

**BLITZ IDENTITY PROVIDER 2.7**

**Руководство по интеграции**

**Москва, 2016**

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b> .....	<b>3</b>
<b>1. ПОДКЛЮЧЕНИЕ ПРИЛОЖЕНИЯ К BLITZ IDENTITY PROVIDER ПО SAML</b> .....	<b>3</b>
1.1. НАСТРОЙКИ В КОНСОЛИ УПРАВЛЕНИЯ BLITZ IDENTITY PROVIDER.....	3
1.2. ОБЕСПЕЧЕНИЕ ВЗАИМОДЕЙСТВИЕ ПРИЛОЖЕНИЯ С BLITZ IDENTITY PROVIDER ПО SAML.....	6
<b>2. ПОДКЛЮЧЕНИЕ ПРИЛОЖЕНИЯ К BLITZ IDENTITY PROVIDER ПО OAUTH 2.0 / OPENID CONNECT 1.0</b> .....	<b>9</b>
2.1. НАСТРОЙКИ В КОНСОЛИ УПРАВЛЕНИЯ BLITZ IDENTITY PROVIDER.....	9
2.2. ОБЕСПЕЧЕНИЕ ВЗАИМОДЕЙСТВИЕ ПРИЛОЖЕНИЯ С BLITZ IDENTITY PROVIDER ПО OAUTH 2.0 / OPENID CONNECT 1.0.....	12
2.2.1. Авторизация приложения с явного разрешения пользователя.....	13
2.2.2. Авторизация приложения посредством предъявления логина и пароля.....	17
2.2.3. Получение данных и идентификационной информации.....	19
2.2.4. Обеспечение логгута при использовании OAuth 2.0.....	22
<b>3. РЕКОМЕНДАЦИИ ПО ДИЗАЙНУ КНОПОК И ССЫЛОК В ПРИЛОЖЕНИИ, НАЖАТИЕ КОТОРЫХ ИНИЦИИРУЕТ ИДЕНТИФИКАЦИЮ И АУТЕНТИФИКАЦИЮ</b> .....	<b>23</b>
<b>4. ВЫЗОВ СЕРВИСА РЕГИСТРАЦИИ И ИЗМЕНЕНИЯ ДАННЫХ ПОЛЬЗОВАТЕЛЯ</b> .....	<b>24</b>
4.1. РЕГИСТРАЦИЯ СИСТЕМЫ-КЛИЕНТА.....	24
4.2. ВЫЗОВ СЕРВИСА РЕГИСТРАЦИИ НОВОГО ПОЛЬЗОВАТЕЛЯ.....	24
4.3. ВЫЗОВ СЕРВИСА ИЗМЕНЕНИЯ ДАННЫХ ПОЛЬЗОВАТЕЛЯ.....	26
4.4. ПРИВЯЗКА НОТР/ГОТР-ГЕНЕРАТОРА К УЧЕТНОЙ ЗАПИСИ ПОЛЬЗОВАТЕЛЯ.....	26

## Введение

В данном документе описываются следующие действия, необходимые для интеграции приложений с Blitz Identity Provider:

- подключение приложений к Blitz Identity Provider по SAML 2.0 для проведения идентификации и аутентификации пользователя, а также для получения данных о нем;
- подключение приложения к Blitz Identity Provider по OAuth 2.0 / Open ID Connect 1.0 для проведения идентификации и аутентификации пользователя, а также для получения данных о нем;
- подключение приложения к Blitz Identity Provider для проведения регистрации пользователя и изменения его данных.

## 1. Подключение приложения к Blitz Identity Provider по SAML

Для подключения приложения необходимо:

- задать настройки подключения приложения в консоли управления Blitz Identity Provider.
- реализовать на стороне подключаемого приложения программный код взаимодействия с поставщиком идентификации по какому-либо из стандартов: SAML 1.0, SAML 1.1, SAML 2.0.

### 1.1. Настройки в консоли управления Blitz Identity Provider

#### 1.1.1. Добавление приложения

Перейти в раздел *Приложения* консоли и выбрать пункт «Добавить приложение» (рис.

1).

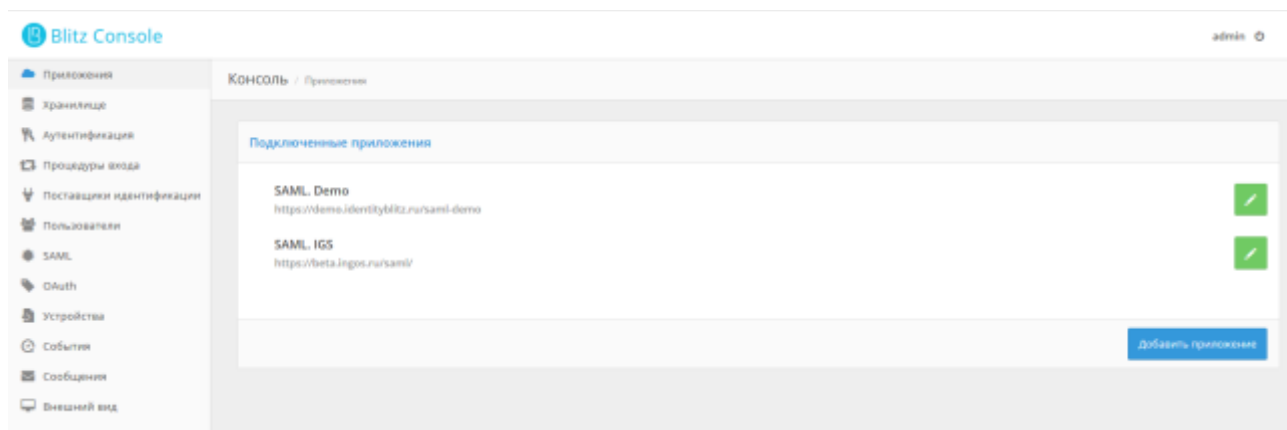


Рисунок 1 – Перечень подключенных приложений

Это действие запустит мастер подключения нового приложения, работа которого включает в себя следующие шаги.

**Шаг 1.** Базовые настройки. Указать идентификатор подключаемого приложения (при подключении по SAML 2.0 он соответствует entityID поставщика услуг, при задании идентификатора недопустимо использовать **двоеточие**), его название и домен, т.е. URL, по которому доступно данное приложение (рис. 2).

Рисунок 2 – Первый шаг подключения приложения

**Шаг 2.** Специфические настройки. После добавления приложения необходимо перейти к редактированию специфических настроек SAML, нажав на кнопку «Редактировать» (🔗). Далее:

- выбрать протокол взаимодействия – SAML – и перейти к его конфигурированию, нажав по ссылке «Сконфигурировать»;
- задать файл метаданных SAML подключаемого приложения (метаданные поставщика услуг);
- указать, должны ли передаваемые подключаемому приложению от Blitz Identity Provider утверждения SAML подписываться («Подписывать утверждения»);
- указать, должны ли передаваемые подключаемому приложению от Blitz Identity Provider утверждения SAML быть зашифрованы («Шифровать утверждения»);
- указать, должны ли передаваемые подключаемому приложению от Blitz Identity Provider идентификаторы пользователя (NameIds) быть зашифрованы («Шифровать идентификаторы (NameIds)»);
- указать, должны ли передаваться приложению утверждения SAML;
- указать перечень передаваемых приложению утверждений SAML.

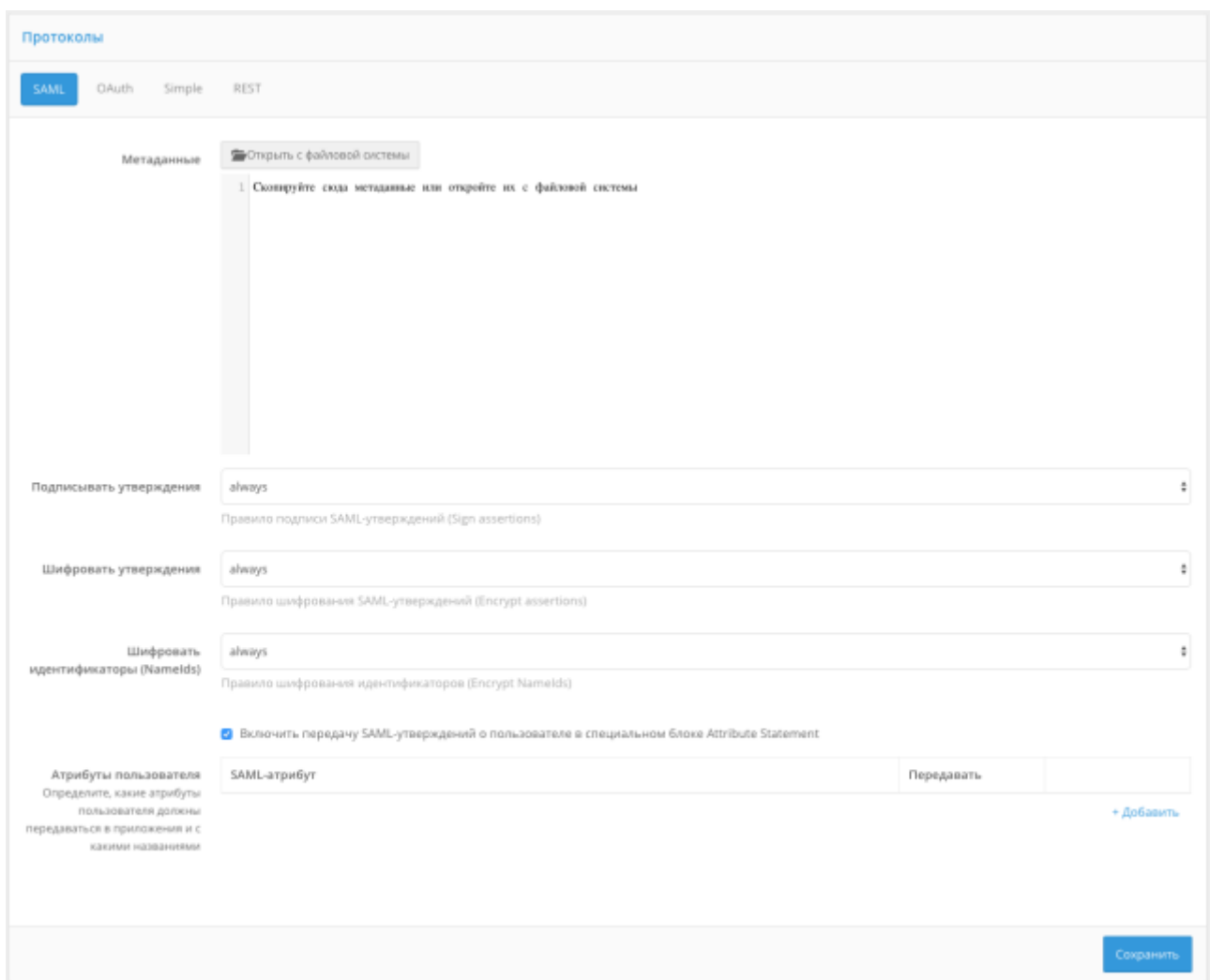


Рисунок 3 – Настройка приложения с использованием консоли (протокол SAML)

### 1.1.2. Создание утверждений SAML

Чтобы создать новое утверждение SAML и сделать его возможным для передачи приложению, необходимо перейти в раздел *SAML* консоли и выбрать пункт «Добавить новый SAML-атрибут» (рис. 4).

При создании SAML-атрибута нужно указать:

- Название SAML-атрибута – текстовая строка с названием атрибута, требующимся для подключаемого приложения.
- Источник сведений для SAML-атрибута – выбрать из списка атрибут хранилища.
- Кодировщик SAML – выбрать тип кодировщика, название, короткое название и формат.

Также в разделе *SAML* консоли управления можно посмотреть URL, по которому расположены метаданные поставщика идентификации Blitz Identity Provider, которые будут необходимы подключаемому приложению (поставщику сервиса).

**Свойства**

URL с метаданными Blitz Identity Provider: </blitz/saml/profile/Metadata/SAML>

При подключении приложений по протоколу SAML в настройках этих приложений должна быть указана данная ссылка, либо загружен файл с метаданными

**Атрибуты**

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

Поиск...

mail  
email  
surname

+ Добавить новый SAML-атрибут

**Свойства SAML-атрибута**

Название: surname

Источник: sn

Сохранить

**Кодировщик**

Тип: SAML2String

Название: surname

Короткое название: sn

Формат имени: urn:oasis:names:tc:SAML:2.0:nameid-format:transient

Удалить

+ Добавить кодировщик

Удалить SAML-атрибут

Рисунок 4 – Настройка SAML-атрибутов

## 1.2. Обеспечение взаимодействие приложения с Blitz Identity Provider по SAML

Для подключения к Blitz Identity Provider в целях идентификации и аутентификации пользователей приложение может использовать стандарт SAML<sup>1</sup> версий 1.0, 1.1, 2.0. При этом процесс взаимодействия приложения и Blitz Identity Provider должен быть построен в соответствии с профилем SAML Web Browser SSO Profile<sup>2</sup>.

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

<sup>1</sup> <http://saml.xml.org/saml-specifications>

<sup>2</sup> <http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf>

Blitz Identity Provider в соответствии с принятой в SAML терминологией представляет собой поставщик идентификации (Identity Provider). Подключаемое к Blitz Identity Provider приложение в свою очередь называется поставщиком услуг (Service Provider).

Для возможности осуществлять взаимодействия поставщик услуг и поставщик идентификации предварительно должны обмениваться настройками взаимодействия, описанными в форме XML-документов и называемых метаданными. Поставщик услуг должен получить настройки Blitz Identity Provider, называемые метаданными поставщика идентификации. Метаданные поставщика идентификации Blitz Identity Provider доступны по URL `http(s)://<hostname>:<port>/blitz/saml/profile/Metadata/SAML`. В свою очередь, метаданные поставщика услуг (приложения) должны быть зарегистрированы в Blitz Identity Provider (см. инструкцию в разделе 1.1.1).

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 HTTP(s), 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 HTTP(s), вызов поставщика услуг осуществляется через HTTP POST.

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

Процесс взаимодействия приложения и Blitz Identity Provider с использованием SAML приведен на рисунке 5.

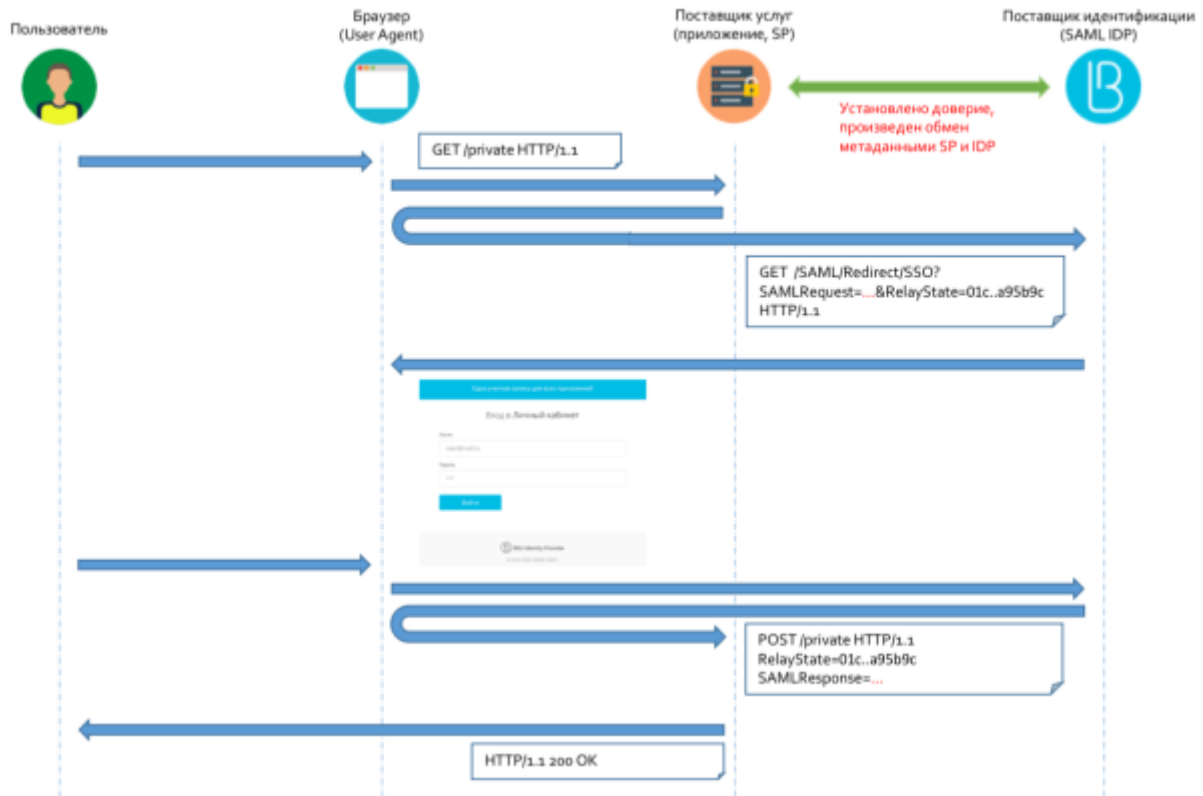


Рисунок 5 – Схема взаимодействия приложения с Blitz Identity Provider по SAML.

Так как самостоятельная разработка программного обеспечения клиента SAML является трудоемкой задачей, а ошибки в реализации могут быть чреваты угрозами безопасности, то рекомендуется при интеграции приложения по SAML использовать существующие популярные библиотеки SAML-клиентов: OIOSAML<sup>3</sup> (Java, .NET), OpenSAML<sup>4</sup> (Java), Spring Security SAML<sup>5</sup> (Java), SimpleSAMLphp<sup>6</sup> (PHP), ruby-saml<sup>7</sup> (Ruby on Rails).

Подключенное по SAML к Blitz Identity Provider приложение также может предусматривать возможность реализации единого выхода (логаута). Для этих целей Blitz Identity Provider поддерживает SAML Single Logout Profile<sup>8</sup>. Приложение может

<sup>3</sup> <http://digitaliser.dk/group/42063/resources>

<sup>4</sup> <https://shibboleth.net/products/opensaml-java.html>

<sup>5</sup> <http://projects.spring.io/spring-security-saml/>

<sup>6</sup> <https://simplesamlphp.org/>

<sup>7</sup> <https://rubygems.org/gems/ruby-saml/>

<sup>8</sup> <https://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf>



направить в Blitz Identity Provider SAML-запрос <LogoutRequest> и в случае успешного завершения единого логута получить от Blitz Identity Provider SAML-ответ <LogoutResponse>. Если приложение должно быть задействовано в едином логaute, 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>.

Если приложению не требуется единый логает, и достаточно при инициировании пользователем логата только завершить собственную сессию приложения и глобальную сессию в Blitz Identity Provider, то вместо реализации сложного протокола в соответствии с SAML Single Logout Profile можно для простого логата инициировать HTTP Redirect на следующий URL в Blitz Identity Provider, передав в качестве query-параметра адрес возврата `http(s)://<hostname>:<port>/blitz/login/logout?redirect=http(s)://redirect_url`. Если Blitz Identity Provider успешно завершит логает, то он перенаправит пользователя по переданному URL.

## 2. Подключение приложения к Blitz Identity Provider по OAuth 2.0 / OpenID Connect 1.0

Для подключения приложения необходимо:

- задать настройки подключения приложения в консоли управления Blitz Identity Provider;
- реализовать на стороне подключаемого приложения программный код взаимодействия с поставщиком идентификации по стандарту OpenID Connect 1.0.

### 2.1. Настройки в консоли управления Blitz Identity Provider

#### 2.1.1. Добавление приложения

Перейти в раздел *Приложения* консоли и выбрать пункт «Добавить приложение» (рис. 1).

Это действие запустит мастер подключения нового приложения, работа которого включает в себя следующие шаги.

**Шаг 1.** Базовые настройки. Указать идентификатор подключаемого приложения (при подключении по OAuth 2.0 он соответствует `client_id`, при задании идентификатора недопустимо использовать **двоеточие**), его название и домен, т.е. URL, по которому доступно данное приложение (рис. 2).

**Шаг 2.** Специфические настройки. После добавления приложения необходимо перейти к редактированию специфических настроек OAuth 2.0, нажав на кнопку «Редактировать» (🔗). Далее:

- выбрать протокол взаимодействия – OAuth – и перейти к его конфигурированию, нажав по ссылке «Сконфигурировать»;
- указать (или оставить сгенерированный системой) секретный ключ подключаемого приложения (`client_secret`), который должен использоваться подключенным приложением при обращении к Blitz Identity Provider;
- предопределенная ссылка возврата (`redirect_uri`) – URL, на который по умолчанию будет переадресован пользователь после прохождения авторизации – опционально;
- префиксы ссылок возврата – префикс используется для проверки ссылок возврата (`redirect_uri`). Если в запросе на аутентификацию указана ссылка возврата и она не соответствует ни одному из указанных префиксов, то в аутентификации будет отказано;
- разрешения по умолчанию – разрешения (`scope`), которые будут по умолчанию выданы приложению после аутентификации. Если не указаны, то в запросе на аутентификацию всегда должны быть явно прописаны требуемые разрешения. Важно! Разрешения должны быть сконфигурированы в разделе *OAuth* (см. далее).

Протоколы

SAML OAuth Simple REST

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

URL для авторизации `/REST/oauth/authorize`  
На данный URL (обязательно с протоколом) должен быть направлен запрос на проведение авторизации пользователя

URL для получения и обновления маркера `/REST/oauth/token`  
На данный URL (обязательно с протоколом) должен быть направлен запрос на получение или обновление маркера доступа

Настройки взаимодействия с приложением

Секрет (`client_secret`):   
Секретный ключ подключаемого приложения (`client_secret`). Если указан, то именно этот секрет должен использоваться подключаемым приложением при обращении к Blitz Identity Provider

Предопределенная ссылка возврата (`redirect_uri`):   
URL, на который по умолчанию будет переадресован пользователь, после прохождения авторизации (`redirect_uri`)

Префиксы ссылок возврата:   
Для добавления нового префикса вводите его и нажмите Enter  
 Префикс используется для проверки ссылок возврата (`redirect_uri`). Если в запросе на аутентификацию указана ссылка возврата и она не соответствует ни одному из указанных префиксов, то в аутентификации будет отказано.  
 Пусто обязательно для ввода

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

Рисунок 6 – Настройка приложения с использованием консоли (протокол OAuth)

### 2.1.2. Конфигурирование OAuth 2.0 / OpenID Connect 1.0

В разделе *OAuth* консоли управления можно посмотреть следующие URL, которые далее потребуются для выполнения запросов:

- для проведения авторизации и аутентификации;
- для получения и обновления маркера.

При необходимости можно изменить время жизни маркера доступа.

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

- название разрешения;
- описание разрешения (оно будет отображаться пользователю на странице согласия на предоставление доступа);
- атрибуты пользователя, которые будут предоставлены по данному разрешению (атрибуты должны быть определены в разделе *Хранилище*).

Для корректной работы аутентификации по OpenID Connect 1.0 нужно убедиться, что разрешение с названием *openid* определено на этой вкладке. Также можно прописать атрибуты, передаваемые по этому разрешению<sup>9</sup>.

---

<sup>9</sup> В этом случае указанные данные могут быть получены по маркеру доступа (access token), выданному на разрешение *openid*.

The screenshot shows the configuration interface for the Blitz Identity Provider, divided into two main sections: "Свойства" (Properties) and "Настройка scopes" (Scope Configuration).

**Свойства (Properties):**

- URL для авторизации:** /blitz/oauth/ae. Description: На данный URL (authorization endpoint) должен быть направлен запрос на проведение авторизации пользователя.
- URL для получения и обновления маркера:** /blitz/oauth/te. Description: На данный URL (token endpoint) должен быть направлен запрос на получение или обновление маркера доступа.
- Время жизни маркера доступа, сек:** 3600.
- Аутентификация систем-клиентов с использованием Proxy TLS.** Для аутентификации систем по Proxy TLS должно быть настроено взаимодействие через прокси-сервер и обеспечено установление двустороннего TLS-соединения. В поле Common Name (CN) сертификата системы должен быть указан домен системы.

**Настройка scopes (Scope Configuration):**

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

Название разрешения	Описание	Атрибуты пользователя
openid	Информация, позволяющая провести идентификацию и аутентификацию	
profile	Данные пользователя	* sn * sn * mail * givenName * uid * telephoneNumber

Buttons: + Добавить scope

Рисунок 7 – Специфические настройки протокола OAuth 2.0

После сохранения данных настроек можно перейти к настройке подключенного приложения.

## 2.2. Обеспечение взаимодействие приложения с Blitz Identity Provider по OAuth 2.0 / OpenID Connect 1.0

В данном разделе рассмотрено взаимодействие приложения с Blitz Identity Provider в рамках следующих моделей авторизации приложения:

- Authorization Code Grant – приложение авторизуется с явного разрешения пользователя, по этой же схеме реализована аутентификация на основе OpenID Connect 1.0.
- Resource Owner Password Credentials Grant – приложение авторизуется посредством предъявления логина и пароля пользователя (данные которого предоставляются).

Более подробно о взаимодействии приложений с поставщиком идентификации по протоколам OAuth 2.0 и OpenID Connect 1.0 можно ознакомиться в спецификациях:

- Спецификация по OpenID Connect 1.0 – [http://openid.net/specs/openid-connect-core-1\\_0.html](http://openid.net/specs/openid-connect-core-1_0.html).
- Спецификация по OAuth 2.0 – <https://tools.ietf.org/html/rfc6749>.

## 2.2.1. Авторизация приложения с явного разрешения пользователя

### 2.2.1.1. *Общая схема авторизации*

Эта модель авторизации используется в случаях, когда приложение в явном виде получает разрешение на доступ к ресурсу (например, данными пользователя) со стороны владельца ресурса (пользователя).

В общем виде схема взаимодействия выглядит следующим образом, для упрощения рассматриваем пользователя в качестве владельца ресурса (данных о себе):

- приложение запрашивает Blitz Identity Provider доступ к ресурсу;
- приложение получает разрешение на доступ (authorization grant) в виде авторизационного кода (Blitz Identity Provider предварительно запрашивает это разрешение у пользователя);
- приложение запрашивает маркер доступа, предъявив авторизационный код Blitz Identity Provider;
- Blitz Identity Provider аутентифицирует приложение, проверяет авторизационный код и выдает маркер доступа и маркер обновления;
- приложение запрашивает у Blitz Identity Provider защищенный ресурс, предъявляя маркер доступа;
- Blitz Identity Provider проверяет маркер доступа, если он валиден, то разрешает доступ к защищенному ресурсу;

Далее возможны следующие шаги:

- приложение через некоторое время запрашивает с помощью выданного ранее маркера доступа к защищенному ресурсу;
- Blitz Identity Provider проверяет маркер, обнаруживает, что срок его действия истек, возвращает сообщение об ошибке;
- приложение обращается к Blitz Identity Provider за получением нового маркера доступа, предъявляя маркер обновления;
- Blitz Identity Provider проверяет валидность маркера обновления и возвращает два новых маркера: доступа и обновления.

Схема взаимодействия представлена на рисунке 8. После того, как приложение получило маркер доступа, оно может неоднократно обращаться за получением соответствующего защищенного ресурса, пока не истечет срок действия этого маркера. Когда это произойдет, системе-клиенту потребуется получить новый маркер доступа.

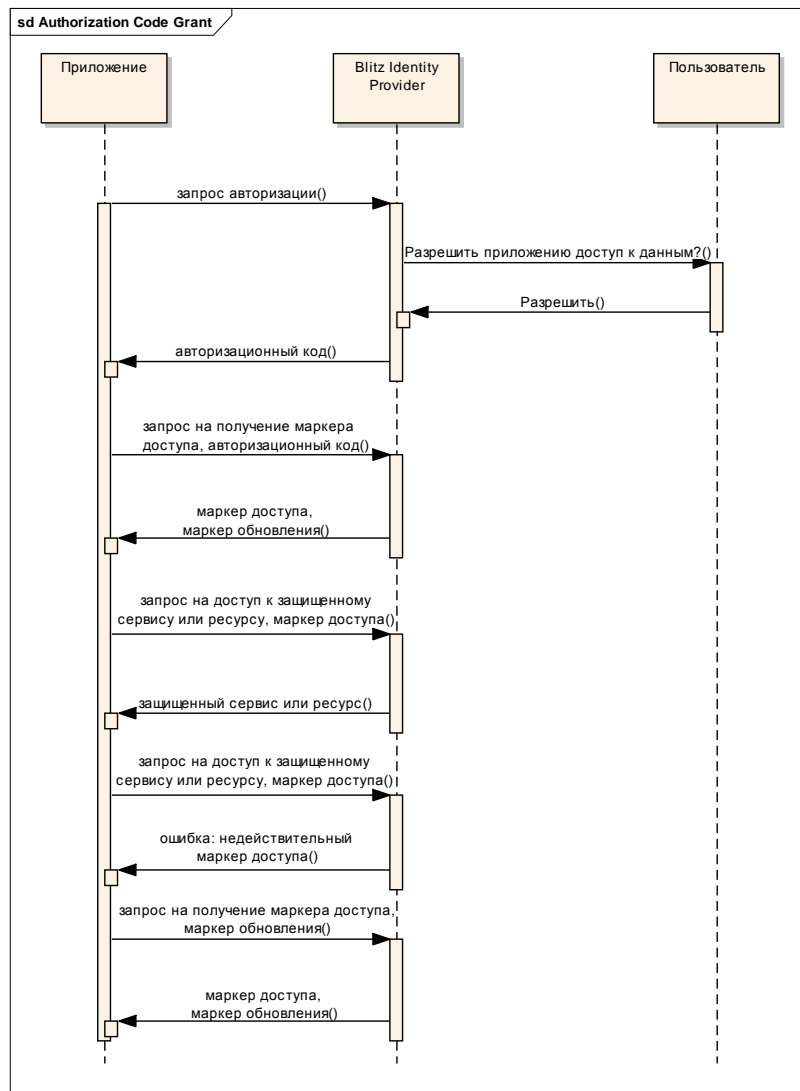


Рисунок 8 – Общая схема взаимодействия при получении маркера доступа с помощью авторизационного кода

#### 2.2.1.2. Схема аутентификации на основе OpenID Connect 1.0

Аутентификация на основе OpenID Connect 1.0 включает в себя те же основные шаги, что и авторизация на основе OAuth 2.0:

- приложение отправляет запрос на аутентификацию в адрес Blitz Identity Provider;
- Blitz Identity Provider аутентифицирует пользователя;
- Blitz Identity Provider получает согласие пользователя на проведение аутентификации и на передачу идентифицирующей информации о нем в приложение;
- Blitz Identity Provider перенаправляет пользователя обратно в приложение и передает авторизационный код;
- приложение формирует запрос с использованием авторизационного кода на получения маркера идентификации;
- приложение получает ответ, содержащий необходимый маркер идентификации;

- приложение проводит валидацию маркера идентификации и извлекает из маркера идентификатор пользователя.

Следует учесть, что в ответ на запрос на получение авторизационного кода Blitz Identity Provider всегда возвращает маркер доступа и маркер обновления, выданные на запрошенное разрешение (scope). Иными словами, если в запросе указано единственное разрешение *openid*, то в ответ Blitz Identity Provider вернет маркер идентификации, а также маркер доступа и маркер обновления. Маркер доступа можно использовать для получения тех данных о пользователе, которые явно прописаны в консоли управления.

### 2.2.1.3. Примеры запросов

Для проведения авторизации и/или аутентификации приложение должно направить пользователя на страницу предоставления прав доступа (URL указан в консоли управления), передав в качестве параметров:

- `scope` – запрашиваемые разрешения (scope), для проведения аутентификации должно быть передано разрешение `openid`;
- `response_type` – тип ответа (принимает значение “code”);
- `redirect_uri` – ссылка для возврата пользователя;
- `state` – набор случайных символов, имеющий вид 128-битного идентификатора запроса (используется для защиты от перехвата), это же значение будет возвращено в ответе – опциональный параметр;
- `nonce` – набор случайных символов (используется для защиты от перехвата), это же значение будет возвращено в маркере идентификации – опциональный параметр (только для аутентификации по OpenID Connect);
- `client_id` – идентификатор клиента;
- `prompt` – параметр, определяющий способ ответа при проведении аутентификации (только для OpenID Connect), может принимать значения:
  - `none` – в этом случае будет проведена фоновая проверка аутентификации пользователя без перенаправления его на страницу входа: если пользователь имеет активную сессию в Blitz Identity Provider (и пользователь ранее давал доступ приложению), то будет возвращен авторизационный код; если он такой сессии не имеет, то будет возвращено сообщение об ошибке;
  - `login` – в этом случае будет проведена принудительная проверка аутентификации пользователя с перенаправлением его на страницу входа (даже если он имеет активную сессию).

Если параметр `prompt` не задан, то страница аутентификации появляется только тогда, когда пользователь не имеет активной сессии.

Пример запроса на получение авторизационного кода (запрошен маркер доступа с разрешением `test`):

```
https://<hostname>/blitz/oauth/ae?scope=test&response_type=code&redirect_uri=http%3A%2F%2Flocalhost%2Fsuccess&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f&client_id=localhost%2Fdemo2
```

Пример запроса на получение авторизационного кода (запрошена аутентификация, а также маркер доступа с разрешением `test`) с параметром `nonce`:

```
https://<hostname>/blitz/oauth/ae?scope=test+openid&response_type=code&redirect_uri=http%3A%2F%2Flocalhost%2Fsuccess&state=342a2c0c-4cd6-4cd6-b328-b67d9baf6a7f&client_id=localhost%2Fdemo2&nonce=987654321
```

Пример запроса на получение авторизационного кода (запрошена аутентификация) с параметрами `nonce` и `prompt`:

```
https://<hostname>/blitz/oauth/ae?scope=openid&response_type=code&redirect_uri=http%3A%2F%2Flocalhost%2Fsuccess&state=342a2c0c-b328-b328-b328-b67d9baf6a7f&client_id=localhost%2Fdemo2&nonce=987654321&prompt=login
```

Пример ответа со значением авторизационного кода (`code`) и параметром `state`:

```
http://localhost/success?code=f954nEzQ08DXju4wxGbSSfCX7TkZ1GvXUR7TzVus8fGnu4AUI-YIosgax-BLXMeQQAlasD6CN2qG_-0KXK5NjARoKykhuR9IpbuzqeFxS0&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f
```

Пример ответа для случая с параметром `prompt=none`, когда пользователь не аутентифицирован:

```
http://localhost/success?error=login_required&error_description=The%2BAuthorization%2BServer%2Brequires%2BEnd-User%2Bauthentication.%2BThis%2Berror%2Bmay%2Bbe%2Breturned%2Bwhen%2Bthe%2Bprompt%2Bparameter%2Bvalue%2Bin%2Bthe%2BAuthentication%2BRequest%2Bis%2Bnone%252C%2Bbut%2Bthe%2BAuthentication%2BRequest%2Bcannot%2Bbe%2Bcompleted%2Bwithout%2Bdisplaying%2Ba%2Buser%2Binterface%2Bfor%2BEnd-User%2Bauthentication.&state=342a2c0c-d9ef-4cd6-b328-b67d9baf6a7f
```

После получения авторизационного кода приложение должно обменять его на маркер доступа / маркер идентификации. Для этого оно должно сформировать запрос методом POST на URL для получения и обновления маркера. Запрос должен содержать заголовок `Authorization` со значением `Basic {secret}`, где `secret` – это `"entity_id:secret"` (например, `localhost/demo2:1234567890`) в формате `base64`. Пример заголовка:

```
Authorization: Basic bG9jYWxob3N0L2RlbW8yOjEyMzQ1Njc4OTA=
```

Тело запроса должно содержать следующие параметры:

- `<code>` – значение авторизационного кода, который был ранее получен;
- `<grant_type>` – принимает значение `"authorization_code"`, если авторизационный код 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 авторизационного кода).

Пример запроса:

```
POST /blitz/oauth/te HTTP/1.1
Host: <hostname>
Authorization: Basic bG9jYWxob3N0L2RlbW8yOjEyMzQ1Njc4OTA=
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache
```



```
grant_type=authorization_code&code=FLZHSMmqXTxU8EFW3bse7qOiyqarfbdbxGadBVlffFENxBltpREKs7dEkN33dFvNyUg
gDb2XpP4nyTIUIZUMf4xBDreSmKrOkrVV7qK8GU&redirect_uri=http%3A%2F%2Flocalhost%2Fsuccess
```

В ответ возвращается маркер доступа и маркер обновления:

```
{
  "access_token": "DsefHaGZ_n0Cw98-Uyhrhuhj_lhxr1WOO1rqtX7_8aVdHQtrV5CHMpqLykrfEmqft351E-
  QKk_pxTduH6jAxmZlpwaeCpYaTxK7_5IAM5E",
  "expires_in": 3600,
  "scope": "test2",
  "refresh_token": "RBBM7i_eM88jeXS0R2SI0sIV7cFG8eIOr7vcxEFDBza91qdwbfIO-
  1AxbdxdGGEZKJRgT5jNcSTOcwubRo6rCzOkMLbjEFTILKvrlx4eVg",
  "token_type": "bearer"
}
```

Если запрошена аутентификация (*scope* – *openid*), то будет также возвращен маркер идентификации:

```
{
  "id_token":
  "eyJhbGciOiJIUzU1NiJ9.eyJub25jZSI6IjE0NzY1NDMyMSIsImV4cCI6MTQ0NTAwNDc3NywiaWF0IjoxNDQ0OTk0MjE5LCJzd
  Wl0IjZemF5dHNldkZkZXUyYmV4dHotaWRwLmV4YyIsImF1ZCI6WyJsb2NhbGhvc3QvZGVtbzliXSwiaXNzIjoiaHR0cHM6Ly9
  kZXUyYmV4dHouc2VhZG9mZC5sb2MifQ.Ckt_dr9J4k514MluQEDY88A026NWzdCcIIIDxOXAzXue52eQlNVxaDXIQ8F0IyG2T
  -2KmeSeVG7UK4RoYWhYWT7Skn5NfF-gFJIfXB4NfUGwt5iic5UGQkp-xGqKzTxCKduuWrp-oVb69Bd8fFCeFYTVhB-
  iWR_V1yZhYTipQt6rLVKaqEFPvRv6iN_cGGIr7k0EJtEvF6Y71ktf6ERnhCXsLn78mPeJv0H0jk6dWW19JJxpZC6RvUEqv4Q8q9
  2xXh7Rj0MQUErNk03J74QLIdn3xYke0ch20fOauMPLYXOc0-cPwo5kjF5zK6-c1chrwfk2FFMCi1bGYpgeICEssQ",
  "access_token": "dO-xymwduYR8uFqvYYK1ghpk-tqantG5PstfomddIO5d2-
  BVzVVaNdHuJYdWxpL__c8MsLWB8IwmiUIEep53tnZR2mflAnYiguE0UyUZNB",
  "expires_in": 3600,
  "scope": "test openid",
  "refresh_token": "1IEWX7IE7SESIXRJNJeN66oPdzoFSfOGDLB3foAgXUDfuzYf1aXS1FuJX5JM50mmDqHuX75t-
  DrtvM0ISa82vs_t0NI901AcAilduRzZ8Iw",
  "token_type": "bearer"
}
```

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>` – значение маркера обновления;
- `<grant_type>` – принимает значение “refresh\_token”, если маркер обновления обменивается на маркер доступа.

Пример запроса:

```
POST /blitz/oauth/te HTTP/1.1
```

```
Host: <hostname>
```

```
Authorization: Basic bG9jYWxob3N0L2RlbW8yOjE5MzQ1Njc4OTA=
```

```
Content-Type: application/x-www-form-urlencoded
```

```
Cache-Control: no-cache
```

```
grant_type=refresh_token&refresh_token=jj2DAYsPi-
```

```
aXMeSrQ7WiucRQKtz4_swxXdJDqVpC9laxIq_LhPmxiA8tPI7TocUkkgkXFqp28P0nC6af6STHP46TvvO1gBTx1-aW9rPbQ
```

## 2.2.2. Авторизация приложения посредством предъявления логина и пароля

### 2.2.2.1. Общая схема

Эта модель контроля доступа предполагает, что приложение получает разрешение на доступ к ресурсу (например, данными пользователя) в результате предъявления логина и пароля пользователя. Эту модель авторизации рекомендуется применять только в случае, если между приложением и Blitz Identity Provider установлены отношения доверия.

В общем виде схема взаимодействия выглядит следующим образом:

- приложение запрашивает Blitz Identity Provider доступ к ресурсу, предъявляя логин и пароль пользователя;
- Blitz Identity Provider аутентифицирует приложение, проверяет логин и пароль пользователя и выдает маркер доступа и маркер обновления;
- приложение запрашивает у Blitz Identity Provider защищенный ресурс, предъявляя маркер доступа.

Дальнейшее взаимодействие осуществляется по стандартной схеме.

Схема взаимодействия представлена на рисунке 9.

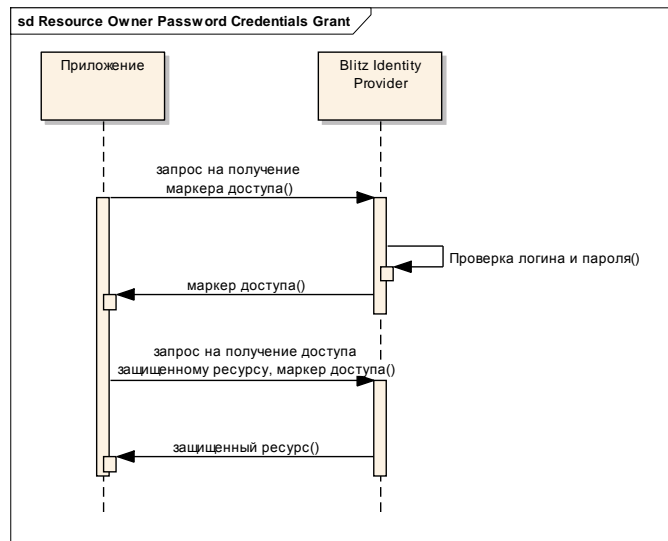


Рисунок 9 – Общая схема взаимодействия при получении маркера доступа посредством предъявления логина и пароля

#### 2.2.2.2. Примеры запросов

Для получения маркера приложение должно отправить POST на URL для получения и обновления маркера. Запрос должен содержать описанный ранее заголовок `Authorization`, а в теле – следующие параметры:

- `<scope>` – запрашиваемые `scope`;
- `<grant_type>` – принимает значение “password”, если для получения маркера доступа предъявляется логин и пароль пользователя;
- `<username>` – имя пользователя;
- `<password>` – пароль пользователя.

```

POST /blitz/oauth/te HTTP/1.1
Host: <hostname>
Authorization: Basic bG9jYWxob3N0L2RlbW8yOjEyMzQ1Njc4OTA=
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

grant_type=password&username=szaytsev&password=QWErty1234&scope=test
  
```

Пример ответа:

```
{
  "access_token": "C1NpcWTI_vgKBQPC7Mpf7uKPUQK1-1tczb79We5FQmA WFTikDAh6SyUgnAFHrLaZoVliqSWRg3_fNYnKZQr0P2dRxtSFrUCkVsY3oLB-pi4",
  "expires_in": 3600,
  "scope": "test",
  "refresh_token": "FgCbFwuUh7DUTK-_-iChKgW8WZAUr_KOJTZqVKGtoPm62UO8klg1btm2bENnhy9FXsseUM87udPqFdCA_d-7E5a4UMmkZCaIXhDQ3yeib0E",
  "token_type": "bearer"
}
```

### 2.2.3. Получение данных и идентификационной информации

#### 2.2.3.1. Данные пользователя

Для запроса данных о пользователе необходимо выполнить запрос методом GET по адресу получения данных пользователя. В запрос должен быть добавлен следующий заголовок:

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

В заголовке <access token> – это маркер доступа, предварительно полученный от Blitz Identity Provider.

Пример запроса:

```
GET /blitz/oauth/me HTTP/1.1
Host: <hostname>
Authorization: Bearer NINxnizbgYYQg94vEd6MjkTPxR3r2SZ3GO0HY0yEKLRXIDKsQ_0fZ-s9IAHBO92AszgTIqItY-_jsuIXqM_8i_6k8ohZcZ6acqpaх-g6e8o
Cache-Control: no-cache
```

В ответе будут отображены только те данные, которые определены в scope, на который получен маркер доступа. Пример ответа:

```
{
  "name": "Тестов Тест",
  "sn": "Тестов"
}
```

#### 2.2.3.2. Данные о маркере доступа

Для запроса данных об имеющемся маркере доступа необходимо выполнить запрос методом POST по адресу сервиса интроспекции маркера доступа<sup>10</sup>. 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).

В теле запроса могут быть указаны параметры:

- token – маркер доступа, данные о котором требуется просмотреть (обязательный параметр);

<sup>10</sup> Создан в соответствии с RFC 7662: <https://tools.ietf.org/html/rfc7662>

- `token_type_hint` – тип маркера доступа (например, `access_token`), предназначен для ускорения поиска (опциональный параметр).

Пример запроса:

```
POST /blitz/oauth/introspect HTTP/1.1
Host: <hostname>
Authorization: Basic bG9jYWxob3N0L2RlbW8zOjA5ODc2NTQzMjE
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache

token=MkvRff63ASFvC0-w6D3fiDxQgaYqmkrnzeDDcz374NDIbux6DA8ByDdB9-
oMfalklyoaNfrk87a4Ep3XzSxbF9m0wA2Z7LIUOaCGuSaKDN0
```

В ответе будут переданы следующие данные о маркере доступа:

- `active` – признак действительности маркера доступа, принимает значения `true/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 (`scope`);
- `client_id` – идентификатор системы-клиента, которая получила данный маркер доступа;
- `username` – 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:

```
{
  "username": "testuser",
  "scope": "test",
  "jti": "10jdlNohfHzuv3xoFurvWSPheEJEC7KHdHr-
dcaVyYYvV3h0l2sh6OVVE4z3ChnRNVbdddjpn3KH4Z76nQsO3q5ca8LbG9KWtKo1xWJKSbM0",
  "token_type": "Bearer",
  "client_id": "localhost/demo2",
  "active": true
}
```

### 2.2.3.3. Идентификационная информация

Для получения данных об идентификации и аутентификации приложению необходимо самостоятельно анализировать содержание маркера идентификации. Маркер 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 описание алгоритма шифрования (параметр “alg”); в настоящее время в Blitz Identity Provider поддерживается алгоритм электронной подписи RSA SHA-256, рекомендуемый спецификацией (соответствует значению “RS256”).

Набор утверждений включает следующие атрибуты:

- идентификатор маркера (“nonce”), передается в неизменном виде из соответствующего запроса на проведение аутентификации;
- время прекращения действия (“exp”), указывается в секундах с 1 января 1970 г. 00:00:00 GMT;
- время выдачи (“iat”), указывается в секундах с 1 января 1970 г. 00:00:00 GMT;
- идентификатор субъекта (“sub”), в качестве значения указывается значение идентификатора, определенного в консоли управления, для данного пользователя;
- адресат маркера (“aud”), указывается client\_id приложения, направившего запрос на аутентификацию;
- организация, выпустившая маркер (“iss”), указывается URL Blitz Identity Provider.

Пример набора утверждений:

```
{
  "nonce": "987654321",
  "exp": 1445004777,
  "iat": 1444994212,
  "sub": "test@dev-blitz-idp.loc",
  "aud": [
    "localhost/demo2"
  ],
  "iss": "https://<hostname>"
}
```

Подпись (signature) маркера осуществляется по алгоритму, который указывается в параметре “alg” маркера. Подпись вычисляется от двух предыдущих частей маркера (HEADER.PAYLOAD).

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

1. Проверка идентификатора Blitz Identity Provider, содержащегося в маркере идентификации.
2. Сверка nonce исходного запроса на проведение аутентификации и nonce в маркере идентификации.
3. Проверка идентификатора приложения, т.е. именно приложение должно быть указано в качестве адресата маркера идентификации.
4. Проверка подписи маркера идентификации (с использованием указанного в маркере

алгоритма).

5. Текущее время должно быть не позднее, чем время прекращения срока действия маркера идентификации.

После валидации маркера идентификации приложение считает пользователя аутентифицированным. Для получения дополнительных данных о пользователе следует использовать маркер доступа.

Для анализа содержания маркера идентификации, а также для упрощения разработки модулей по его проверке можно воспользоваться доступными онлайн-декодерами и библиотеками<sup>11</sup>.

#### 2.2.4. Обеспечение логута при использовании OAuth 2.0

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 логута. 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 в Blitz Identity Provider приложение после закрытия своей локальной сессии должно осуществить HTTP Redirect на следующий URL в Blitz Identity Provider, передав в качестве query-параметра адрес возврата в приложение: `http(s)://<hostname>:<port>/blitz/login/logout?redirect=http(s)://redirect_url`. Если Blitz Identity Provider успешно завершит логат, то он перенаправит пользователя по переданному URL обратно в приложение.

---

<sup>11</sup> См., например: <http://jwt.io/>

### 3. Рекомендации по дизайну кнопок и ссылок в приложении, нажатие которых инициирует идентификацию и аутентификацию

В целях унификации внешнего вида ссылок и кнопок приложений, по нажатию которых инициируется обращение к Blitz Identity Provider, рекомендуется при оформлении ссылок и кнопок использовать рядом с ними изображение иконки замка, присутствующего на стандартной странице входа Blitz Identity Provider.

Примеры оформления ссылок и кнопок приведены на рисунке 10.

Набор иконок разных размеров (формат PNG) и образцы кнопок (формат PSD) можно загрузить по ссылкам <http://identityblitz.ru/login-icons-examples> и <http://identityblitz.ru/login-buttons-examples>.



Рисунок 10 – Образцы оформления ссылок и кнопок на авторизацию

## 4. Вызов сервиса регистрации и изменения данных пользователя

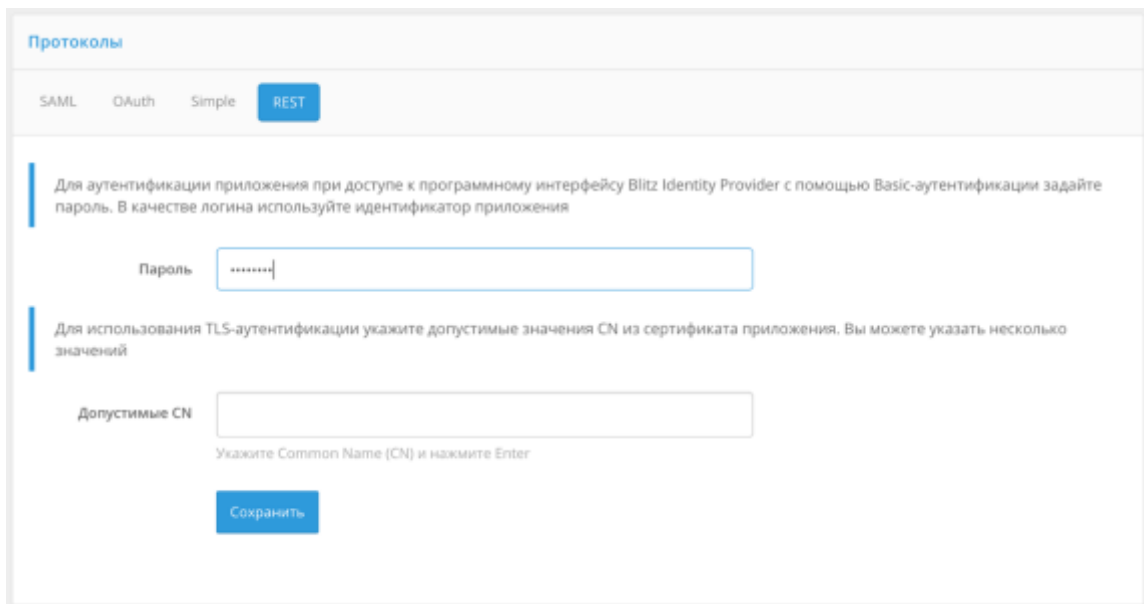
### 4.1. Регистрация системы-клиента

Для вызова сервиса регистрации и изменения данных предварительно необходимо настроить приложение, которое будет выступать в качестве системы-клиента REST-сервисов Blitz Identity Provider. Для этого нужно зарегистрировать новое приложение в разделе *Приложения* согласно стандартной схеме (см. *рис. 1*):

- нажать «Добавить приложение»;
- указать идентификатор, название и домен системы;
- нажать «Сохранить».

Далее перейти к настройкам приложения, в качестве протокола подключения указать REST и заполнить следующие данные (см. *рис. 11*):

- пароль – будет использоваться при HTTP Basic-аутентификация, в качестве логина – идентификатор системы-клиента; если параметр не задан, то HTTP Basic-аутентификация не будет возможна для данной системы-клиента;
- допустимые CN – перечень значений атрибута CN сертификата, используемого при TLS-аутентификация; если не заданы параметры, то TLS-аутентификация не будет возможна для данной системы-клиента.



The screenshot shows a web interface titled "Протоколы" (Protocols). It has four tabs: "SAML", "OAuth", "Simple", and "REST". The "REST" tab is selected. Below the tabs, there is a blue vertical bar on the left. To the right of the bar, there is a text box for "Пароль" (Password) with a masked input field. Below that, there is another text box for "Допустимые CN" (Allowed CN). Below the "Допустимые CN" field, there is a small instruction: "Укажите Common Name (CN) и нажмите Enter". At the bottom of the form, there is a blue button labeled "Сохранить" (Save).

Рисунок 11 – Настройка приложения с использованием консоли (протокол REST)

### 4.2. Вызов сервиса регистрации нового пользователя

Инициирование регистрации в Blitz Identity Provider нового пользователя осуществляется методом PUT по адресу `/blitz/rest/usr/reg`. Запрос включает в себя заголовок (header) для проведения HTTP Basic авторизации, в качестве query-параметра может быть указан `activate` – системный атрибут, принимающий следующие значения:



- true – необходимо автоматически создавать учетную запись пользователя активированной, т.е. после вызова сервиса регистрации пользователь может сразу входить с помощью своего логина и пароля;
- false – вход в учетную запись требуется только после прохождения процедуры активации, в ходе которой пользователь подтверждает владение своим контактом.

Если параметр `activate` не указан, то учетная запись создается не активированной, т.е. она требует активации.

В теле запроса указывается `json`, имеющий вид `{"attr": value}`, где:

- `attr` – название атрибута. Оно должно соответствовать либо атрибуту LDAP-хранилища, либо дополнительному атрибуту;
- `value` – значение атрибута.

Если в числе передаваемых атрибутов отсутствует пароль (атрибут `password`), то возможно следующее поведение системы:

- пароль генерируется автоматически и отсылается пользователю на указанный адрес электронной почты (для случая, когда параметр `activate` принимает значение `true`);
- пароль задается пользователем после прохождения по ссылке, отправленной пользователю на адрес электронной почты (для случая, когда параметр `activate` принимает значение `false`);

Пример запроса:

```
PUT /blitz/rest/usr/reg?activate=true HTTP/1.1
Host: demo.identityblitz.ru
Authorization: Basic bWFudWFsLXRlc3QtY2xpZW50OjEyMzQ1Njc4OTA=
Content-Type: application/json
Cache-Control: no-cache

{
  "givenName": "Maxim",
  "telephoneNumber": "+7(000)1234567",
  "cn": "Maxim Filimonov",
  "sn": "Filimonov",
  "mail": "mfilimonov@bip-test.ru",
  "password": "4354545565"
}
```

Если запрос выполнен успешно, то сервис вернет статус 200 ОК и уникальный идентификатор (`sub`) сохраненного пользователя.

При наличии ошибки возможны следующие ответы сервиса:

- статус 401 (Unauthorized) – система-клиент не зарегистрирована или неверно произведена его авторизация;
- статус 400 (Bad request) – отправлены некорректные данные;
- статус 500 (Internal Server Error) – возникла внутренняя ошибка сервера.

В тех случаях, когда это возможно, сервис также возвращает json, содержащий тип и описание ошибки.

#### 4.3. Вызов сервиса изменения данных пользователя

Запрос на добавление/изменение атрибута осуществляется методом POST по адресу `/blitz/rest/usr/{sub}/alt`, где `sub` – имя (идентификатор пользователя). Запрос включает в себя заголовок (header) для проведения HTTP Basic авторизации, а в теле запроса json, содержащий два вложенных объекта:

- `updated` – перечень атрибутов, которые должны быть изменены или добавлены;
- `deleted` – перечень атрибутов, которые должны быть удалены.

Пример запроса:

```
POST /blitz/rest/usr/BIP8745897438975/alt HTTP/1.1
Host: demo.identityblitz.ru
Authorization: Basic bWFudWFsLXRlc3QtY2xpZW50OjEyMzQ1Njc4OTA=
Content-Type: application/json
Cache-Control: no-cache

{"updated":
  {
    "givenName": "Maxim",
    "telephoneNumber": "+7(000)1234567",
    "cn": "Maxim Filimonov"
  },
"deleted": ["email", "middlename"]
}
```

В качестве удаляемых атрибутов не могут быть указаны обязательные (например, уникальный идентификатор) и системные атрибуты. Также в качестве изменяемых атрибутов не могут быть указаны системные атрибуты.

Если атрибут не указан в запросе, то он не изменяется.

Если запрос выполнен успешно, то сервис вернет статус 200 ОК.

При наличии ошибки возможны следующие ответы сервиса:

- статус 401 (Unauthorized) – система-клиент не зарегистрирована или неверно произведена его авторизация;
- статус 400 (Bad request) – отправлены некорректные данные;
- статус 404 (Not found) – пользователь с указанным в запросе идентификатором не найден;
- статус 500 (Internal Server Error) – возникла внутренняя ошибка сервера.

В тех случаях, когда это возможно, сервис также возвращает json, содержащий тип и описание ошибки.

#### 4.4. Привязка НОТР/ТОТР-генератора к учетной записи пользователя

С использованием API приложение может назначить пользователю НОТР/ТОТР аппаратный ключ для осуществления двухфакторной аутентификации.

Запрос на привязку HOTP/TOTP-генератора кодов к учетной записи пользователя осуществляется методом PUT по адресу `/blitz/rest/usr/{username}/attach/hotp`, где `username` – имя (идентификатор пользователя). Запрос включает в себя заголовок (header) для проведения HTTP Basic авторизации, а в теле запроса json, имеющий вид:

```
{
  "serial" : "AN064433",
  "otp1" : "200299",
  "otp2" : "136915",
  "otp3" : "024689"
}
```

где:

- `serial` – значение серийного номера HOTP-генератора;
- `otp1`, `otp2`, `otp3` – значение последовательности трех разовых кодов HOTP-устройства.

Пример запроса:

```
PUT: /blitz/rest/usr/ivanov/attach/hotp
Host: demo.identityblitz.ru
Authorization: Basic bWFudWFsLXRlc3QtY2xpZW50OjEyMzQ1Njc4OTA=
Content-Type: application/json
Cache-Control: no-cache
{
  "serial" : "AN064433",
  "otp1" : "200299",
  "otp2" : "136915",
  "otp3" : "024689"
}
```

Должна осуществляться только привязка тех HOTP/TOTP-генераторов, по которым в Blitz Identity Provider предварительно администратор выполнил загрузку файла с описанием устройств, полученный от производителя устройств (см. Руководство администратора Blitz Identity Provider).