password' \
--header "Content-Type: application/x-www-form-urlencoded" \
--data 'login=логин&password=пароль&proofOfWork=решение'
```

Blitz Identity Provider при получении запроса выполняет необходимые проверки безопасности (пройдена ли CAPTCHA, решен ли ProofOfWork, не заблокирована ли учетная запись). Если проверки безопасности пройдены, то Blitz Identity Provider проверяет переданные логин и пароль.

Если проверки логина и пароля успешные и если пройденной аутентификации достаточно, то Blitz Identity Provider автоматически перенаправит пользователя на адрес обработчика *redirect_uri*, добавив к запросу код авторизации и параметр *state*. Используя полученный код авторизации приложение продолжит стандартное взаимодействие по OpenID Connect для получения маркеров безопасности и данных учетной записи.

Пример ответа с перенаправлением, если сессия SSO-сессия уже существует:

```
…
< HTTP/2 302
…
< location: https://…?code=…&state=…
…
```

Если какие-либо проверки завершились ошибкой или если необходимы дальнейшие действия от пользователя, то Blitz Identity Provider возвращает одну из инструкций.

Пример ответа в случае ошибки проверки логина и пароля:

```
{
    "inquire": "login_with_password",
    "errors": [
        {
            "code": "invalid_credentials",
            "params": {}
        }
    ]
}
```

При получении такого ответа приложение может отобразить текст ошибки и предложить пользователю ввести еще логин и пароль, после чего можно повторить проверку логина и пароля.

Если пользователь ввел пароль, который ранее был в учетной записи, или если учетная запись заблокирована, то ошибка будет иметь вид:

```
{
    "inquire": "login_with_password",
    "captchaId": "9cf48a75-6be1-4008-b34e-8906220c472f",
    "proofOfWork": "1:15:220313184752:abe…539::Ekf…w==:",
    "errors": [
        {
            "code": "invalid_credentials",
            "params": {
                "_cause": "used_old_password"
            }
        }
    ]
}
```

Пример получения ошибки, что не прошла проверка CAPTCHA:

```
{
    "inquire": "login_with_password",
    "captchaId": "9cf48a75-6be1-4008-b34e-8906220c472f",
    "errors": [
        {
            "code": "invalid_captcha",
            "params": {}
        }
    ]
}
```

Пример ошибки, что не прошла проверка решения Proof of Work:

```
{
    "inquire": "handle_error",
    "errors": [
        {
            "code": "doesNotMatch",
            "params": {}
        }
    ]
}
```

Если в Blitz Identity Provider включена специальная защита на задержку проверки логина и пароля, то при проверке логина и пароля можно получить от Blitz Identity Provider следующую инструкцию, что нужен повторный вызов проверки пароля спустя определенное число секунд:

```
{
    "inquire": "delayed_login_with_password",
    "delayedFor": 5
}
```

Повторный вызов должен быть сделан, когда пройдет требуемое время. В повторный вызов необходимо передать параметр *isDelayed=true*.

Пример повторного вызова проверки пароля в ответ на инструкцию *delayed_login_with_password:*

```
curl -v -b cookies.txt -c cookies.txt \
--request POST 'https://login.company.com/blitz/login/methods/headless/password' \
--header "Content-Type: application/x-www-form-urlencoded" \
--data 'login=логин&password=пароль&proofOfWork=решение&isDelayed=true'
```

Если в Blitz Identity Provider включена специальная защита от перебора пароля, то Blitz Identity Provider при проверке пароля по данной учетной записи может запросить

дополнительно проверку CAPTCHA. Различаются две возможные ситуации:

- Пользователь передал неправильный пароль, после чего включилась защита, и CAPTCHA нужна для очередной попытки аутентификации.
- Защита от подбора пароля для учетной записи включалась ранее. Текущий переданный пароль не проверялся, так как не проводился тест CAPTCHA.

В первом случае нужно сообщить пользователю, что логин и пароль неправильный, и для новой попытки дополнительно к вводу пароля запросить пройти тест CAPTCHA.

Во втором случае нужно попросить пользователя пройти тест CAPTCHA, после чего направить на проверку ранее введенные логин и пароль.

Пример ответа для первого случая, что пароль неправильный и нужен тест CAPTCHA:

```
{
    "inquire":"login_with_password",
    "captchaId":"1c9e4047-c8c4-47ad-a447-cc1809bd3e6c",
    "errors":[
        {
            "code":"invalid_credentials",
            "params":{}
        }
    ]
}
```

Пример ответа для второго случая, что пароль не проверялся и нужен тест CAPTCHA:

```
{
    "inquire":"login_with_password",
    "captchaId":"2f818f5d-3a89-428d-b424-cde38c19051e",
    "errors":[
        {
            "code":"bypass_captcha",
            "params":{}
        }
    ]
}
```

Пример ошибки, если учетная временно заблокирована:

```
{
    "inquire":"login_with_password",
    "errors": [
        {
            "code":"pswd_method_temp_locked",
            "params":{"0":"2"}
        }
    ]
}
```

Если пароль учетной записи не соответствует парольной политике, то может возникнуть необходимость сменить пароль при входе. В этом случае Blitz Identity Provider вернет инструкцию, что необходимо перенаправить пользователя на страницу с указанным адресом:

```
{
    "inquire":"go_to_web",
    "redirect_uri":
        "https://…/blitz/login/methods/password/change?f=false&c=password_policy_violated"
}
```

Если логин и пароль успешны, но дополнительно требуется подтвердить вход, то вернется инструкция с возможными способами подтверждения:

```
{
    "inquire":"choose_one",
    "items": [
        {
            "inquire":"ask_to_send_sms"
        },
        {
            "inquire":"go_to_web",
            "redirect_uri":"https://login.company.com/blitz/login/methods2/sms"
        }
    ]
}
```

Можно или перенаправить пользователя на веб-страницу, чтобы он продолжил подтверждение входа на веб-странице Blitz Identity Provider, или продолжить использовать HTTP API для подтверждения входа по коду из SMS – см. п. В.7.

Если в процедуре входа, установленной для приложения, настроен вызов дополнительного экрана после входа (например, см. Приложение Г), то Blitz Identity Provider переадресует пользователя на этот экран.

В.5. Вход по телефону и коду подтверждения

Вход по телефону и коду подтверждения состоит из следующих шагов:

- Отправка пользователю кода подтверждения по SMS.
- Проверка введенного пользователем кода подтверждения.

Для отправки пользователю кода подтверждения по SMS приложение должно направить в AJAX к Blitz Identity Provider запрос HTTP POST (обязательно с *withCredentials: true*) на URL *https://login.company.com/blitz/login/methods/headless/sms/bind* с *Content-Type x-www-form-urlencoded* и Body, содержащим *login* пользователя. В качестве *login* рекомендуется передавать номер телефона, введенный пользователем.

Пример запроса:

```
curl -v -b cookies.txt -c cookies.txt \
--request POST 'https://login.company.com/blitz/login/methods/headless/sms/bind' \
--header "Content-Type: application/x-www-form-urlencoded" \
--data 'login=логин'
```

Если учетная запись с переданным логином не найдена, то Blitz Identity Provider вернет ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"no_subject_found",
            "params":{}
        }
    ]
}
```

Если учетная запись найдена, но по ней ранее был зафиксирован перебор кодов подтверждения, то Blitz Identity Provider вернет ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"method_temp_locked",
```

```
            "params":{}
        }
    ]
}
```

Если учетная запись найдена и для нее возможен вход данным способом, то Blitz Identity Provider отправит пользователю SMS с кодом подтверждения и вернет ответ:

```
{
    "inquire":"enter_sms_code",
    "contact":"+79991234567",
    "ttl":300,
    "remain_attempts":3
}
```

В полученном ответе указано, сколько секунд у пользователя остается для отправки кода на проверку (ttl), сколько попыток ввести код у него есть (remain_attempts), на какой номер телефона ему был отправлен код (*contact*).

Для проверки введенного пользователем кода подтверждения приложение должно направить в AJAX к Blitz Identity Provider запрос HTTP POST (обязательно с *withCredentials: true*) на URL *https://login.company.com/blitz/login/methods/headless/sms/bind* с *Content-Type x-www-form-urlencoded* и Body, содержащим *sms-code* с кодом подтверждения.

Пример запроса:

```
curl -v -b cookies.txt -c cookies.txt \
--request POST 'https://login.company.com/blitz/login/methods/headless/sms/bind' \
--header "Content-Type: application/x-www-form-urlencoded" \
--data 'sms-code=123456'
```

Если код неправильный, то Blitz Identity Provider вернет ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"invalid_otp",
            "params":{}
        }
    ],
    "contact":"+79991234567",
    "remain_attempts":2,
    "ttl":276
}
```

Если количество попыток проверки кода закончилось, то Blitz Identity Provider вернет ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"no_attempts",
            "params":{}
        }
    ]
}
```

Если срок действия кода истек, то Blitz Identity Provider вернет ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"expired",
            "params":{}
        }
    ]
}
```

В случае этой ошибки можно запросить отправку нового кода подтверждения. Для этого приложение должно вызвать Blitz Identity Provider следующим образом:

```
curl -v -b cookies.txt -c cookies.txt \
--request POST 'https://login.company.com/blitz/login/methods/headless/sms/bind' \
--header "Content-Type: application/x-www-form-urlencoded" \
--data 'sms-send=sms'
```

Если запросить переотправку кода до истечения срока действия предыдущего, то вернется ошибка:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"code_not_expired",
            "params":{}
        }
    ]
}
```

Если общее количество попыток входа по коду подтверждения из SMS превышено, то Blitz Identity Provider осуществляет временное блокирование входа для учетной записи по коду подтверждения. В этом случае при очередной попытке ввода неправильного кода подтверждения Blitz Identity Provider может вернуть ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"method_temp_locked",
            "params":{}
        }
    ]
}
```

Если введенный код подтверждения правильный, и этого достаточно для завершения входа, то Blitz Identity Provider автоматически перенаправит пользователя на адрес обработчика *redirect_uri*, добавив к запросу код авторизации и параметр *state*. Используя полученный код авторизации приложение продолжит стандартное взаимодействие по OpenID Connect для получения маркеров безопасности и данных учетной записи.

Пример ответа в случае успешного входа:

```
…
< HTTP/2 302
…
< location: https://…?code=…&state=…
…
```

Если проверка кода подтверждения успешна, но дополнительно требуется подтвердить вход, то вернется инструкция с возможными способами подтверждения:

```
{
    "inquire":"choose_one",
    "items":[
        {
            "inquire":"go_to_web",
            "redirect_uri":"https://login.company.com/blitz/login/methods2/email"
        }
    ]
}
```

B.6. Вход по QR-коду

Вход по QR-коду состоит из следующих шагов:

– Отображение пользователю QR-кода на компьютере, на котором выполняется вход;

- Периодическая проверка, выполнил ли пользователь сканирование QR-кода мобильным приложением;
- Периодическая проверка, подтвердил или отклонил пользователь в мобильном приложении запрос на вход по QR-коду;
- Обновление устаревшего QR-кода.

Приложение должно отобразить пользователю QR-код, закодировав в него строку, полученную от Blitz Identity Provider. Ниже показан фрагмент инструкции для входа по QR-коду (см. описание ранее в п. B.3).

```
{
    "inquire":"choose_one",
    "items":[
        …
        {
            "inquire":"show_qr_code",
            "link":"https://…?code=dde087f0-8f4a-478e-886b-5354b0283362",
            "expires":1660905165,
            "logo":"https:/…"
        }
    ]
}
```

Пояснения по полученным от Blitz Identity Provider параметрам:

- *inquire* – инструкция с доступным вариантом входа, в случае входа по QR-коду имеет значение *show_qr_code*;
- *link* – ссылка, которая должна быть закодирована в QR-коде, отображаемом пользователю;
- *expires* – время (в Unix Epoch), до которого действителен QR-код. По истечении срока действия рекомендуется отобразить пользователю, что QR-код просрочен;
- *logo* – если в Blitz Identity Provider настроено отображение маленького логотипа в центре поверх QR-кода, то в указанной настройке вернется URL-адрес логотипа.

Когда приложение отобразило пользователю QR-код, необходимо дождаться, чтобы пользователь прочитал QR-код специальным мобильным приложением. Интеграция мобильного приложения для встраивания функции входа по QR-коду описано в п. 3.4.9.

Веб-приложение может периодически выполнять проверку, был ли считан мобильным приложением QR-код. Для этого необходимо выполнить в AJAX к Blitz Identity Provider запрос HTTP GET (обязательно с *withCredentials: true*) на URL *https://login.company.com/blitz/login/methods/headless/qrCode/pull*.

Пример запроса:

```
curl -v -b cookies.txt -c cookies.txt \
--request GET 'https://login.company.com/blitz/login/methods/headless/qrCode/pull'
```

Если QR-код еще не был считан, то вернется ответ:

```
{
    "command":"showQRCode"
}
```

Если QR-код считан, то вернется ответ:

```
{
    "command":"askForConfirm"
}
```

В этом случае можно обновить пользователю веб-страницу и написать на ней, что ожидается подтверждение входа в мобильном приложении.

Если QR-код просрочен, то вернется ответ:

```
{
    "command":"needRefresh",
    "cause":"qr_code_expired"
}
```

Если пользователь отклонил в мобильном приложении запрос входа по QR-коду, то вернется ответ:

```
{
    "command":"needRefresh",
    "cause":"refused_login"
}
```

В случае если QR-код просрочен или пользователь отклонил вход по QR-коду, то можно предложить пользователю получить новый QR-код. Для этого выполнить в AJAX к Blitz Identity Provider запрос HTTP POST (обязательно с *withCredentials: true*) на URL *https://login.company.com/blitz/login/methods/headless/qrCode/refresh*.

Пример запроса:

```
curl -v -b cookies.txt -c cookies.txt \
--request POST 'https://login.company.com/blitz/login/methods/headless/qrCode/refresh'
```

Пример ответа:

```
{
    "link":"https://…?code=4ddf1667-d57f-4f86-b8f2-3ee53b367dfe",
    "expires":1660922807,
    "logo":"https:/…"
}
```

Если пользователь подтвердил в мобильном приложении запрос входа по QR-коду, то сервис *https://login.company.com/blitz/login/methods/headless/qrCode/pull* вернется ответ:

```
{
    "command":"needComplete"
}
```

В ответ на этот запрос для завершения входа необходимо выполнить в AJAX к Blitz Identity Provider запрос HTTP POST (обязательно с *withCredentials: true*) на URL *https://login.company.com/blitz/login/methods/headless/qrCode/complete*.

Пример запроса:

```
curl -v -b cookies.txt -c cookies.txt \
--request POST 'https://login.company.com/blitz/login/methods/headless/qrCode/complete'
```

Если пройденной аутентификации достаточно для завершения входа, то Blitz Identity Provider автоматически перенаправит пользователя на адрес обработчика *redirect_uri*, добавив к запросу код авторизации и параметр *state*. Используя полученный код авторизации приложение продолжит стандартное взаимодействие по OpenID Connect для получения маркеров безопасности и данных учетной записи.

Пример ответа в случае успешного входа:

```
…
< HTTP/2 302
…
< location: https://…?code=…&state=…
…
```

Если требуется пройти дополнительно подтверждение входа, то вернется инструкция с возможными способами подтверждения:

```
{
    "inquire":"choose_one",
    "items":[
        {
            "inquire":"go_to_web",
            "redirect_uri":"https://login.company.com/blitz/login/methods2/email"
        }
    ]
}
```

B.7. Подтверждение входа по коду подтверждения

Подтверждение входа с помощью кода подтверждения по SMS состоит из следующих шагов:

- Отправка пользователю кода подтверждения по SMS.
- Проверка введенного пользователем кода подтверждения.

Для отправки пользователю кода подтверждения по SMS приложение должно направить в AJAX к Blitz Identity Provider запрос HTTP POST (обязательно с *withCredentials: true*) на URL *https://login.company.com/blitz/login/methods/headless/sms/bind* с *Content-Type x-www-form-urlencoded* без Body:

Пример запроса:

```
curl -v -b cookies.txt -c cookies.txt \
--request POST 'https://login.company.com/blitz/login/methods/headless/sms/bind' \
--header "Content-Type: application/x-www-form-urlencoded"
```

Blitz Identity Provider отправит пользователю SMS с кодом подтверждения и вернет ответ:

```
{
    "inquire":"enter_sms_code",
    "contact":"+79991234567",
    "ttl":300,
    "remain_attempts":3
}
```

В полученном ответе указано, сколько секунд у пользователя остается для отправки кода на проверку (ttl), сколько попыток ввести код у него есть (remain_attempts), на какой номер телефона ему был отправлен код (*contact*).

Для проверки введенного пользователем кода подтверждения приложение должно направить в AJAX к Blitz Identity Provider запрос HTTP POST (обязательно с *withCredentials: true*) на URL *https://login.company.com/blitz/login/methods/headless/sms/bind* с *Content-Type x-www-form-urlencoded* и Body, содержащим *sms-code* с кодом подтверждения.

Пример запроса:

```
curl -v -b cookies.txt -c cookies.txt \
```

```
--request POST 'https://login.company.com/blitz/login/methods/headless/sms/bind' \
--header "Content-Type: application/x-www-form-urlencoded" \
--data 'sms-code=123456'
```

Если код неправильный, то Blitz Identity Provider вернет ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"invalid_otp",
            "params":{}
        }
    ],
    "contact":"+79991234567",
    "remain_attempts":2,
    "ttl":276
}
```

Если количество попыток проверки кода закончилось, то Blitz Identity Provider вернет ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"no_attempts",
            "params":{}
        }
    ]
}
```

Если срок действия кода истек, то Blitz Identity Provider вернет ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"expired",
            "params":{}
        }
    ]
}
```

В случае этой ошибки можно запросить отправку нового кода подтверждения. Для этого приложение должно вызвать Blitz Identity Provider следующим образом:

```
curl -v -b cookies.txt -c cookies.txt \
--request POST 'https://login.company.com/blitz/login/methods/headless/sms/bind' \
--header "Content-Type: application/x-www-form-urlencoded" \
--data 'sms-send=sms'
```

Если запросить переотправку кода до истечения срока действия предыдущего, то вернется ошибка:

```
{
    "inquire":"handle_error",
    "errors":[
        {
            "code":"code_not_expired",
            "params":{}
        }
    ]
}
```

Если общее количество попыток подтверждения входа по коду подтверждения из SMS превышено, то Blitz Identity Provider осуществляет временное блокирование подтверждения входа для учетной записи по коду подтверждения. В этом случае при очередной попытке ввода неправильного кода подтверждения Blitz Identity Provider может вернуть ошибку:

```
{
    "inquire":"handle_error",
    "errors":[
        {
```

```
            "code":"method_temp_locked",
            "params":{}
        }
    ]
}
```

Если введенный код подтверждения правильный, и этого достаточно для завершения входа, то Blitz Identity Provider автоматически перенаправит пользователя на адрес обработчика *redirect_uri*, добавив к запросу код авторизации и параметр *state*. Используя полученный код авторизации приложение продолжит стандартное взаимодействие по OpenID Connect для получения маркеров безопасности и данных учетной записи.

Пример ответа в случае успешного входа:

```
…
< HTTP/2 302
…
< location: https://…?code=…&state=…
…
```

Приложение Г. Вызов вспомогательного приложения в момент входа

В момент входа Blitz Identity Provider имеет возможность вызвать вспомогательное приложение. Это приложение может выполнить дополнительные операции (например, показать пользователю информационное сообщение или запросить актуализацию сведений), после чего вернуть пользователя в Blitz Identity Provider для последующего входа в целевое приложение.

Вспомогательное приложение должно выполнять следующие действия:

- Обработать запрос на открытие вспомогательного приложения.
- Вернуть пользователя в Blitz Identity Provider после окончания обработки.

Г.1. Прием запроса об открытии вспомогательного приложения

Переход во вспомогательное приложение происходит посредством перенаправления пользователя на предоставленную вспомогательным приложением ссылке. Ссылка в качестве параметра будет содержать код авторизации (*code*). Пример ссылки для инициирования запроса:

```
https://<app_hostname>/?lang=ru&theme=default&code=0Tj…qw
```

При необходимости получить данные пользователя приложение должно обменять код авторизации на маркер доступа согласно спецификации OAuth 2.0. Вспомогательное приложение должно быть предварительно зарегистрировано в Blitz Identity Provider с учетом следующих особенностей:

- должен быть указан предопределенный URL возврата, именно он далее должен быть использован для получения токена;
- должны быть настроены разрешения по умолчания (scope), именно они определяют объем данных, получаемых вспомогательным приложением.

Пример запроса:

```
curl -k -d
"grant_type=authorization_code&redirect_uri=https%3A%2F%2Fapp.company.com%2F&client_id=app&client
_secret=EW…l0&code=0Tj…qw" -X POST https://login.company.com/blitz/oauth/te
```

В ответ будет возвращен маркер доступа, который должен быть использован для получения идентификатора сессии, а также при необходимости данных пользователя.

Пример полученного маркера доступа:

```
{
  "access_token": "ey…J9.ey…n0.Wa…Pw",
  "token_type": "Bearer",
  "expires_in": 3600,
  "scope": "profile"
}
```

Г.2. Возврат пользователя в Blitz Identity Provider

Выполнив необходимые действия (например, показав пользователю информационное сообщение), вспомогательное приложение должно вернуть пользователя в Blitz Identity Provider. Для этого необходимо декодировать полученный маркер доступа, полученный в формате JWT, и извлечь из него утверждение с сессией пользователя (*sessionId*).

Пример тела декодированного *access_token*:

```
{
    "scope": "blitz_api_user blitz_api_user_chg blitz_api_usec_chg",
    "jti": "kfP…jA",
    "client_id": "app",
    "exp": 1631026605,
    "sessionId": "ce9f3109-ac79-46b4-b277-099ff1aa1ff0",
    "iat": 1631023005,
    "sub": "8b970179-e141-43b9-b9d5-25997be99261",
    "aud": [
        "app"
    ],
    "crid": "u9th2LzMXZdwb3rRmI3Paw",
    "iss": "https://login.company.com"
}
```

После этого вспомогательное приложение должно сделать POST-запрос на URL обработчика завершения аутентификации Blitz Identity Provider, имеющий вид */login/pipe/save/<sessionId>*. В теле запроса может быть указан набор утверждений (claims), которые следует добавить в сессию пользователя, либо информацию об ошибке (error).

Пример запроса:

```
curl -v --location --request POST 'https://login.company.com/blitz/login/pipe/save/ce9f3109-ac79-
46b4-b277-099ff1aa1ff0' \
--header 'Content-Type: application/json' \
--header 'Authorization: Basic Z2…ww' \
--data-raw '{"claims":{"org_id":"12345678"}}'
```

В случае успеха Blitz Identity Provider вернет *HTTP 204 No Content*. Получив его, вспомогательное приложение должно вернуть браузер пользователя по адресу */login/pipe/callback*, чтобы пользователь завершил вход в целевое приложение. Пример ссылки для перенаправления:

```
https://login.company.com/blitz/login/pipe/callback
```

Приложение Д. Программные сервисы администрирования Blitz Identity Provider

Администрировать Blitz Identity Provider можно с помощью:

- консоли управления Blitz Console;
- конфигурационных файлов;
- административные REST-сервисов.

Административные REST-сервисы в Blitz Identity Provider в текущей версии позволяют выполнять следующие действия:

- регистрация приложений;
- получение настроек приложений;
- изменение настроек приложений;
- удаление приложений.

Административные REST-сервисы доступны по адресу https://login.company.com/blitz/admin/api/v3/...

Для включения административных сервисов предварительно должны быть сделаны настройки на веб-сервере, используемом Blitz Identity Provider. Не рекомендуется публиковать административные REST-сервисы в сети Интернет.

Пример блока location в настройках веб-сервера nginx для включения доступности административных REST-сервисов:

```
location /blitz/admin/api {
    proxy_intercept_errors off;
    proxy_pass  http://blitz-console/blitz/admin/api;
}
```

Доступ к административным REST-сервисам регулируется с помощью разрешений (scope), приведенных в таблице:

Таблица 11 – Разрешения (scope) для административных REST API

№	Разрешение	Название	Описание
1.	blitz_api_sys_app	Разрешение на чтение настроек приложений	Для использования сервиса GET /blitz/admin/api/v3/app/{appId}
2.	blitz_api_sys_app_chg	Разрешение на внесение изменений в настройки приложений	Для использования сервисов: PUT /blitz/admin/api/v3/app/{appId} POST /blitz/admin/api/v3/app/{appId} DELETE /blitz/admin/api/v3/app/{appId}

Чтобы получить маркер доступа на системное разрешение, приложение должно выполнить запрос методом POST на URL для получения маркера (*https://login.company.com/blitz/oauth/te*). Запрос должен содержать заголовок *Authorization*

со значением *Basic {secret}*, где *secret* – это *client_id:client_secret* (например, *app:topsecret*) в формате Base64.

Пример заголовка:

```
Authorization: Basic YWlzOm…XQ=
```

Тело запроса должно содержать следующие параметры:

- *grant_type* – принимает значение *client_credentials*;
- *scope* – запрашиваемое системное разрешение.

Пример запроса:

```
POST blitz/oauth/te HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Authorization: Basic ZG5ld…lg
Cache-Control: no-cache

grant_type=client_credentials&scope=blitz_api_sys_app+blitz_api_sys_app_chg
```

В ответ приложение получит маркер доступа (*access_token*), время его жизни (*expires_in*) и тип маркера (*token_type*). Возможные ошибки при вызове */oauth/te* соответствуют RFC 6749 и описаны по ссылке[56].

Пример ответа с успешным выполнением запроса:

```
{
    "access_token": "QFiJ9mPgERPuusd36mQvD4mfzYolH_CmuddAJ3YKTOI",
    "expires_in": 3600,
    "scope": "blitz_api_sys_app blitz_api_sys_app_chg",
    "token_type": "Bearer"
}
```

Рекомендуется, чтобы приложение кэшировало полученный маркер доступа для многократного использования на время, немного меньшее, чем параметр *expires_in*, после чего осуществляло получение нового маркера доступа для обновления в кэше.

Если приложение попытается вызвать с просроченным маркером доступа соответствующий ему REST-сервис, то получит ошибку *HTTP 401 Unauthorized*.

Д.1. Получение настроек приложений

Для получения настроек приложения по его идентификатору необходимо методом GET вызвать сервис по адресу *https://login.company.com/blitz/admin/api/v3/app/{appId}*.

Необходимые разрешения: *blitz_api_sys_app*.

В результате выполнения запроса Blitz Identity Provider вернет JSON, содержащий настройки приложения.

Пример запроса:

```
GET /blitz/api/v3/app/test-app HTTP/1.1
Authorization: Bearer cNw…Nz
Content-Type: application/json
```

[56] См.: https://tools.ietf.org/html/rfc6749#section-5.2

Пример ответа:

```
HTTP/2 200
…
content-type: application/json
etag: 96_1658847045000

{
    "name":"…",
    "domain":"…",
    "startPageUrl":"…",
    "oauth": {
        "clientSecret":"…",
        "redirectUriPrefixes":["…"],
        "predefinedRedirectUri":"…",
        "availableScopes":["…","…"],
        "defaultScopes":["…"],
        "enabled":true,
        "autoConsent":true,
        "idToken":{"claims":["…"]},
        "accessTokenTtl":3600,
        "defaultAccessType":"online",
        "refreshTokenTtl":86400,
        "dynReg":{
            "isAllow":true,
            "allowedPlainJsonClaims":["device_type"]
        },
        "pixyMandatory":true,
        "deviceGrant": {
            "userCodeFormat":"[0-9]{3,3}-[0-9]{3,3}-[0-9]{3,3}",
            "userCodeTtl":120,
            "verificationUrl":"…",
            "useCompleteUri":true
        },
        "teAuthMethod":"client_secret_basic",
        "grantTypes":["authorization_code","client_credentials"],
        "responseTypes":["code"],
        "extraClientSecret":"…",
        "accessTokenFormat":"jwt",
        "logout": {
            "logoutAutoConsent":false,
            "logoutUriPrefixes":["…"],
            "predefinedLogoutUri":"…",
            "frontchannelLogoutUri":"…",
            "frontchannelLogoutSessionRequired":true,
            "backchannelLogoutUri":"…"
        }
    },
    "simple": {
        "ssl":true,
        "formSelector":"…",
        "loginSelector":"…",
        "logoutUrl":"…",
        "postLogoutUrl":"…"
    },
    "rest": {
        "Basic":{"pswd":"…"},
        "TLS":[]
    },
    "theme":"default",
    "saml": {
        "spMetadata":"…",
        "spAttributeFilterPolicy": {
            "id":"test-app",
            "attributeRules":[{"attr":"…","isPermitted":true}]
        },
        "saml2SSOProfile": {
            "signAssertions":"always",
            "encryptAssertions":"always",
            "encryptNameIds":"always",
            "includeAttributeStatement":true
        }
    }
}
```

Содержимое ответа может отличаться в зависимости от заданных для приложения настроек и сконфигурированных протоколов подключения. Блоки *saml*, *oauth*, *simple*, *rest* могут отсутствовать, если соответствующие протоколы для приложения не настроены.

В ответе сервиса присутствует заголовок *etag*. Значение из этого заголовка следует использовать в заголовке *If-Match*, если планируется после получения настроек приложения вызывать сервисы регистрации приложения, редактирования настроек приложения или удаления приложения. С помощью *etag* Blitz Identity Provider проверяет, что между последним получением *etag* и вызовом операции изменения настроек с *If-Match* не выполнялись какие-либо еще изменения в конфигурационном файле на сервере в параллельных сеансах (оптимистичное блокирование).

При использовании SAML в настройке *spMetadata* будет находиться закодированный в Base64URL файл метаданных для приложения (Service Provider Metadata).

Имена возвращаемых сервисом настроек соответствуют именам в конфигурационном файле *blitz.conf*.

Если настройки приложения по переданному *appId* не будут найдены, то сервер Blitz Identity Provider вернет ошибку *HTTP 404 Not found*.

Д.2. Регистрация приложения

Для регистрации приложения необходимо выполнить запрос методом PUT по адресу *https://login.company.com/blitz/admin/api/v3/app/{appId}*.

Необходимые разрешения: *blitz_api_sys_app_chg*.

В запрос может быть (опционально) добавлен заголовок *If-Match*, содержащий последнее полученное от сервера значение *etag*.

Тело запроса должно содержать значения настроек регистрируемого приложения.

Пример запроса:

```
PUT /blitz/admin/api/v3/app/test-app2 HTTP/1.1
Authorization: Bearer cNw…Nz
Content-Type: application/json
If-Match: 98_1658857264000

{
    "name":"…",
    "domain":"…",
    "startPageUrl":"…",
    "oauth": {
        "clientSecret":"…",
        "redirectUriPrefixes":["…"],
        "predefinedRedirectUri":"…",
        "availableScopes":["…","…"],
        "defaultScopes":["…"],
        "enabled":true,
        "autoConsent":true,
        "idToken":{"claims":["…"]},
        "accessTokenTtl":3600,
        "defaultAccessType":"online",
        "refreshTokenTtl":86400,
        "dynReg":{
            "isAllow":true,
            "allowedPlainJsonClaims":["device_type"]
        },
        "pixyMandatory":true,
        "deviceGrant": {
            "userCodeFormat":"[0-9]{3,3}-[0-9]{3,3}-[0-9]{3,3}",
```

```
            "userCodeTtl":120,
            "verificationUrl":"…",
            "useCompleteUri":true
        },
        "teAuthMethod":"client_secret_basic",
        "grantTypes":["authorization_code","client_credentials"],
        "responseTypes":["code"],
        "extraClientSecret":"…",
        "accessTokenFormat":"jwt",
        "logout": {
            "logoutAutoConsent":false,
            "logoutUriPrefixes":["…"],
            "predefinedLogoutUri":"…",
            "frontchannelLogoutUri":"…",
            "frontchannelLogoutSessionRequired":true,
            "backchannelLogoutUri":"…"
        }
    },
    "simple": {
        "ssl":true,
        "formSelector":"…",
        "loginSelector":"…",
        "logoutUrl":"…",
        "postLogoutUrl":"…"
    },
    "rest": {
        "Basic":{"pswd":"…"},
        "TLS":[]
    },
    "theme":"default",
    "saml": {
        "spMetadata":"…",
        "spAttributeFilterPolicy": {
            "id":"…",
            "attributeRules":[{"attr":"…","isPermitted":true}]
        },
        "saml2SSOProfile": {
            "signAssertions":"always",
            "encryptAssertions":"always",
            "encryptNameIds":"always",
            "includeAttributeStatement":true
        }
    }
}
```

При регистрации приложения, работающего по SAML, нужно учесть следующие особенности:

- в *spMetadata* нужно передавать содержимое метаданных приложения, закодированное в формате Base64URL.
- в настройку *id* в *spAttributeFilterPolicy* необходимо передать тот же *id*, что передан в URL в качестве *appId*.

Если регистрация успешна, то сервер вернет *HTTP 200*, актуальные данные приложения и актуальное значение *etag*.

Пример ответа:

```
HTTP/2 200
…
content-type: application/json
etag: 99_1658857631000

{
    "id":"test-app2",
    "name":"…",
    …
    "oauth": {
        …
    },
    …
}
```

Если при регистрации приложения будет обнаружено, что данные в конфигурационном файле на сервере были изменены между получением etag и вызовом регистрации, то сервер вернет ответ с кодом *HTTP 412 Precondition Failed* и телом ошибки:

```
{
    "type":"process_error",
    "error":"cas_mismatch",
    "desc":"cas_mismatch"
}
```

Если при регистрации приложения возникла ошибка, то сервер вернет ответ с кодом *HTTP 400 Bad Request* с описанием ошибки.

Пример ответа с ошибкой регистрации:

```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "json.error.mandatory.field",
            "desc": "json.error.expected.array",
            "pos": "oauth.redirectUriPrefixes"
        },
        …
    ]
}
```

Д.3. Изменение настроек приложения

Для изменения настроек приложения необходимо выполнить запрос методом POST по адресу *https://login.company.com/blitz/admin/api/v3/app/{appId}*.

Необходимые разрешения: *blitz_api_sys_app_chg*.

В запрос должен быть добавлен заголовок *If-Match*, содержащий последнее полученное от сервера значение *etag*.

Тело запроса должно содержать значения настроек приложения после редактирования. Должны быть переданы все настройки, а не только изменяемые.

Пример запроса:

```
POST /blitz/admin/api/v3/app/test-app HTTP/1.1
Authorization: Bearer cNw…Nz
Content-Type: application/json
If-Match: 98_1658857264000

{
    "name":"…",
    "domain":"…",
    "startPageUrl":"…",
    "oauth": {
        "clientSecret":"…",
        "redirectUriPrefixes":["…"],
        "predefinedRedirectUri":"…",
        "availableScopes":["…","…"],
        "defaultScopes":["…"],
        "enabled":true,
        "autoConsent":true,
        "idToken":{"claims":["…"]},
        "accessTokenTtl":3600,
        "defaultAccessType":"online",
        "refreshTokenTtl":86400,
        "dynReg":{
            "isAllow":true,
            "allowedPlainJsonClaims":["device_type"]
        },
        "pixyMandatory":true,
        "deviceGrant": {
```

```
            "userCodeFormat":"[0-9]{3,3}-[0-9]{3,3}-[0-9]{3,3}",
            "userCodeTtl":120,
            "verificationUrl":"…",
            "useCompleteUri":true
        },
        "teAuthMethod":"client_secret_basic",
        "grantTypes":["authorization_code","client_credentials"],
        "responseTypes":["code"],
        "extraClientSecret":"…",
        "accessTokenFormat":"jwt",
        "logout": {
            "logoutAutoConsent":false,
            "logoutUriPrefixes":["…"],
            "predefinedLogoutUri":"…",
            "frontchannelLogoutUri":"…",
            "frontchannelLogoutSessionRequired":true,
            "backchannelLogoutUri":"…"
        }
    },
    "simple": {
        "ssl":true,
        "formSelector":"…",
        "loginSelector":"…",
        "logoutUrl":"…",
        "postLogoutUrl":"…"
    },
    "rest": {
        "Basic":{"pswd":"…"},
        "TLS":[]
    },
    "theme":"default",
    "saml": {
        "spMetadata":"…",
        "spAttributeFilterPolicy": {
            "id":"…",
            "attributeRules":[{"attr":"…","isPermitted":true}]
        },
        "saml2SSOProfile": {
            "signAssertions":"always",
            "encryptAssertions":"always",
            "encryptNameIds":"always",
            "includeAttributeStatement":true
        }
    }
}
```

Если изменение успешно, то сервер вернет HTTP 200, актуальные значения настроек приложения и новый *etag*.

Пример ответа:

```
HTTP/2 200
…
content-type: application/json
etag: 99_1658857631000

{
    "id":"test-app",
    "name":"…",
    …
    "oauth": {
        …
    },
    …
}
```

Если при редактировании приложения будет обнаружено, что данные в конфигурационном файле на сервере были изменены между получением etag и вызовом редактирования, то сервер вернет ответ с кодом *HTTP 412 Precondition Failed* и телом ошибки:

```
{
    "type":"process_error",
    "error":"cas_mismatch",
    "desc":"cas_mismatch"
}
```

Если при редактировании приложения возникла ошибка, что переданы неправильные данные, то сервер вернет ответ с кодом *HTTP 400 Bad Request* с описанием ошибок.

Пример ответа с ошибкой:

```
{
    "type": "input_error",
    "error": "wrong_values",
    "errors": [
        {
            "type": "input_error",
            "error": "json.error.mandatory.field",
            "desc": "json.error.expected.array",
            "pos": "oauth.redirectUriPrefixes"
        },
        …
    ]
}
```

Д.4. Удаление приложения

Для удаления приложения необходимо выполнить запрос методом DELETE по адресу *https://login.company.com/blitz/admin/api/v3/app/{appId}*.

Необходимые разрешения: *blitz_api_sys_app_chg*.

В запрос должен быть добавлен заголовок *If-Match*, содержащий последнее полученное от сервера значение *etag*.

Пример запроса:

```
DELETE /blitz/admin/api/v3/app/test-app HTTP/1.1
Authorization: Bearer cNw…Nz
Content-Type: application/json
If-Match: 99_1658857631000
```

Если приложение успешно удалено, то сервер вернет HTTP 204.

Если при удалении приложения будет обнаружено, что данные в конфигурационном файле на сервере были изменены между получением etag и вызовом удаления, то сервер вернет ответ с кодом *HTTP 412 Precondition Failed* и телом ошибки:

```
{
    "type":"process_error",
    "error":"cas_mismatch",
    "desc":"cas_mismatch"
}
```

Приложение Е. Вызов стороннего приложения регистрации пользователей

В Blitz Identity Provider можно настроить использование стороннего приложения регистрации пользователей. В этом случае Blitz Identity Provider сможет вызвать приложение регистрации пользователей со страницы входа (при переходе по ссылке «Зарегистрироваться») или в результате первого входа пользователя через внешний поставщик идентификации. При этом доступны следующие возможности:

- В случае если регистрация запущена в результате первого входа через внешний поставщик идентификации, то Blitz Identity Provider передаст приложению регистрации полученные из внешнего поставщика идентификации атрибуты. Приложение сможет их использовать для предзаполнения формы регистрации.
- Если пользователь успешно пройдет регистрацию, то он сможет продолжить процесс входа. Например, можно обеспечить автоматический вход зарегистрированного пользователя в приложение аналогично тому, как это происходит при использовании встроенного в Blitz Identity Provider приложения регистрации.

Для подключения к Blitz Identity Provider стороннего приложения регистрации необходимо на стороне веб-приложения регистрации поддержать сервисы в соответствии с описанными в последующих разделах требованиями.

Е.1. Сервис инициирования регистрации

Стороннее приложение регистрации должно предоставить HTTP POST сервис инициирования регистрации. Адрес сервиса задается в настройках Blitz Identity Provider (см. «Руководство администратора»).

Сервис должен принимать следующие параметры (в виде JSON):

- *id* – идентификатор заявки на регистрацию;
- *entryPoint* – сведения о точке входа. Возможны следующие значения:
  - *SOCIAL* – регистрация вызвана вследствие входа нового пользователя через внешний поставщик идентификации;
  - *WEB* – пользователь самостоятельно инициировал регистрацию (выбрал «Зарегистрироваться» на странице входа).
- *appId* – идентификатор приложения, в которое изначально хотел войти пользователь, в результате чего запустился процесс регистрации;
- *expires* – время окончания действия заявки на регистрацию. Указывается в Unix time, в секундах;

- *source* – источник сведений о пользователе (в случае получения сведений из внешнего поставщика входа). Содержит идентификатор внешнего поставщика входа (например, *esia:esia_1* в случае ЕСИА);
- перечень атрибутов, полученных из внешнего поставщика идентификации. Передаются атрибуты из настроек связывания учетных записей соответствующего внешнего поставщика идентификации.
- *hints* – подсказки, переданные в вызов формы входа. Например, тут может быть передан логин пользователя, в случае если пользователь инициировал самостоятельную регистрацию с формы входа, которая в свою очередь была открыта с параметром *login_hint*;
- *lang* – текущий язык интерфейса пользователя на странице входа.

Пример запроса (при вызове в режиме входа через ЕСИА):

```
POST /reg/url HTTP/1.1
Content-Type: application/json

{
   "id":"6DXDHyyiZ2hByUN-sCRUEdvAoQun7WwQ",
   "entryPoint":"SOCIAL",
   "appId":"portal",
   "expires":1608129702,
   "source":"esia:esia_1",
   "hints": {},
   "attrs":[
      {
         "esia_family_name":"Петров",
         "esia_given_name":"Иван",
         "esia_middle_name":"Сергеевич",
         "esia_passport":
"{\"issueDate\":\"01.01.2016\",\"stateFacts\":[\"EntityRoot\"],\"eTag\":\"452E4EEA3A9FBCD244766D6
549B8E7E616478BD2\",\"vrfStu\":\"VERIFIED\",\"type\":\"RF_PASSPORT\",\"issueId\":\"111001\",\"num
ber\":\"123456\",\"series\":\"4567\",\"issuedBy\":\"РУВД г.Москвы\",\"id\":38226}",
         "esia_trusted":true,
         "esia_id":"1000334562",
         "esia_gender": "M",
         "esia_birthdate": "01.01.1999",
         "esia_birthplace": "Москва",
         "esia_email": "johndoe@company.ru",
         "esia_snils": "123-456-789 12",
         "esia_inn": "123456789012",
         "esia_phone_number": "+7(999)1234567",
         "esia_liv_address":
{\"stateFacts\":[\"Identifiable\"],\"id\":24243131,\"type\":\"PRG\",\"addressStr\":\"г Москва, ул
Онежская\",\"fiasCode\":\"06690b31-d4ae-463d-ad12-
cf3963e0d7ed\",\"flat\":\"56\",\"countryId\":\"RUS\",\"house\":\"16\",\"zipCode\":\"125414\",\"st
reet\":\"Онежская\",\"region\":\"Москва\",\"vrfDdt\":\"0,10,0\",\"eTag\":\"0C7C02CA3BC3623B2628A7
603DA342792D5CE491\"}",
         "esia_reg_address":
{\"stateFacts\":[\"Identifiable\"],\"id\":24343142,\"type\":\"PRG\",\"addressStr\":\"г Москва, ул
Онежская\",\"fiasCode\":\"06690b31-d4ae-463d-ad12-
cf3963e0d7ed\",\"flat\":\"56\",\"countryId\":\"RUS\",\"house\":\"16\",\"zipCode\":\"125414\",\"st
reet\":\"Онежская\",\"region\":\"Москва\",\"vrfDdt\":\"0,10,0\",\"eTag\":\"0C7C02CA3BC3623B2628A7
603DA342792D5CE591\"}"
      }
   ],
   "lang": "ru"
}
```

Пример запроса (при нажатии пользователем «Зарегистрироваться» на странице входа):

```
POST /reg/url HTTP/1.1
Content-Type: application/json

{
```

```
   "id":"6DXDHyyiZ2hByUN-sCRUEdvAoQun7WwQ",
   "entryPoint":"WEB",
   "appId":"portal",
   "expires":1608129702,
   "hints": {},
   "attrs": {},
   "lang": "ru"
}
```

В ответ сервис инициирования регистрации должен вернуть либо HTTP-ответ для выполнения в браузере пользователя (например, код HTML-страницы или инициировать перенаправление пользователя в браузере на страницу регистрации), либо сообщение об ошибке.

Пример ответа:

```
HTTP/1.1 302 Found
Location: https://www.company.ru/register/
```

В результате пользователь будет перенаправлен из Blitz Identity Provider в стороннее приложение регистрации.

E.2. Сервис завершения регистрации

Когда пользователь в стороннем приложении регистрации ввел все данные, необходимые для регистрации учетной записи, стороннее приложение регистрации должно вызвать в Blitz Identity Provider сервис завершения регистрации учетной записи пользователя. Сервис вызывается методом POST по адресу *https://login.company.com/blitz/reg/api/v1/users/{id}*, где в качестве *id* в URL сервиса передается идентификатор заявки на регистрацию, ранее полученный от Blitz Identity Provider.

В запрос должен быть добавлен следующий заголовок, где *secret* – это присвоенные приложению при регистрации в Blitz Identity Provider *client_id:rest_secret* в формате Base64:

```
Authorization: Basic <secret>
```

Список атрибутов приведен в качестве образца. Содержание списка необходимо скорректировать в зависимости от конкретных настроек, сделанных при внедрении Blitz Identity Provider. См. «Руководство администратора».

Тело запроса должно содержать атрибуты регистрируемой учетной записи:

- *first_name* – фамилия;
- *name* – имя;
- *middle_name* – отчество;
- *phone_number* – номер мобильного телефона в виде составного объекта с атрибутами:
  - *value* – номер телефона в формате 7XXXXXXXXXX;
  - *verified* – признак, что телефон подтвержден – *true* или *false*;

- *email* – адрес электронной почты в виде составного объекта с атрибутами:
  - *value* – адрес электронной почты;
  - *verified* – признак, что адрес подтвержден – *true* или *false*;
- *password* – пароль для создаваемой учетной записи пользователя (должен соответствовать настроенной парольной политике).

Пример запроса (регистрация с подтвержденными email и телефоном):

```
POST /blitz/reg/api/v1/users/6DXDHyyiZ2hByUN-sCRUEdvAoQun7WwQ HTTP/1.1
Authorization: Basic YXBwX2lkOmFwcF9zZWNyZXQ=
Content-Type: application/json
Cache-Control: no-cache

{
    "first_name":"Иванов",
    "name":"Иван",
    "middle_name":"Иванович",
    "phone_number": {
        "value": "79991234567",
        "verified": true
    },
    "email": {
        "value":"mail@example.com",
        "verified": true
    },
    "password":"QWErty$123"
}
```

В ответ Blitz Identity Provider в случае успешного завершения регистрации вернет JSON со следующими данными:

- *subject* – идентификатор зарегистрированного пользователя;
- *origin* – ссылку, на которую необходимо направить браузер пользователя;
- *cookies* – куки, которые нужно установить при перенаправлении браузера пользователя на общем с Blitz Identity Provider домене;
- *instanceId*, *instructions* – прочие технологические сведения, которые нужно проигнорировать.

Пример ответа:

```
{
   "instanceId": "amRiY2lkcG9zdGdyZXM6YzhjMGExYzEtYzdmYS00ZDg3LWFiYmMtZTNiYzg1YTk4",
   "subject":"5cffd68f-2cb8-4f7a-b0f3-9fa69a1fbbcd",
   "context":"6DXDHyyiZ2hByUN-sCRUEdvAoQun7WwQ",
   "cookies": [{
      "name": "css",
      "value": "TSQA-AruOjUNphGZ984eLgzT_ROebNiBsjyjEg4n-nL-PdsiXqq"
    }],
    "origin": "/blitz/profile?",
    "instructions": []
}
```

После перенаправления сторонним приложением регистрации браузера пользователя по ссылке, указанной в *origin*, и с указанными *cookies* Blitz Identity Provider создаст сессию и обеспечит вход пользователя в приложение, для входа в которое пользователь осуществил регистрацию учетной записи.