

Повышение уровня авторизации

Оглавление

- # Термины и определения
 - # Редирект на виджет повышения уровня авторизации
 - # Формат запроса на повышение уровня авторизации
 - # Формат успешного ответа
 - # Поддерживаемые типы ошибок
 - # Формат неуспешного ответа
 - # Формат ответа в случае отмены сценария пользователем
 - # Встраивание виджета повышения уровня авторизации inline
 - # Список требований к защищаемому сервису
 - # Рендеринг виджета
 - # Отправка и валидация SMS-кода
 - # Успешное повышение уровня авторизации
 - # Неуспешное повышение уровня авторизации
 - # Отложенный рендеринг
 - # Схема взаимодействия
 - # Получение токена доступа повышенного уровня авторизации
 - # Валидация токена доступа повышенного уровня авторизации
-

Термины и определения

- **RooX UIDM** — Сервер аутентификации и авторизации, это централизованный сервис по управлению учетными записями и доступом к ресурсам, который включает в себя:
 - Identity Provider — единую точку идентификации/аутентификации, управления учетными записями и правилами доступа;
 - Access Management — разграничение доступа к ресурсам;
 - Single Sign-On (SSO) — обеспечение единого доступа, когда пользователю достаточно аутентифицироваться один раз для доступа к группе сервисов;
 - Federation — обеспечение федеративной аутентификации между несколькими доменами систем
- **авторизация** — процесс проверки полномочий пользователя по отношению к запрашиваемым сервисам
- **уровень авторизации** — числовое значение, определяющее множество операций, доступных внешнему сервису для выполнения от лица пользователя. Операция внешнего сервиса может иметь минимальный требуемый уровень авторизации и выполняться только если текущий уровень авторизации пользователя больше или равен минимальной.

- **OAuth 2.0 code** — строковый идентификатор, выдаваемый пользователю для последующего обмена на access_token
- **access_token** — строковый идентификатор, предоставляющий внешнему сервису выполнять действия от лица пользователя или определяет список идентификаторов защищенных ресурсов, которые могут быть использованы сервисом для данного пользователя
- **Widget Rendering System (WRS)** — система, выполняющая функции рендеринга виджетов. У нее можно запросить отрендерить (отобразить в виде пригодном для браузера) тот или иной виджет.
- **Виджет** — небольшой независимый программный модуль, работающий в некоторой среде (напр. сайте, браузере, мобильном телефоне) и исполняющий, как правило, одну определённую функцию. В данном документе речь идет о виджете повышения уровня авторизации.

Редирект на виджет повышения уровня авторизации

Если пользователь хочет произвести некоторое действие на защищаемом ресурсе, серверное приложение защищаемого ресурса должно выполнить запрос на валидацию access_token. Если для текущего access_token уровень авторизации оказывается меньше, чем необходимо для выполнения запрошенного действия, то пользователю предоставляется возможность пройти процесс повышения уровня авторизации.

Повышение уровня авторизации методом редиректа происходит через OAuth 2.0 с передачей запрашиваемого уровня авторизации в качестве параметра auth_level. Если уровень авторизации пользователя меньше запрашиваемого, он будет перенаправлен на виджет повышения уровня по SMS-коду.

Если SMS-код был успешно подтвержден, будет возвращен OAuth 2.0 code, по которому можно получить access_token с повышенным уровнем авторизации auth_level

ВАЖНО

Ранее выданные access_token также продолжают действовать.

Рекомендуется использовать access_token с повышенным уровнем авторизации только для выполнения действий, требующих такой уровень. Если для выполнения действия достаточно стандартного уровня авторизации, рекомендуется использовать access_token полученный при первоначальной авторизации по OAuth 2.0.

Формат запроса на повышение уровня авторизации

```
GET <sso_host>/sso/oauth2/authorize?
  realm=%2Fcustomer&
  response_type=code&
  client_id=<client_id>&
  redirect_uri=<redirect_uri>&
  auth_level=<value>&
  success_link_uri=<success_link_uri>&
  cancel_link_uri=<cancel_link_uri>&
  error_link_uri=<error_link_uri>&
  method=otp_sms
```

Параметры

- `<sso_host>` — базовый адрес сервера RooX UIDM, например sso.rooxteam.com, может содержать порт, например sso.rooxteam.com:8080
- `realm` — группа пользователей RooX UIDM, всегда используется значение %2Fcustomer, которое является uri-encoded значением /customer

- `response_type` — способ авторизации пользователя, всегда используется значение `code`
- `client_id` — идентификатор клиента, например `selfcare`, возможные значения зависят от конфигурации
- `redirect_uri` — URL обратного редиректа, URL-encoded, например `https%3A%2F%2Fsso.rooxteam.com%2F%2Fsso%2Fsecure%2F%2Flk.jsp` для адреса `https://sso.rooxteam.com/sso/secure/lk.jsp`. RooX UIDM производит редирект браузера пользователя после прохождения успешного или неуспешного повышения уровня авторизации. Адрес должен быть в списке разрешенных в RooX UIDM, список разрешенных адресов конфигурируется администратором RooX UIDM.
- `success_link_uri` — опциональный параметр, целевой URL в случае успешного повышения уровня авторизации, URL-encoded, например `https%3A%2F%2Fsso.rooxteam.com%2F%2Fsso%2Fsecure%2F%2FchangePassword` для адреса `https://sso.rooxteam.com/sso/secure/changePassword`. Адрес должен быть в списке разрешенных в RooX UIDM, список разрешенных адресов конфигурируется администратором RooX UIDM.
- `cancel_link_uri` — опциональный параметр, целевой URL в случае отмены сценария пользователем, URL-encoded, например `https%3A%2F%2Fsso.rooxteam.com%2F%2Fsso%2Fsecure%2F%2Fsettings` для адреса `https://sso.rooxteam.com/sso/secure/settings`. Адрес должен быть в списке разрешенных в RooX UIDM, список разрешенных адресов конфигурируется администратором RooX UIDM.
- `error_link_uri` — опциональный параметр, целевой URL в случае ошибки повышения уровня авторизации, URL-encoded, например `https%3A%2F%2Fsso.rooxteam.com%2F%2Fsso%2Fsecure%2F%2Fforbidden` для адреса `https://sso.rooxteam.com/sso/secure/forbidden`. Адрес должен быть в списке разрешенных в RooX UIDM, список разрешенных адресов конфигурируется администратором RooX UIDM.
- `auth_level` — желаемый уровень авторизации
- `method` — опциональный параметр, который задает способ повышения уровня авторизации, сейчас доступен только способ `otp_sms`. Если параметр опущен, предполагается, что RooX UIDM проведет авторизацию наиболее подходящим способом, достаточным для достижения уровня авторизации, указанного в `auth_level`

Пояснение работы параметров "link_uri".

RooX UIDM в числе параметров запроса на повышение уровня авторизации принимает ряд параметров, которые заканчиваются на `link_uri`. Эти параметры используются для проброса информации о целевом пункте назначения пользователя после возврата из RooX UIDM. Поскольку OAuth2.0 предполагает наличие только одной точки возврата — `redirect_uri`, а целевым пунктом назначения, как правило, является совсем другой URL, то сервису необходимо организовать хранение состояния для организации переадресации пользователя.

Эта задача является типовой для подключаемых к RooX UIDM сервисов, поэтому она решена на уровне API повышения уровня.

В `link_uri` параметрах сервис может передать адреса возврата при различных результатах повышения уровня авторизации. Обязанностью RooX UIDM является проверка этих адресов на вхождение в список разрешенных для сервиса и передача обратно сервису после окончания сценария в одноименных параметрах.

ЗАМЕТКА

Автоматической переадресации на эти адреса не происходит. В обязанности сервиса входит анализ кода возврата из RooX UIDM согласно спецификации OAuth2.0 и дальнейшего перенаправления пользователя.

Формат успешного ответа

В случае, если пользователь успешно выполнил действия, необходимые для повышения уровня авторизации, он будет перенаправлен на `redirect_uri`, указанный в запросе, с новым кодом авторизации в GET-параметре

```
GET <redirect_uri>?code=<auth_code>
```

Параметры

- `<redirect_uri>` — URL обратного редиректа, например `https://sso.rooxteam.com/sso/secure/lk.jsp`
- `<auth_code>` — код авторизации повышенного уровня, например `1d601e9a-992d-44a7-be21-bca07fbc762c`
- `<success_link_uri>` — URL, переданный сервисом в запросе на повышение уровня авторизации в одноименном параметре, URL-encoded, например ``+https%3A%2F%2Fsso.rooxteam.com%2Fsso%2Fsecure%2FchangePassword`` для адреса <https://sso.rooxteam.com/sso/secure/changePassword>.
- `<cancel_link_uri>` — URL, переданный сервисом в запросе на повышение уровня авторизации в одноименном параметре, URL-encoded, например ``+https%3A%2F%2Fsso.rooxteam.com%2Fsso%2Fsecure%2Fsettings`` для адреса <https://sso.rooxteam.com/sso/secure/settings>.
- `<error_link_uri>` — URL, переданный сервисом в запросе на повышение уровня авторизации в одноименном параметре, URL-encoded, например ``+https%3A%2F%2Fsso.rooxteam.com%2Fsso%2Fsecure%2Fforbidden`` для адреса <https://sso.rooxteam.com/sso/secure/forbidden>.

Поддерживаемые типы ошибок

Код ошибки	Описание
invalid_request	Были переданы неправильные параметры
server_error	Внутренняя ошибка сервера
destination_unreachable	Сервер недоступен
redirect_uri_mismatch	Редирект урл неправильный
insufficient_scope	В случае отмены сценария пользователем

Формат неуспешного ответа

Пользователь может превысить число попыток ввода SMS-кода или может не пройти одна из проверок сервиса или пользователя. В этом случае вместо кода авторизации в GET-параметре будет описание ошибки.

```
GET <redirect_uri>?error=<access_denied>&error_description=<error_description>
```

Параметры

- `<redirect_uri>` — URL обратного редиректа, например `https://sso.rooxteam.com/sso/secure/lk.jsp`
- `<error>` — код ошибки согласно спецификации OAuth 2.0 [RFC 6749 пункт 5.2](#), например `access_denied`.
- `<error_description>` — текстовое описание ошибки, например `The%20resource%20owner%20or%20authorization%20server%20denied%20the%20request`.
- `<success_link_uri>` — URL, переданный сервисом в запросе на повышение уровня авторизации в одноименном параметре, URL-encoded, например ``+https%3A%2F%2Fsso.rooxteam.com%2Fsso%2Fsecure%2FchangePassword`` для адреса <https://sso.rooxteam.com/sso/secure/changePassword>.
- `<cancel_link_uri>` — URL, переданный сервисом в запросе на повышение уровня авторизации в одноименном параметре, URL-encoded, например ``+https%3A%2F%2Fsso.rooxteam.com%2Fsso%2Fsecure%2Fsettings`` для

адреса <https://sso.rooxteam.com/sso/secure/settings> .

- `<error_link_uri>` — URL, переданный сервисом в запросе на повышение уровня авторизации в одноименном параметре, URL-encoded, например ``+https%3A%2F%2Fsso.rooxteam.com%2Fsso%2Fsecure%2Fforbidden`` для адреса <https://sso.rooxteam.com/sso/secure/forbidden> .

Формат ответа в случае отмены сценария пользователем

В случае, если пользователь отменил сценарий повышения уровня, он будет перенаправлен на `redirect_uri`, указанный в запросе, с параметром `error = unsufficient_scope`

```
GET <redirect_uri>?  
error=insufficient_scope&error_description=The%20request%20requires%20higher%20privileges%  
20than%20provided%20by%20the%20access%20token.
```

Параметры

- `<redirect_uri>` — URL обратного редиректа, например <https://sso.rooxteam.com/sso/secure/ik.jsp>
- `<error>` — код ошибки, всегда `insufficient_scope` в данном случае
- `<error_description>` — текстовое описание ошибки, например `The%20request%20requires%20higher%20privileges%20than%20provided%20by%20the%20access%20token.`
- `<success_link_uri>` — URL, переданный сервисом в запросе на повышение уровня авторизации в одноименном параметре, URL-encoded, например ``+https%3A%2F%2Fsso.rooxteam.com%2Fsso%2Fsecure%2FchangePassword`` для адреса <https://sso.rooxteam.com/sso/secure/changePassword> .
- `<cancel_link_uri>` — URL, переданный сервисом в запросе на повышение уровня авторизации в одноименном параметре, URL-encoded, например ``+https%3A%2F%2Fsso.rooxteam.com%2Fsso%2Fsecure%2Fsettings`` для адреса <https://sso.rooxteam.com/sso/secure/settings> .
- `<error_link_uri>` — URL, переданный сервисом в запросе на повышение уровня авторизации в одноименном параметре, URL-encoded, например ``+https%3A%2F%2Fsso.rooxteam.com%2Fsso%2Fsecure%2Fforbidden`` для адреса <https://sso.rooxteam.com/sso/secure/forbidden> .

Встраивание виджета повышения уровня авторизации inline

Если пользователь хочет произвести определенное действие на защищаемом ресурсе, серверное приложение защищаемого ресурса должно выполнить запрос на валидацию `access_token`. Если для текущего `access_token` уровень авторизации оказывается меньше, чем необходимо для выполнения запрошенного действия, то пользователю предоставляется возможность пройти процесс повышения уровня авторизации.

Защищаемый ресурс может встроить в свою страницу виджет повышения уровня авторизации. Для этого серверное приложение защищаемого ресурса должно выполнить запрос к RooX UIDM, полученный в ответе HTML с виджетом должен быть непосредственно внедрен в код страницы защищаемого ресурса.

Запрос на повышение уровня авторизации может быть выполнен только при наличии валидного `access_token`, полученного от RooX UIDM в результате авторизации по OAuth 2.0.

Встроенный в страницу защищаемого ресурса виджет будет самостоятельно делать запросы к RooX UIDM для отправки SMS-кода, валидации кода, введенного пользователем, обрабатывать лимиты отправки кодов и попыток ввода.

При успешном завершении процесса повышения уровня авторизации виджет публикует событие, содержащее код авторизации, который может быть обменян в RooX UIDM на токен повышенного уровня авторизации. Защищаемый ресурс должен подписаться на это событие, передать код авторизации на серверное приложение защищаемого ресурса, после чего серверным вызовом RooX UIDM код должен быть обменян на `access_token` с повышенным

уровнем авторизации.

Рекомендуется использовать `access_token` с повышенным уровнем авторизации только для выполнения действий, требующих такой уровень. Если для выполнения действия достаточно стандартного уровня авторизации, рекомендуется использовать `access_token` полученный при авторизации по OAuth 2.0.

При неуспешном завершении процесса повышения уровня авторизации или при возникновении ошибок в работе самой системы (например, недоступность сервера) виджет публикует событие с описанием ошибки.

Список требований к защищаемому сервису

Интеграция по OAuth 2.0

Защищаемый сервис должен быть интегрирован с RooX UIDM по протоколу OAuth 2.0.

Поддержка Server-Side Include

Защищаемый сервис должен реализовать на стороне сервера получение виджета по HTTP-запросу на RooX UIDM и встраивание полученного виджета в пользовательскую web-страницу.

Поддержка javascript callbacks

Защищаемый сервис должен поддерживать в клиентском javascript-коде обработчики успешного и неуспешного завершения операции повышения уровня авторизации. Обработчик успешного завершения должен принимать OAuth 2.0 code и отправлять его на сервер защищаемого ресурса для замены на токен. Обработчики должны быть зарегистрированы подпиской на события виджета повышения уровня, с помощью вызова javascript-методов встроеного виджета.

Поддержка endpoint успешного повышения уровня авторизации

Сервер защищаемого сервиса должен поддерживать конечную точку (endpoint), принимающую OAuth 2.0 code. Сервер должен обменять OAuth 2.0 code на `access_token` аналогично сценарию получения токена доступа в OAuth 2.0. Полученный `access_token` будет иметь повышенный уровень авторизации. Серверу защищаемого сервиса необходимо сохранить токен с повышенным уровнем авторизации в сессии. При совершении действий требующих повышенного уровня авторизации нужно передавать в RooX UIDM для валидации именно токен повышенного уровня авторизации. Результаты авторизации с токеном повышенного уровня запрещается кешировать.

ВАЖНО

Владелец подключаемого сервиса должен заполнить пункт AUTHN.17 в документе "Анкета, которую должны заполнить владельцы сервисов, интегрируемых с RooX UIDM"

Рендеринг виджета

Пример запроса

```
POST /sso/widgetApi/v1/widgetZone
Host: <sso_host>
Content-Type: application/json
Accept: application/json
```

```
{
  "service": "increase-auth-level",
  "access_token": "7bdaeacc-3d80-415c-920f-a7c30ca5e743",
  "required_auth_level": "<value>",
  "wrs": {
    "mid": "0",
    "country": "RU",
    "lang": "ru",
    "uh_X-Forwarded-For": "10.2.3.4"
  }
}
```

Параметры

- `<sso_host>` — базовый адрес сервера RooX UIDM, например `sso.rooxteam.com`
- `service` — название цепочки RooX UIDM, для виджета повышения уровня авторизации нужно передать значение `increase-auth-level`
- `access_token` — токен доступа с обычным уровнем авторизации, полученный ранее от RooX UIDM в результате авторизации по протоколу OAuth 2.0
- `required_auth_level` — желаемый уровень авторизации
- `wrs` — дополнительный набор параметров для рендеринга виджета

Параметры WRS

`mid`

- Тип параметра — `int`
- Описание — Идентификатор виджета в рамках страницы-контейнера. Целое число от 0, указывающее на порядковый номер виджета.
- Обязательный — нет
- Значение по умолчанию — 0

`country`

- Тип параметра — `string`
- Описание — 2-буквенный ISO код страны для `i18n`, например, `RU`.
- Обязательный — нет
- Значение по умолчанию — Берется из настроек сервера виджетов или из настроек браузера.

`lang`

- Тип параметра — `string`
- Описание — 2-буквенный ISO код языка для `i18n`, например, `ru`.
- Обязательный — нет
- Значение по умолчанию — Берется из настроек сервера виджетов или из настроек браузера.

`up_<USER_PROPERTY>`

- Тип параметра — `string`
- Описание — Устанавливает пользовательскую переменную `<USER_PROPERTY>`. Пользовательские переменные передаются в код виджета, и могут быть обработаны каким-то специфичным для виджета образом.

- Обязательный — нет
- Значение по умолчанию — Отсутствует

nocache

- Тип параметра — int
- Описание — Если установлен в 1, заставляет контейнер перечитывать дескриптор виджета при каждом запросе. Так же добавляются заголовки выключающие кэширование в браузере. Удобно включать во время разработки и отладки.
- Обязательный — нет
- Значение по умолчанию — 0

refresh

- Тип параметра — int
- Описание — Время жизни для браузерного кэша в секундах.
- Обязательный — нет
- Значение по умолчанию — 300

delayed

- Тип параметра — boolean
- Описание — Заставляет отрисовывать виджет в отложенном режиме. Имеет смысл использовать для inline. Выключаются runOnLoadHandlers и обертка становится скрытой, виджет можно запустить вызовом `com.rooxteam.renderWidget(mid, moveToElement)`
- Обязательный — нет
- Значение по умолчанию — false

uh_<USER_HEADER>

- Тип параметра — string
- Описание — Передает заголовок <USER_HEADER>. Пользовательские заголовки передаются в код виджетной платформы, как будто они пришли напрямую из пользовательского HTTP(S) запроса, и могут быть обработаны каким-то специфичным образом, например для аутентификации.
- Обязательный — нет
- Значение по умолчанию — Отсутствует

uh_X-Forwarded-For

- Тип параметра — string
- Описание — Исходный IP-адрес пользователя
- Обязательный — нет
- Значение по умолчанию — Отсутствует

features_loaded

- Тип параметра — string
- Описание — Список фичей, которые wgs будет считать уже загруженными и не будет выдавать js-код для них. Это также касается фичей, от которых зависят фичи из переданного списка (транзитивно). Формат — список, разделенный двоеточиями: `feature1:feature2:feature3`

- Обязательный — нет
- Значение по умолчанию — Отсутствует

Пример успешного ответа

```
HTTP 200 OK
```

```
{
  "content": "<span id=widget_wrapper_0><link ....><script>.....</script></span>",
  "type": "HTML"
}
```

Параметры

- content — содержимое виджета, которое должно быть встроено в страницу защищенного ресурса
- type — формат содержимого, сейчас всегда возвращается HTML

Примеры неуспешного ответа

Отсутствуют обязательные параметры

```
HTTP 400 Bad Request
```

```
{
  "error": {
    "code": 400,
    "message": "Missing required parameter 'service'"
  }
}
```

Невалидный access token

```
HTTP 401 Unauthorized
```

```
{
  "error": {
    "code": 401,
    "message": "Invalid access token"
  }
}
```

```
}
```

В случае получения этой ошибки защищаемый ресурс должен сам выполнить необходимые действия, например перевести пользователя в виджет аутентификации.

OAuth клиент, которому принадлежит access token, заблокирован

HTTP 403 Forbidden

```
{
  "error": {
    "code": 403,
    "message": "OAuth client is blocked"
  }
}
```

В случае получения этой ошибки защищаемый ресурс должен сам выполнить необходимые действия, например вывести сообщение о том, что клиент заблокирован.

OAuth клиенту, которому принадлежит access token, запрещено повышение уровня авторизации

HTTP 403 Forbidden

```
{
  "error": {
    "code": 403,
    "message": "Operation is not allowed for client"
  }
}
```

В случае получения этой ошибки защищаемый ресурс должен сам выполнить необходимые действия, например вывести сообщение о том, что для клиента запрещено повышение уровня авторизации.

Параметры неуспешного ответа

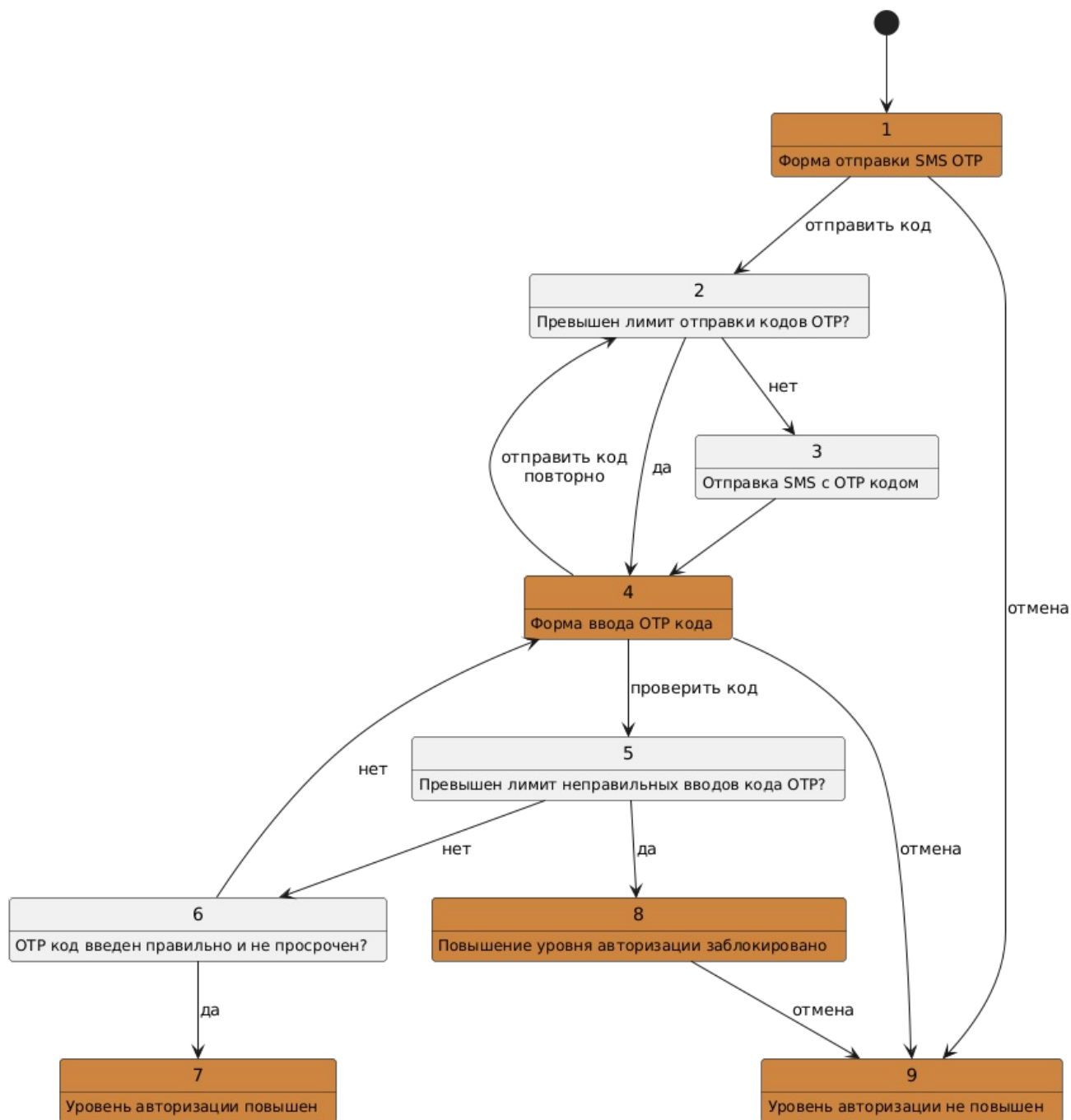
- code — код ответа, повторяет HTTP статус ответа
- message — краткое описание ошибки

Отправка и валидация SMS-кода

Встроенный в страницу защищаемого ресурса виджет будет самостоятельно делать запросы к RooX UIDM для отправки SMS-кода, валидации кода, введенного пользователем, обрабатывать лимиты отправки кодов и попыток

ввода.

Алгоритм отправки и валидации SMS-кодов представлен на схеме:



Успешное повышение уровня авторизации

Если SMS-код был успешно подтвержден, будет сгенерирован OAuth 2.0 code для повышенного уровня авторизации. code будет передан виджету, а виджет опубликует событие `com.rooxteam.otp.success` и передаст OAuth 2.0 code в него.

Защищаемый ресурс должен подключить на страницу библиотеку для работы с событиями и подписаться на событие, добавив код следующего вида:

```
com.rooxteam.pubsub.on(
  "com.rooxteam.otp.success",
  function(event) {
    //send oauthCode on server
    handleOAuthCode(event.oauth_code);
  });
```

Сервер защищаемого ресурса должен, используя OAuth 2.0 code, выполнить к RooX UIDM запрос на получение access_token. Если был передан валидный OAuth 2.0 code, RooX UIDM вернет новый короткоживущий access_token с повышенным уровнем авторизации.

ВАЖНО

Ранее выданные access_token также продолжают действовать.

Рекомендуется использовать access_token с повышенным уровнем авторизации только для выполнения действий, требующих такой уровень. Если для выполнения действия достаточно стандартного уровня авторизации, рекомендуется использовать access_token полученный при авторизации по OAuth 2.0.

ЗАМЕТКА

Если при старте сценария повышения уровня у пользователя уже имеется достаточный уровень, то сразу после запуска виджета будет отправлено событие com.rooxteam.otp.success.

Неуспешное повышение уровня авторизации

После истечения количества попыток ввода SMS-кода, либо в случае ошибки работы системы, виджет публикует событие com.rooxteam.otp.error и передает в него описание ошибки. Защищаемый ресурс может подписаться на это событие:

```
com.rooxteam.pubsub.on(
  "com.rooxteam.otp.error",
  function(error) {
    if (error.error == 'invalid_grant') {
      //token error
    } else if (error.error == 'canceled') {
      //canceled by user
    } else if (error.error == 'server_error') {
      //server error
    }
  });
```

Формат объекта error представлен ниже. Объект может иметь дополнительные поля для некоторых видов ошибок, описанных ниже.

```
{
  "error": "<error>",
  "error_description": "<error_description>"
}
```

ВАЖНО

Несмотря на то, что виджет отрисовывает страницы ошибок пользователю, соответствующие события все равно будут отправляться интегрирующей его странице.

Параметры ошибки

- error — тип ошибки
- error_description — текстовое описание ошибки

Поддерживаемые типы ошибок

Код ошибки	Описание
invalid_grant	Был предоставлен истекший, отозванный или неверный токен
invalid_request	Внутренняя ошибка виджета
canceled	Пользователь отменил повышение уровня авторизации
access_denied	Пользователю запрещено повышение
server_error	Внутренняя ошибка сервера
destination_unreachable	Сервер недоступен

Ошибка доступа access_denied

Для ошибки с этим типом в полях error_subtype и expires_in предоставляется дополнительная информация:

```
{
  "error": "<error>",
  "error_subtype": "<error_subtype>",
  "error_description": "<error_description>",
  "expires_in": "<expiration_period>"
}
```

Отсутствие поля error_subtype следует трактовать как ошибку при работе с токеном. Поле expires_in присутствует, если error_subtype отражает факт блокировки (например, user_blocked). Отсутствие поля при блокировке означает бессрочную блокировку.

Параметры ошибки

- error — тип ошибки
- error_subtype — подтип ошибки
- error_description — текстовое описание ошибки
- expiration_period — время до истечения блокировки (в секундах)

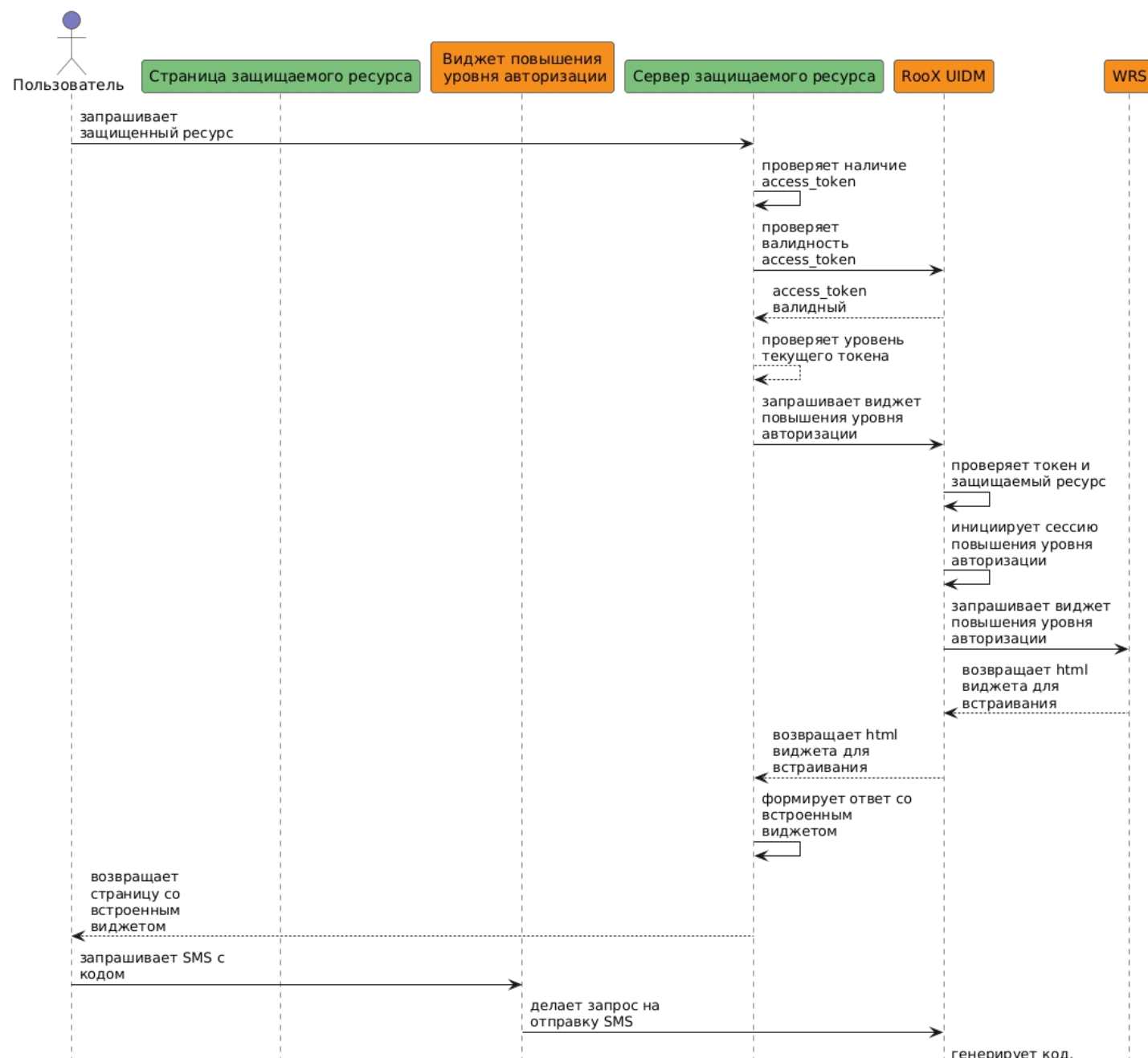
Возможные подтипы ошибки

Подтип	Описание
revoked	Пользователь аннулировал токен

Пользователь заблокирован

OAuth клиент заблокирован

Если защищаемый ресурс хочет отображать виджет повышения уровня авторизации не сразу при загрузке страницы, а при совершении пользователем каких-либо действий на странице, можно использовать функцию отложенного рендеринга. Для этого нужно передать в серверном запросе на рендеринг виджета WRS-параметры `delayed=true` и `mid=0`. Чтобы отрисовать такой виджет нужно сделать явный вызов функции, передав в нее DOM-элемент для встраивания.



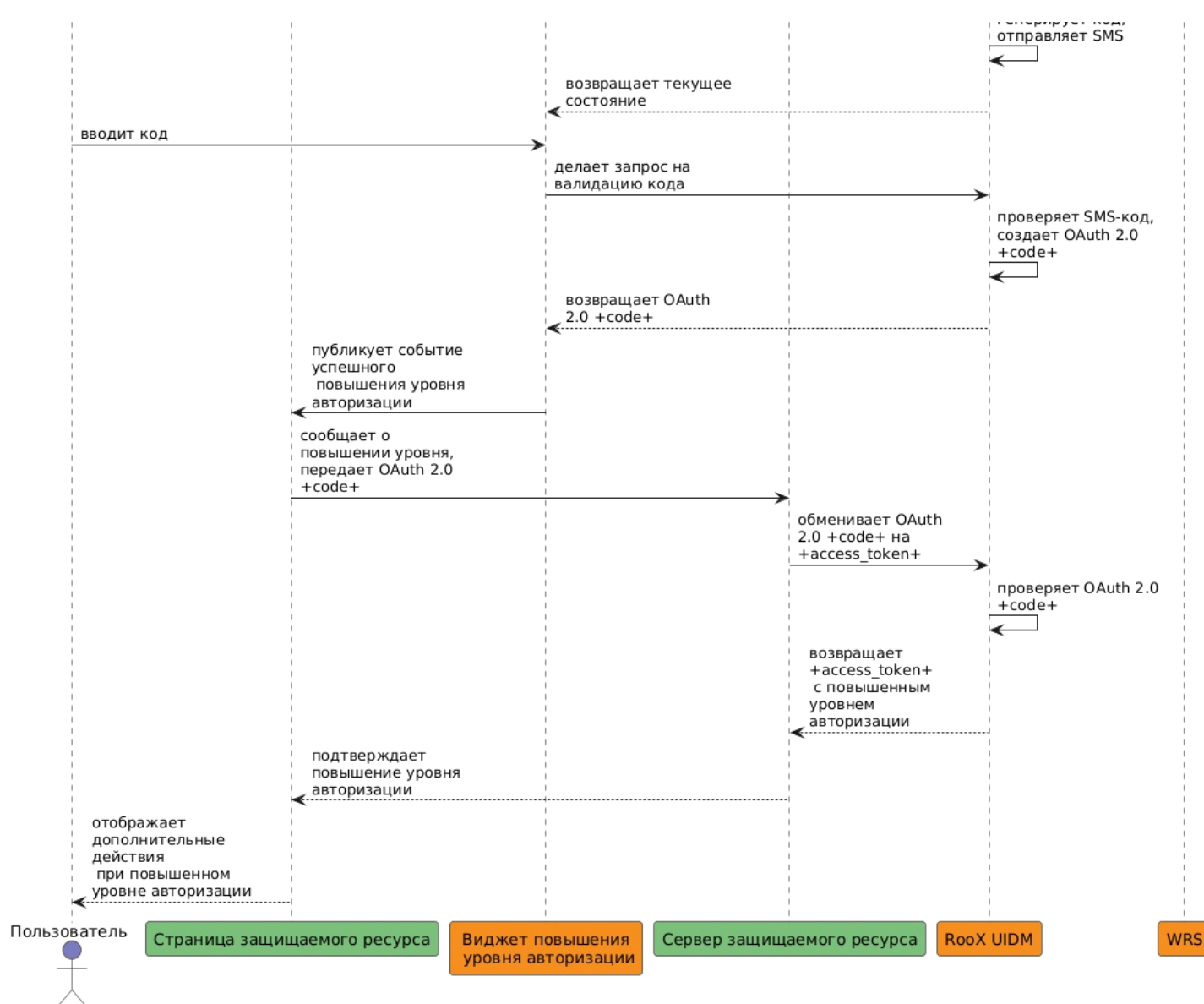


Рисунок 1. Повышение уровня авторизации встраиванием виджета inline

Получение токена доступа повышенного уровня авторизации

Независимо от способа интеграции — с помощью редиректа на виджет или встраивания виджета inline, при успешном повышении уровня авторизации будет возвращен OAuth код. Этот код можно обменять на токен доступа в соответствии со сценарием "Получение токена доступа" документа [Интеграция с RooX UIDM по OAuth 2.0](#).

Валидация токена доступа повышенного уровня авторизации

Валидация токена доступа повышенного уровня авторизации производится в соответствии со сценарием "Валидация токена доступа" документа [Интеграция с RooX UIDM по OAuth 2.0](#). При использовании токена доступа с повышенным уровнем авторизации успешный ответ будет содержать `auth_level`, соответствующий запрашиваемому изначально уровню.

