Спецификация M2M API

Оглавление

Версия продукта: UIDM-{project-version}

1. История изменений

Версия документа	Дата изменения	Комментарий
v0.1.24	2016-05-31	Создание документа.
v0.1.62	2017-05-17	Добавлена аутентификация по пин-коду
v0.1.63	2018-07-04	Добавлена информация по протоколу SRP
v0.1.64	2018-07-25	Добавлены типы и обязательность полей

2. Связанные документы

Название документа	Описание
rfc6749.pdf	Базовая спецификация протокола OAuth2.0.
rfc7009.pdf	Расширение базовой спецификации протокола OAuth2.0, описывающее token revocation.

3. Термины и определения

- **WebSSO** - 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.
- **аутентификация** - проверка принадлежности субъекту доступа предъявленного им идентификатора, подтверждение подлинности.
- **авторизация** - процесс проверки полномочий пользователя по отношению к запрашиваемым сервисам.
- **уровень авторизации** - числовое значение, определяющее множество операций, доступных защищаемому сервису для выполнения от лица пользователя. Операция защищаемого сервиса может иметь минимальный требуемый уровень авторизации и выполняться только если текущий уровень авторизации пользователя больше или равен минимальной. Подробнее об уровнях авторизации см. документ "SSO Auth Model".
- **токен** - access token.
- **access token** - строковый идентификатор, предоставляющий защищаемому сервису выполнять действия от лица пользователя.
- **идентификатор клиента** - строка, идентифицирующая защищаемый сервис. Конфигурируется администатором WebSSO.
- **scope** - атрибут access token, который может определять набор данных пользователя, к которым получает доступ защищаемый сервис. Может, так же, определять список идентификаторов защищенных ресурсов, которые могут быть использованы сервисом для данного пользователя.
- **URL обратного редиректа** - HTTP URL на который необходимо выполнить переход после успешного или неуспешного завершения аутентификации.
- **encoded URL** - Обычный HTTP URL, в котором все зарезервированные URI символы заменены на их представление в виде %XX.
- **токен автоматического входа** - зашифрованный токен в формате Json Web Token с информацией о пользователе.

4. Общие положения

- Все параметры запросов и ответов, атрибуты пользователей и прочие параметры являются регистрозависимыми
- Все параметры запросов и ответов являются обязательными, если явно не указано обратное
- Переносы строк в некоторых примерах запросов добавлены для удобства чтения, реальная строка запроса должна быть без них
- Клиент должен поддерживать HTTP cookies
- При запросах к API, ошибки со статусом 503 всегда приходят в HTML.

5. Автоматический вход

НАПОМИНАЕМ

Данный шаг является опциональным и может быть пропущен, если автоматический вход не требуется.

Для возможности автоматического входа по технологии HTTP Header Enrichment необходимо получить токен автоматического входа отдельным POST-запросом. Запрос должен обязательно идти по протоколу http, а не https.

5.1. Формат запроса

```
POST /sso/auth/autologin
Host: <sso_host>
Accept: application/json
```

- ‹sso_host› - базовый адрес сервера WebSSO, например sso.rooxteam.com

5.1.1. Формат ответа

В ответе всегда возвращается зашифрованный токен, который нужно передать в запросе на access_token:

```
HTTP/1.1 200 OK
```

```
{
  "auto-login-jwt": "eyJhbGciOiJSU0EtT0FFUCIsImVuYyI6IkExMjhHQ00ifQ.enSg9mzML5y-
RE_tFDMh1Ixwrb8yYX-bPN8X-
rz9nHFQFhb1yGyQ5RvfEDFl2hpfDPbMTi5WzHaC79FFwkHWwdneDYMCnFlGBMTpZGNRrQxu6yItycAY_GEAURpW3_3
BT2Dkj78Q-faD4XoeREodhaPsfQrAOTunMLODCwia5Rs.6r2M6pqz47qe5aGQ6L6n5I-
U55qZ5ISd4aik4YWa.jO9EBlwfaDONNn6BZRo0dNgndUejGW QxpeLljnxY.SJ8AtbswxqHFf6Wb_gd5BQ"
}
```

Параметры

- ‹auto-login-jwt› - токен автоматического входа, строка

6. Аутентификация

Аутентификация выполняется по протоколу OAuth2.0 с передачей специального значения параметра grant_type. Для попытки автоматического входа может быть передан параметр auto-login-jwt, полученный в ответе на шаге Автоматический вход.

В случае успешного автовхода, в ответе будет сразу возвращен access_token. В случае неуспешного автовхода, ответ будет содержать данные о форме для входа по логину и паролю.

6.1. Формат запроса

```
POST /sso/oauth2/access_token
Host: <sso_host>
Accept: application/json
Content-Type: application/x-www-form-urlencoded

client_id=<client_id>&
client_secret=<client_secret>&
scope=<scope>&
grant_type=urn:roox:params:oauth:grant-type:m2m&
realm=%2Fcustomer&
service=dispatcher&
auto-login-jwt=<auto-login-jwt>&
form_type=<form_type>
```

6.1.1. Параметры

- <sso_host> - базовый адрес сервера WebSSO, строка, например sso.rooxteam.com
- <realm> - группа пользователей WebSSO, строка, всегда используется значение %2Fcustomer, которое является uri-encoded значением /customer
- <client_id> - идентификатор клиента, строка, например selfcare
- <client_secret> - пароль клиента, строка
- <scope> - OAuth2.0 scope в кодировке UTF-8, строка, **опционально**, **регистрозависимо**
- <grant_type> - способ авторизации пользователя, строка, всегда используется значение urn:roox:params:oauth:grant-type:m2m
- <service> - имя цепочки аутентификации, строка, всегда dispatcher
- <auto-login-jwt> - токен автоматического входа, строка, **опционально**
- <form_type> - тип формы: (login-passwordlpin), строка, **опционально**

6.1.2. Формат ответа при успешном автовходе

```
HTTP/1.1 200 OK
```

```
{
```

```
  "scope": [
    "cn"
  ],
  "expires_in": 59,
  "token_type": "Bearer",
  "access_token": "89fcd81f-e72b-4dbe-a2aa-71b57c519de2"
}
```


Параметры

- scope - список scope (массив строк) разрешенных для использования от имени пользователя, возможные значения задаются конфигурацией
- expires_in - время до истечения срока действия токена, челое число, в секундах
- token_type - тип выданного токена, строка, всегда Bearer
- access_token - выданный access token, строка

6.1.3. Формат ответа сервера после неуспешного автовхода

Если автовход был пропущен, либо закончился неуспешно, WebSSO отправляет форму для получения параметров входа (таких как логин/пароль):

```
HTTP/1.1 200 OK
```


```
{
  "form": {
    "errors": [],
    "name": "loginForm",
    "fields": {
      "username": {
        "constraints": [
          {
            "name": "NotNull"
          },
          {
            "name": "Size",
            "attributes": {
              "min": 10,
              "max": 25
            }
          },
          {
            "name": "FilteredSize",
            "attributes": {
              "skip": "(^[^9]+)|([^0-9])",
              "min": 10,
              "max": 10
            }
          }
        ]
      },
      "password": {
        "constraints": [
          {
            "name": "Size",
            "attributes": {
              "min": 4,
              "max": 1024
            }
          },
          {
            "name": "NotNull"
          }
        ]
      }
    }
  },
  "view": {
    "blockedFor": null,
    "isBlocked": false
  },
  "serverUrl": "https://sso.rooxteam.com/sso/auth/websso",
  "step": "auth_form",
```

```
  "execution": "dec7dacf-as2d-4345-aae5-d08410cb8f95_H4sIAAAAAAAAAN1WTWwcRRYu27GdxE5wEnYF"
}
```


Параметры

- form - форма получения параметров аутентификации пользователя, структура
- errors - список ошибок, сообщение может быть привязано к полю (field), массив
- form.name - имя формы, строка
- fields - список полей формы, структура
- constraints - список ограничений на поле, массив структур
- constraints.name - имя ограничения, строка
- constraints.attributes - перечень атрибутов ограничений, структура
- constraints.attributes.min - минимальное значение для данного атрибута, целое число
- constraints.attributes.max - максимальное значение для данного атрибута, целое число
- constraints.attributes.skip - регулярное выражение валидации значения для поля, строка, **опционально**
- view - дополнительные данные текущего шага, структура
- isBlocked - признак блокировки пользователя, bool
- blockedFor - время до разблокировки в секундах, целое число, **опционально**
- serverUrl - URL для запросов на сервер, строка
- step - идентификатор шага, строка
- execution - идентификатор предсессии аутентификации, строка

Если form_type=login-password:

- username - атрибуты, относящиеся к имени пользователя
- password - атрибуты, относящиеся к паролю

Если form_type=pin:

- username - `атрибуты`, относящиеся к имени пользователя
- password - атрибуты, относящиеся к паролю

Если form_type=login-lite:

- msisdn - номер телефона, владельца легкой УЗ

6.2. Передача логина и пароля

В этом запросе используется значение параметра execution из предыдущего ответа сервера. В случае успешной аутентификации будет возвращен access_token. В случае неуспешной аутентификации будет возвращен ответ, аналогичный ответу сервера после неуспешного автовхода, содержащий описание ошибок.

6.2.1. Формат запроса

```
POST /sso/oauth2/access_token HTTP/1.1
Host: <sso_host>
Accept: application/json
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded

client_id=<client_id>&
client_secret=<client_secret>&
scope=<scope>&
grant_type=urn:roox:params:oauth:grant-type:m2m&
realm=%2Fcustomer&
service=dispatcher&
execution=<execution>&
username=<username>&
password=<password>&
_eventId=next&
```

Если form_type=login-password:

```
username=<username>&
password=<password>
```

Если form_type=pin:

```
pin_code=<pin_code>&
install_id=<install_id>&
device_info=<device_info>
```

Если form_type=login-lite:

```
msisdn=<msisdn>
```

Параметры

- ‹sso_host› - базовый адрес сервера WebSSO, например sso.rooxteam.com, строка
- ‹client_id› - идентификатор клиента, например selfcare, строка

- <client_secret> - пароль клиента, строка
- <scope> - OAuth2.0 scope в кодировке UTF-8, строка, **опционально**, **регистрозависимо**
- <grant_type> - способ авторизации пользователя, строка, всегда используется значение urn:roox:params:oauth:grant-type:m2m
- <realm> - группа пользователей WebSSO, строка, всегда используется значение %2Fcustomer, которое является uri-encoded значением /customer
- <execution> - идентификатор предсессии аутентификации, строка, значение берется из предыдущего ответа сервера
- <service> - имя цепочки аутентификации, строка, всегда dispatcher
- <username> - имя пользователя (номер телефона), строка
- <password> - пароль пользователя, строка
- <pin_code> - пин-код, строка
- <install_id> - идентификатор установки приложения, строка
- <device_info> - информация об устройстве, структура, пример:

```
{
 "app_version": 224,
 "device_id": "000000000000000",
 "device_root": 1,
 "device_locale": "en",
 "device_os": 2,
 "device_os_version": "6.0"
}
```

- <_eventId> - идентификатор действия, строка, всегда next

Формат успешного ответа

Если form_type=login-password:

```
HTTP/1.1 200 OK
```

```
{
  "scope": [
    "cn"
  ],
  "expires_in": 59,
  "token_type": "Bearer",
  "access_token": "89fcd81f-e72b-4dbe-a2aa-71b57c519de2"
}
```

Если form_type=pin:

```
HTTP/1.1 200 OK
```


```
{
  "user_info": {
    "phone": "79001112233",
    "region_id": "77",
    "client_id": "782990245",
    "hashed_client_id": "",
    "type": "",
    "info_message": null,
    "info_title": null,
    "full_name": "Чирков Павел Александрович",
    "pseudonym": "",
    "manager_name": "",
    "manager_phone": "",
    "p2p_transfer_url": ""
  },
  "scope": "cid cn givenname sn telephoneNumber user_name",
  "authenticated": true,
  "JWTToken": "eyAiYWxn...vlyYyevg",
  "expires_in": 1199,
  "strings": {
    "applepay_info_button_text": "...",
    "applepay_card_added_text": "...",
    "digital_card_info_title": "...",
    "applepay_action_title": "...",
    "digital_card_info_text": "...",
    "applepay_action_text": "...",
    "applepay_info_title": "...",
    "applepay_info_text": "...",
    "card_replacement_price_text": "...",
    "card_replacement_time_text": "...",
    "card_replacement_hint_text": "..."
  },
  "clientId": "sso_____46756",
  "PolicyContext": "eyAiYWxn....9-m2lPxA",
  "userId": "sso_____39621",
  "token_type": "JWTToken",
  "login": "tilitilichki",
  "old_token": "tb92sW4O...==172657",
  "refresh_token": "95b122b0-db18-443b-bc8f-5e1423a99139",
  "access_token": "eyAiYWxn...vlyYyevg"
}
```


Параметры

- scope - список scope (массив строк), разрешенных для использования от имени пользователя, возможные значения задаются конфигурацией
- expires_in - время до истечения срока действия токена в секундах, целое число
- token_type - тип выданного токена, строка, всегда Bearer
- access_token - выданный access token, строка
- refresh_token - выданный refresh token, строка
- old_token - токен, выданный внешней системой, строка

Формат ответа при неверном пароле

```
HTTP/1.1 200 OK
```


```
{
  "form": {
    "errors": [
      {
        "message": "invalid_credentials"
      }
    ],
    "name": "loginForm",
    "fields": {
      "username": {
        "constraints": [
          {
            "name": "NotNull"
          },
          {
            "name": "Size",
            "attributes": {
              "min": 10,
              "max": 25
            }
          },
          {
            "name": "FilteredSize",
            "attributes": {
              "skip": "(^[^9]*)|([^\\d])",
              "min": 10,
              "max": 10
            }
          }
        ]
      },
      "password": {
        "constraints": [
          {
            "name": "Size",
            "attributes": {
              "min": 4,
              "max": 1024
            }
          },
          {
            "name": "NotNull"
          }
        ]
      }
    }
  },
  "view": {
    "blockedFor": null,
    "isBlocked": false
  },
  "serverUrl": "https://sso.rooxteam.com/sso/auth/websso",
  "step": "auth_form",
```

```
  "execution": "dec7dacf-as2d-4345-aae5-d08410cb8f95_H4sIAAAAAAAAAN1WTWwcRRYu27GdxE5wEnYF"
}
```


Параметры

- form - форма получения параметров аутентификации пользователя, структура
- errors - список ошибок, сообщение может быть привязано к полю (field), структура
- message - сообщение об ошибке, строка
- form.name - имя формы, строка
- fields - список полей формы, массив структур
- username - атрибуты, относящиеся к имени пользователя, структура
- password - атрибуты, относящиеся к паролю, структура
- constraints - список ограничений на поле, массив структур
- constraints.name - имя ограничения, строка
- constraints.attributes - перечень атрибутов ограничений, структура
- constraints.attributes.min - минимальное значение для данного атрибута, целое число
- constraints.attributes.max - максимальное значение для данного атрибута, целое число
- constraints.attributes.skip - регулярное выражение валидации значения для поля, строка, **опционально**
- view - дополнительные данные текущего шага, структура
- isBlocked - признак блокировки пользователя, bool
- blockedFor - время до разблокировки в секундах, целое число, **опционально**
- serverUrl - URL для запросов на сервер, строка
- step - идентификатор шага, строка
- execution - идентификатор предсессии аутентификации, строка

Формат неуспешного ответа с требованием ввода символов captcha

```
HTTP/1.1 200 OK
```


```
{
  "form": {
    "errors": [
      {
        "message": "need_captcha"
      }
    ],
    "name": "captchaLoginForm",
    "fields": {
      "username": {
        "constraints": [
          {
            "name": "NotNull"
          },
          {
            "name": "Size",
            "attributes": {
              "min": 10,
              "max": 25
            }
          },
          {
            "name": "FilteredSize",
            "attributes": {
              "skip": "(^[^9]*)|([^\\d])",
              "min": 10,
              "max": 10
            }
          }
        ]
      },
      "captchaCode": {
        "constraints": [
          {
            "name": "Size",
            "attributes": {
              "min": 1,
              "max": 256
            }
          },
          {
            "name": "NotNull"
          }
        ]
      },
      "password": {
        "constraints": [
          {
            "name": "Size",
            "attributes": {
              "min": 4,
              "max": 1024
            }
```

```
          },
          {
            "name": "NotNull"
          }
        ]
      }
    }
  },
  "view": {
    "captchaUrl": "https://sso.rooxteam.com/sso/api/captchas/79e2333a-fede-4c77-bf5d-
65762121bd07"
  },
  "serverUrl": "https://sso.rooxteam.com/sso/auth/websso",
  "step": "captcha_auth_form",
  "execution": "dec7dacf-as2d-4345-aae5-d08410cb8f95_H4sIAAAAAAAAAN1WTWwcRRYu27GdxE5wEnYF"
}
```


Параметры

- form - форма получения параметров аутентификации пользователя, структура
- errors - список ошибок, сообщение может быть привязано к полю (field), массив структур
- form.name - имя формы, строка
- fields - список полей формы, структура
- username - атрибуты, относящиеся к имени пользователя, структура
- password - атрибуты, относящиеся к паролю, структура
- constraints - список ограничений на поле, массив структур
- constraints.name - имя ограничения, строка
- constraints.attributes - перечень атрибутов ограничений, структура
- constraints.attributes.min - минимальное значение для данного атрибута, целое число
- constraints.attributes.max - максимальное значение для данного атрибута, целое число
- constraints.attributes.skip - регулярное выражение валидации значения для поля, строка, **опционально**
- view - дополнительные данные текущего шага, структура
- isBlocked - признак блокировки пользователя, bool
- blockedFor - время до разблокировки в секундах, целое число, **опционально**
- serverUrl - URL для запросов на сервер, строка
- step - идентификатор шага, строка
- execution - идентификатор предсессии аутентификации, строка

6.3. Проверка логина-пароля при использовании в решении протокола SRP

НАПОМИНАЕМ

Необходимость использования протокола SRP определяется требованиями информационной безопасности к данной инсталляци. Уточните у руководителя проекта, какой вариант надо реализовывать на стороне клиента в данном проекте.

В сценарии SRP используются те же операции, что и в сценарии с передачей логина-пароля. Но данный сценарий требует выполнения нескольких шагов и наличия предварительно настроенных констант на стороне клиента и сервера.

Клнфигурация:

И для клиента и для сервера устанавлюваются общие константы:

Таблица 1. Конфигурация SRP для клиента (приведены параметыр виджета логина, в других клиентах могут использоваться свои названия)

Параметр	Описание	Пример
com.rooxteam.widgets.srp.n	"безопасное простое" = 2*q+1 простое, где q тоже простое число. Число в 10-чной системе.	11144252…13345603
com.rooxteam.widgets.srp.g	генератор по модулю N = › для любого 0 ‹ X ‹ N существует и единственный x такой, что g^x % N = X.	2
com.rooxteam.widgets.srp.h	односторонняя функция хеширования, достаточно безопасная на текущий момент	SHA-1
com.rooxteam.widgets.srp.salt_bytes	размер соли в байтах	16

Таблица 2. Конфигурация SRP для сервера

Параметр	Описание	Пример
com.rooxteam.sso.srp.n	"безопасное простое" = 2*q+1 простое, где q тоже простое число. Число в 10-чной системе.	11144252…13345603
com.rooxteam.sso.srp.g	генератор по модулю N = › для любого 0 ‹ X ‹ N существует и единственный x такой, что g^x % N = X.	2
com.rooxteam.sso.srp.h	односторонняя функция хеширования, достаточно безопасная на текущий момент	SHA-1
com.rooxteam.sso.auth_with_challenge	(boolean) включить аутентификацию через SRP-Challenge	true

Сценарий аутентификации:

Таблица 3. Сценарий работы SRP

Шаг	Клиент	Сервер
1	- Получает от пользователя логин login и пароль password - Отправляет серверу login	- Вытаскивает из базы соль salt и верификатор verifier по логину login - Вычисляет переменную B = step1(login, salt, verifier) - Возвращает клиенту salt и B
2	- Вычисляет и передаёт серверу A и M1	- Проверяет, можно ли доверять A и M1 - Успешно завершает аутентификацию - Вычисляет и отправляет клиенту M2
3	- Проверяет, можно ли доверять M2	

Шаг 1

Выходные данные (от клиента к серверу)

имя	обязательный	описание	констрейнты
_eventId	да	next	
username	да	Логин пользователя	набор из символов, минимум 1, максимум 1024
password	да	Требуется любое непустое значение	набор из символов, минимум 1, максимум 1024

Входные данные (от сервера к клиенту)

имя	обязательный	описание	констрейнты
execution	да	Идентификатор предсессии аутентификации	
view.salt	да	Соль	Число в 16-чном формате
view.B	да	Переменная B	Число в 16-чном формате

Шаг 2

Выходные данные (от клиента к серверу)

имя	обязательный	описание	констрейнты

_eventId	да	next	
A	да	Переменная A	Число в 16-чном формате
M1	да	Переменная M1	Число в 16-чном формате

Входные данные (от сервера к клиенту)

имя	обязательный	описание	констрейнты
execution	да	Идентификатор предсессии аутентификации	
view.M2	да	Переменная M2	Число в 16-чном формате

6.4. Получение графического изображения captcha по идентификатору

После превышения заданного числа попыток входа WebSSO передает ссылку для получения графического изображения captcha и требует в следующем запросе передать символы этого изображения.

6.4.1. Формат запроса

```
GET <captcha_url>
Host: <captcha_host>
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
```

Параметры

- <captcha_url> - URL для загрузки графического изображения captcha, строка, берется из предыдущего ответа сервера
- <captcha_host> - базовый адрес сервера для получения графического изображения captcha, строка

В ответе придет графическое изображение символов captcha.

6.5. Передача логина, пароля и символов captcha

В этом запросе используется значение параметра execution из неуспешного ответа с требованием ввода символов captcha. В случае успешной аутентификации будет возвращен access_token. В случае неуспешной аутентификации будет возвращен ответ, содержащий описание ошибок.

6.5.1. Формат запроса

```
POST /sso/oauth2/access_token
Host: <sso_host>
Accept: application/json
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded

client_id=<client_id>&
client_secret=<client_secret>&
scope=<scope>&
grant_type=urn:roox:params:oauth:grant-type:m2m&
realm=%2Fcustomer&
service=dispatcher&
execution=<execution>&
username=<username>&
password=<password>&
captchaCode=<captcha_code>&
_eventId=next
```

Параметры

- <realm> - группа пользователей WebSSO, строка, всегда используется значение %2Fcustomer, которое является uri-encoded значением /customer
- <sso_host> - базовый адрес сервера WebSSO, строка, например sso.rooxteam.com
- <client_id> - идентификатор клиента, строка, например selfcare
- <service> - имя цепочки аутентификации, строка, всегда dispatcher
- <client_secret> - пароль клиента, строка
- <scope> - OAuth2.0 scope в кодировке UTF-8, строка, **опционально**, **регистрозависимо**
- <execution> - идентификатор предсессии аутентификации, строка, значение берется из предыдущего ответа сервера
- <grant_type> - способ авторизации пользователя, строка, всегда используется значение urn:roox:params:oauth:grant-type:m2m
- <username> - имя пользователя, строка
- <password> - пароль пользователя, строка
- <captcha_code> - символы captcha, строка
- <_eventId> - идентификатор действия, строка, всегда next

Формат успешного ответа

```
HTTP/1.1 200 OK
```

```
{
  "scope": [
    "cn"
  ],
  "expires_in": 59,
  "token_type": "Bearer",
  "access_token": "d557415e-c8a5-4daf-a9f8-1c7a5f8b5af7"
}
```

Параметры

- <scope> - список scope (массив структур), разрешенных для использования от имени пользователя, возможные значения задаются конфигурацией
- <expires_in> - время до истечения срока действия токена в секундах, целое число
- <token_type> - тип выданного токена, строка, всегда Bearer
- <access_token> - выданный access token, строка

Формат неуспешного ответа при неверном значении captcha

```
HTTP/1.1 200 OK
```


```
{
  "form": {
    "errors": [
      {
        "field": "captchaCode",
        "message": "invalid_captcha"
      }
    ],
    "name": "captchaLoginForm",
    "fields": {
      "username": {
        "constraints": [
          {
            "name": "NotNull"
          },
          {
            "name": "Size",
            "attributes": {
              "min": 10,
              "max": 25
            }
          },
          {
            "name": "FilteredSize",
            "attributes": {
              "skip": "(^[^9]*)|([^\\d])",
              "min": 10,
              "max": 10
            }
          }
        ]
      },
      "captchaCode": {
        "constraints": [
          {
            "name": "Size",
            "attributes": {
              "min": 1,
              "max": 256
            }
          },
          {
            "name": "NotNull"
          }
        ]
      },
      "password": {
        "constraints": [
          {
            "name": "Size",
            "attributes": {
              "min": 4,
              "max": 1024
```

```
            }
          },
          {
            "name": "NotNull"
          }
        ]
      }
    }
  },
  "view": {
    "captchaUrl": "https://sso.rooxteam.com/sso/api/captchas/e1bd0719-f8cd-4aa9-9bc6-
d8c1c834932e"
  },
  "serverUrl": "https://sso.rooxteam.com/sso/auth/websso",
  "step": "captcha_auth_form",
  "execution": "dec7dacf-as2d-4345-aae5-d08410cb8f95_H4sIAAAAAAAAAN1WTWwcRRYu27GdxE5wEnYF"
}
```


Параметры

- form - форма получения параметров аутентификации пользователя, структура
- errors - список ошибок, сообщение может быть привязано к полю (field), массив структур
- field - поле, к которому привязоно сообщение об ошибке, строка
- form.name - имя формы, строка
- fields - список полей формы, структура
- username - атрибуты, относящиеся к имени пользователя, структура
- password - атрибуты, относящиеся к паролю, структура
- constraints - список ограничений на поле, массив структур
- constraints.name - имя ограничения, строка
- constraints.attributes - перечень атрибутов ограничений, структура
- constraints.attributes.min - минимальное значение для данного атрибута, целое число
- constraints.attributes.max - максимальное значение для данного атрибута, целое число
- constraints.attributes.skip - регулярное выражение валидации значения для поля, строка, **опционально**
- view - дополнительные данные текущего шага, структура
- isBlocked - признак блокировки пользователя, bool
- blockedFor - время до разблокировки в секундах, число, **опционально**
- serverUrl - URL для запросов на сервер, строка
- step - идентификатор шага, строка
- execution - идентификатор предсессии аутентификации, строка

6.6. Блокировка пользователя

После заданного количества неверных попыток ввода пароля для одного и того же логина требуется ввод символов captcha. После заданного количества неверных попыток ввода пароля или символов captcha для одного и того же логина происходит временная блокировка по логину. После заданного количества неверных попыток ввода логина/пароля/captcha с одного и того же IP-адреса в течение определенного количества секунд, происходит блокировка этого адреса по IP.

6.6.1. Формат ответа при временной блокировке по логину

```
HTTP/1.1 200 OK
```


```
{
  "form": {
    "errors": [
      {
        "message": "user_blocked"
      }
    ],
    "name": "loginForm",
    "fields": {
      "username": {
        "constraints": [
          {
            "name": "NotNull"
          },
          {
            "name": "Size",
            "attributes": {
              "min": 10,
              "max": 25
            }
          },
          {
            "name": "FilteredSize",
            "attributes": {
              "skip": "(^[^9]*)|([^\\d])",
              "min": 10,
              "max": 10
            }
          }
        ]
      },
      "password": {
        "constraints": [
          {
            "name": "Size",
            "attributes": {
              "min": 4,
              "max": 1024
            }
          },
          {
            "name": "NotNull"
          }
        ]
      }
    }
  },
  "view": {
    "blockedFor": 3000,
    "isBlocked": true
  },
  "serverUrl": "https://sso.rooxteam.com/sso/auth/websso",
```

```
  "step": "auth_form",
  "execution": "dec7dacf-as2d-4345-aae5-d08410cb8f95_H4sIAAAAAAAAAN1WTWwcRRYu27GdxE5wEnYF"
}
```


Параметры

- form - форма получения параметров аутентификации пользователя, структура
- errors - список ошибок, сообщение может быть привязано к полю (field), массив структур
- form.name - имя формы, строка
- fields - список полей формы, структура
- username - атрибуты, относящиеся к имени пользователя, структура
- password - атрибуты, относящиеся к паролю, структура
- constraints -список ограничений на поле, структура
- constraints.name - имя ограничения, строка
- constraints.attributes - перечень атрибутов ограничений, структура
- constraints.attributes.min - минимальное значение для данного атрибута, целое число
- constraints.attributes.max - максимальное значение для данного атрибута, целое число
- constraints.attributes.skip - регулярное выражение валидации значения для поля, строка, **опционально**
- view - дополнительные данные текущего шага, структура
- isBlocked - признак блокировки пользователя, bool
- blockedFor - время до разблокировки в секундах, число, **опционально**
- serverUrl - URL для запросов на сервер, строка
- step - идентификатор шага, строка
- execution - идентификатор предсессии аутентификации, строка

6.7. Блокировка доступа абонента к конкретному ресурсу по client_id

Абоненту может быть заблокирован доступ к конкретному ресурсу по client_id, в этом случае, после передачи логина и пароля вернется ответ с HTTP кодом 400 "Bad Request"

6.7.1. Формат ответа

```
HTTP/1.1 400 Bad Request
```


```
{
  "error_description": "The resource owner or authorization server denied the request.",
  "error": "access_denied"
}
```


Параметры

- error_description - текстовое описание ошибки, строка
- error - код ошибки согласно спецификации OAuth 2.0 (https://tools.ietf.org/html/rfc6749#section-5.2[RFC 6749 пункт 5.2]), строка

6.8. Возможные ограничения на поля формы

название	описание	атрибуты	описание атрибута
NotNull	Обязательность поля		
Size	Длина строкового параметра	min	Минимальная длина
		max	Максимальная длина
Pattern	Regexp для строкового параметра	regexp	Регулярное выражение
Min	Минимальное значение целого числового параметра	value	Минимальное значение
Max	Максимальное значение целого числового параметра	value	Максимальное значение

название	описание	атрибуты	описание атрибута
FilteredSize	Ограничения для номера телефона	min	Минимальная длина
		max	Максимальная длина
		skip	Регулярное выражение для фильтрации. Перед отправкой нужно удалять все, что ему удовлетворяет. Например, происходит очищение введенных данных ото всех символов кроме цифр, далее удаляются все символы слева до первой девятки.

6.9. Возможные коды ошибок

код	описание	комментарий
ip_blocked	Заблокирован IP-адрес	Пока не реализовано
user_blocked	Заблокирован пользователь по данному логину	
invalid_captcha	Неверное значение captcha	
need_captcha	Отсутствует значение captcha	Возникает, когда требуется ввод символов captcha, но они не передаются после начала новой сессии
invalid_credentials	Неверная пара логин/пароль	
may not be null	Поле не может быть пустым	Ошибка валидации
size must be between {min} and {max}	Значение должно быть в заданном диапазоне	Ошибка валидации
required on {field name}	Обязательное поле	Ошибка валидации

7. Валидация токена доступа

Ранее полученный токен может оказаться просрочен или инвалидирован, поэтому перед выполнением клиентом действий от имени пользователя, необходимо выполнять валидацию токена. Для валидации токена нужно выполнить запрос на соответствующий URL WebSSO с указанием access token. Результатом будет информация о переданном токене либо сообщение об ошибке.

Помимо простой валидации токена можно запрашивать разрешение на доступ к защищаемому ресурсу. Для этого нужно передать в запрос на валидацию токена дополнительный параметр scope. Можно передать любой scope, не обязательно указанный при аутентификации. Если доступ разрешен, то ответ будет аналогичен ответу без указания метода. Если нет, ответ будет содержать информацию о требуемом уровне авторизации для получения доступа к запрашиваемому методу.

Если требуется передать дополнительные параметры для аудита доступа к защищенным ресурсам, необходимо выполнить POST-запрос и передать параметры в теле запроса в виде json-объекта.

7.1. Формат запроса

```
POST /sso/oauth2/tokeninfo?access_token=<access_token>&scope=<scope>
Host: <sso_host>
Content-Type: application/json
Accept: application/json
```

```
{
  "httpMethod": "POST",
  "url": "http://example.com/some/url",
  "headers": {
    "User-Agent": [
      "Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/40.0.2214.115 Safari/537.36"
    ],
    "Referer": [
      "https://www.google.com/"
    ],
    "X-Nokia-MSISDN": [
      "9876543210"
    ],
    "X-Forwarded-For": [
      "10.20.30.40",
      "10.10.35.46",
      "192.168.12.74"
    ]
  }
}
```

7.1.1. Параметры

- ‹sso_host› - базовый адрес сервера WebSSO, строка, например sso.rooxteam.com, может содержать порт, например sso.rooxteam.com:8080
- access_token - полученный в результате аутентификации access token, строка, например 7bdaeacc-3d80-415c-920f-a7c30ca5e743

- scope - идентификатор ресурса защищаемого сервиса в кодировке UTF-8, см. подробнее в пункте Использование параметра scope, строка, **опционально**, **регистрозависимо**
- httpMethod - HTTP метод для которого запрашивается доступ, строка, **опционально**
- url - URL для которого запрашивается доступ, строка, **опционально**
- headers - список HTTP заголовков из запроса пользователя к защищаемому сервису, передаются без дополнительной обработки или фильтрации, массив строк, **опционально**

7.1.2. Формат успешного ответа

```
HTTP/1.1 200 OK
```

```
{
  "scope": [
    "cn",
    "networkAuthenticationType"
  ],
  "cn": "9263752235",
  "realm": "/customer",
  "token_type": "Bearer",
  "expires_in": 28,
  "access_token": "7bdaeacc-3d80-415c-920f-a7c30ca5e743",
  "auth_level": "2",
  "networkAuthenticationType": "AUTO",
  "client_id": "selfcare"
}
```

7.1.3. Параметры

- scope - список scope (массив строк) разрешенных для использования от имени пользователя, возможные значения задаются конфигурацией
- realm - группа пользователей WebSSO, строка, всегда возвращается значение /customer
- token_type - тип выданного токена, строка, всегда Bearer
- expires_in - время до истечения срока действия токена в секундах, целое число
- access_token - проверямый access token, строка
- auth_level - выданный уровень авторизации пользователя, строка
- client_id - идентификатор клиента, которому выдан токен, строка, возможные значения зависят от конфигурации
- cn - номер телефона пользователя, строка
- networkAuthenticationType - разрешен ли автоматический вход, строка

Набор возможных значений scope зависит от конфигурации.

7.1.4. Формат неуспешного ответа

При передаче невалидного токена или токена с истекшим сроком действия, а также при блокировке учетной записи абонента и при блокировке абоненту доступа к конкретному ресурсу по client_id будет возвращен ответ:

```
HTTP/1.1 401 Unauthorized
```

```
{
  "error": "expired_token",
  "error_description": "The request contains a token no longer valid."
}
```

Параметры

- error - код ошибки согласно спецификации OAuth 2.0 (https://tools.ietf.org/html/rfc6749#section-5.2[RFC 6749 пункт 5.2]), строка
- error_description - текстовое описание ошибки, строка

7.1.5. Формат неуспешного ответа при недостаточном уровне авторизации

```
HTTP/1.1 403 Forbidden
```

```
{
  "scope": [
    "cn"
  ],
  "cn": "9876543210",
  "realm": "/customer",
  "token_type": "Bearer",
  "expires_in": 31,
  "advices": {
    "required_auth_level": "5"
  },
  "auth_level": "2",
  "access_token": "5fdfeafd-3061-4b1c-9076-0fe460f91fc8",
  "client_id": "selfcare"
}
```

Параметры

- scope - список scope (массив строк) разрешенных для использования от имени пользователя, возможные значения задаются конфигурацией
- realm - группа пользователей WebSSO, строка, всегда возвращается значение /customer
- token_type - тип выданного токена, строка, всегда Bearer
- expires_in - время до истечения срока действия токена в секундах, целое число
- advices - перечень необходимых условий, которые требуется удовлетворить для получения доступа к защищаемому ресурсу, структура
- required_auth_level - требуемый уровень авторизации для данного scope, строка
- access_token - проверямый access token, строка

- auth_level - выданный уровень авторизации пользователя, строка
- client_id - идентификатор клиента, которому выдан токен, строка, возможные значения зависят от конфигурации

7.1.6. Формат ответа при блокировке OAuth 2.0 клиента

```
HTTP/1.1 403 Forbidden
```

```
{
  "error": "client_blocked",
  "error_description": "Client is blocked."
}
```

Параметры

- error - код ошибки согласно спецификации OAuth 2.0 Token Revocation (http://tools.ietf.org/html/rfc7009#section-4.1.1[RFC 7009 пункт 4.1.1]), строка
- error_description - текстовое описание ошибки, строка

8. Повышение уровня авторизации WebSSO с помощью OTP

8.1. Схема повышения уровня авторизации

Цветом выделены состояния которые передают управление виджету.

Повышение уровня авторизации происходит через OAuth 2.0 с передачей параметра auth_level. Если уровень авторизации пользователя меньше запрашиваемого, он будет перенаправлен на виджет повышения уровня по OTP коду.

Если SMS-код был успешно подтвержден, будет возвращен access_token с повышенным уровнем авторизации auth_level

Повышенный уровень авторизации действует фиксированное время (по-умолчанию 3 минуты), по истечение которого уровень авторизации возвращается к исходному значению.

ЗАМЕТКА

Следует заметить, что старый access token продолжает действовать и для запросов, где не нужен повышеный уровень авторизации, нужно использовать его.

ЗАМЕТКА

Время действия нового access token меньше чем у старого

ВАЖНО

Возможные ограничения на поля формы описаны в пункте Возможные ограничения на поля формы

ЗАМЕТКА

После успешного повышения уровня авторизации OAuth 2.0 /tokeninfo вернет auth_level соответствующий запрашиваемому

ЗАМЕТКА

В каждый запрос нужно подставлять текущий execution - идентификатор предсессии аутентификации.

8.2. Формат запроса на повышение уровня авторизации

```
POST /sso/oauth2/access_token
Host: <sso_host>
Accept: application/json
Content-Type: application/x-www-form-urlencoded

client_id=<client_id>&
client_secret=<client_secret>&
scope=<scope>&
grant_type=urn:roox:params:oauth:grant-type:m2m&
realm=/customer&
service=dispatcher&
auth_level=<auth_level>&
access_token=<access_token>&
method=<method>
```

8.2.1. Параметры

- sso_host - базовый адрес сервера WebSSO, например sso.rooxteam.com, может содержать порт, например sso.rooxteam.com:8080
- client_id - идентификатор клиента, например selfcare, возможные значения зависят от конфигурации
- client_secret - пароль клиента, возможные значения зависят от конфигурации
- scope - список запрашиваемых scope через пробел в кодировке UTF-8. **опционально**, **регистрозависимо**
- grant_type - способ авторизации пользователя, всегда используется значение urn:roox:params:oauth:grant-type:m2m
- service - имя цепочки аутентификации, всегда dispatcher
- realm - группа пользователей WebSSO, всегда используется значение %2Fcustomer, которое является uri-encoded значением /customer
- auth_level - желаемый уровень авторизации для текущего пользователя.
- method - предпочтительный способ повышения уровня авторизации, например otp_sms. **опционально**

Результатом выполения запроса, будет валидация существующего токена и отображение номера телефона абонента на который будет отправлена SMS.

8.3. Формат успешного ответа на повышение уровня авторизации

После получения результата надо отобразить пользователю форму для отправки OTP кода.

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
```

```
{
  "view": {
```

```
    "msisdn": "79876543210"
  },
  "serverUrl": "http://sso.rooxteam.com/sso/auth/otp-sms",
  "step": "send_otp_form",
  "execution": "dec7dacf-as2d-4345-aae5-d08410cb8f95_H4sIAAAAAAAAAN1WTWwcRRYu27GdxE5wEnYF"
}
```


8.3.1. Параметры

- serverUrl - URL для следующего запроса, например sso.rooxteam.com, может содержать порт, например sso.rooxteam.com:8080
- execution - идентификатор предсессии аутентификации
- step - код состояния
- view.msisdn - номер телефона, на который будет отправлен OTP SMS

8.4. Формат запроса на отправку OTP абоненту

Данный запрос отправит OTP код абоненту.

```
POST /sso/auth/otp-sms
Content-Type: application/x-www-form-urlencoded
Accept: application/json

execution=dec7dacf-as2d-4345-aae5-
d08410cb8f95_H4sIAAAAAAAAAN1WTWwcRRYu27GdxE5wEnYF&_eventId=send
```

8.4.1. Параметры

- execution - идентификатор предсессии аутентификации
- _eventId - идентификатор действия, определяется отдельно для каждого состояния

8.5. Формат успешного ответа на отправку OTP абоненту

Полученый ответ содержит в себе описание параметров с ограничениями на них, которые WebSSO будет ждать в следующем запросе.

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
```

```
{
  "form": {
    "fields": {
      "otpCode": {
        "constraints": [
          {
            "name": "NotNull"
          }
        ]
      }
    },
    "errors": [],
    "name": "otpForm"
  },
  "serverUrl": "http://sso.rooxteam.com/sso/auth/otp-sms",
  "step": "enter_otp_form",
  "execution": "dec7dacf-as2d-4345-aae5-
d08410cb8f95_H4sIAAAAAAAAAN1WTWwcRRYu27GdxE5wEnYF",
  "view": {
    "otpCodeAvailableAttempts": 2,
    "msisdn": "79876543210",
    "nextOtpPeriod": 120,
    "blockedFor": 0,
    "isBlocked": false
  }
}
```

8.5.1. Параметры

- serverUrl - URL для следующего запроса, например sso.rooxteam.com, может содержать порт, например sso.rooxteam.com:8080
- execution - идентификатор предсессии аутентификации
- step - код состояния
- form.fileds[] - поля, ожидаемые для отправки.
- view.msisdn - номер телефона, на который будет отправлен OTP SMS
- view.otpCodeAvailableAttempts - кол-во попыток ввода OTP кода
- view.nextOtpPeriod - время в секундах до наступления возможности отправки нового OTP кода
- view.isBlocked - признак блокировки пользователя
- view.blockedFor - время до разблокировки в секундах

8.6. Формат запроса на валидацию OTP

Данный запрос валидирует введенный OTP код пользователем. И результатом выполнения данного запроса будет окончание повышения авторизации пользователя с выдачей нового access token с повышеным уровнем авторизации.

```
POST /sso/auth/otp-sms
Content-Type: application/x-www-form-urlencoded
Accept: application/json

execution=dec7dacf-as2d-4345-aae5-
d08410cb8f95_H4sIAAAAAAAAAN1WTWwcRRYu27GdxE5wEnYF&_eventId=validate&otpCode=1234
```

8.6.1. Параметры

- execution - идентификатор предсессии аутентификации
- _eventId - идентификатор действия, определяется отдельно для каждого состояния
- otpCode - полученный OTP код

8.7. Формат успешного ответа на валидацию OTP

Полученый ответ содержит в себе новый access token с повышеным уровнем авторизации.

ЗАМЕТКА

Следует заметить, что старый access token продолжает действовать и для запросов, где не нужен повышеный уровень авторизации, нужно использовать его.

ЗАМЕТКА

Время действия нового access token меньше чем у старого

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "scope": [
    "cn"
  ],
  "expires_in": 59,
  "token_type": "Bearer",
  "access_token": "89fcd81f-e72b-4dbe-a2aa-71b57c519de2"
}
```

8.7.1. Параметры

- scope - список scope (в формате JSON Array) разрешенных для использования от имени пользователя, возможные значения задаются конфигурацией
- expires_in - время до истечения срока действия токена в секундах
- token_type - тип выданного токена. Всегда Bearer
- access_token - выданный access token

8.8. Формат ошибки валидации OTP

[source,http]j

HTTP/1.1 200 OK
Content-Type: application/json


```
{
  "form": {
    "errors": [
      {
        "field": "otpCode",
        "message": "invalid_otp"
      }
    ],
    "name": "otpForm",
    "fields": {
      "otpCode": {
        "constraints": [
          {
            "name": "NotNull"
          }
        ]
      }
    }
  },
  "view": {
    "otpCodeAvailableAttempts": 2,
    "msisdn": "79876543210",
    "nextOtpPeriod": 120,
    "blockedFor": 0,
    "isBlocked": false
  },
  "serverUrl": "http://sso.ocb.hosted:8080/sso/auth/otp-sms",
  "step": "otp_form",
  "execution": "dec7dacf-as2d-4345-aae5-d08410cb8f95_H4sIAAAAAAAAAN1WTWwcRRYu27GdxE5wEnYF"
}
```


8.8.1. Параметры

- execution - идентификатор предсессии аутентификации
- _eventId - идентификатор действия, определяется отдельно для каждого состояния
- view.msisdn - номер телефона, на который будет отправлен OTP SMS
- view.otpCodeAvailableAttempts - кол-во попыток ввода OTP кода
- view.nextOtpPeriod - время в секундах до наступления возможности отправки нового OTP кода
- view.isBlocked - признак блокировки пользователя
- view.blockedFor - время до разблокировки в секундах
- form.errors[].message - сообщение об ошибке валидации, **опционально**
- form.errors[].field - поле в котором найдена ошибка, **опционально**

8.9. Формат ответа при блокировке пользователя

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "form": {
    "errors": [
      {
        "message": "too_many_wrong_code"
      }
    ]
  },
  "view": {
    "blockedTo": "2015-02-18T12:00:00.000+00:00"
  },
  "serverUrl": "http://sso.rooxteam.com/sso/auth/otp-sms",
  "step": "otp_blocked_form",
  "execution": "dec7dacf-as2d-4345-aae5-d08410cb8f95_H4sIAAAAAAAAAN1WTWwcRRYu27GdxE5wEnYF"
}
```

8.9.1. Параметры

- execution - идентификатор предсессии аутентификации
- _eventId - идентификатор действия, определяется отдельно для каждого состояния
- view.blockedTo - дата, время в формате UTC определяет срок блокировки.
- form.errors[].message - сообщение об ошибке валидации, **опционально**
- form.errors[].field - поле в котором найдена ошибка, **опционально**

8.10. Возможные коды ошибок

код	описание	комментарий
invalid_otp	Неверный OTP код	
error_sending_otp	Ошибка приотправке SMS с кодом	
too_many_sms	Превышен лимит отправки SMS	
too_many_wrong_code	Превышен лимит попыток ввода OTP кода	
may not be null	Поле не может быть пустым	Ошибка валидации
size must be between {min} and {max}	Значение должно быть в заданном диапазоне	Ошибка валидации
required on {field name}	Обязательное поле	Ошибка валидации

9. Выход через API

Выданный ранее токен может быть инвалидирован отправкой запроса на соответствующий адрес WebSSO. Дополнительная аутентификация для этого метода не требуется, достаточно передать access token.

Для инвалидации выданного токена клиентское приложение защищаемого сервиса должно сделать ajax запрос на сервер защищаемого ресурса, после чего серверное приложение защищаемого ресурса должно выполнить запрос на отзыв токена в WebSSO, передав access_token пользователя.

9.1. Формат запроса

```
POST /sso/oauth2/revoke HTTP/1.1
Host: <sso_host>
Content-Type: application/x-www-form-urlencoded
Accept: application/json

token=<token>&token_type_hint=<token_type_hint>
```

9.1.1. Параметры

- ‹sso_host› - базовый адрес сервера WebSSO, строка, например sso.rooxteam.com, может содержать порт, например sso.rooxteam.com:8080
- ‹token› - токен для инвалидации, строка, например 5fdfeafd-3061-4b1c-9076-0fe460f91fc8
- ‹token_type_hint› - тип токена, строка, всегда access_token

9.1.2. Формат успешного ответа

```
HTTP/1.1 200 OK
```

9.1.3. Формат неуспешного ответа

Пример ответа при передаче невалидного типа токена:

```
HTTP/1.1 400 Bad Request
```

```
{
  "error_description": "Requested token type is not supported.",
  "error": "unsupported_token_type"
}
```

Параметры

- error - код ошибки согласно спецификации OAuth 2.0 Token Revocation (http://tools.ietf.org/html/rfc7009#section-4.1.1[RFC 7009 пункт 4.1.1]), строка
- error_description - текстовое описание ошибки, строка

10. Вызов callback URL при инвалидации токенов

WebSSO поддерживает отправку запроса на URL защищаемого сервиса с оповещением о факте инвалидации токена пользователя или инвалидации всех токенов защищаемого сервиса.

Защищаемый сервис может подписаться на события инвалидации токенов, выданных этому сервису ранее. Для этого необходимо сообщить один или несколько callback URL, на которые WebSSO будет отправлять оповещения. Список URL для оповещения настраивается администратором WebSSO.

10.1. Инвалидация токена пользователя

При инвалидации токена конкретного пользователя WebSSO отправляет соответствующее оповещение на callback URL. Защищаемый сервис, получив такое оповещение, должен немедленно инвалидировать сессию пользователя. Если защищаемый сервис использует кеширование результатов авторизации, кеш результатов авторизации данного пользователя также должен быть инвалидирован.

10.1.1. Формат оповещения о инвалидации токена

```
POST <callback_url>
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded

event=token_revoked&
global=false&
cn=<cn>&
access_token=<access_token>
```

Параметры

- callback_url - URL для оповещения из списка, заданного администратором WebSSO, строка
- event - имя события, строка
- global - флаг полной блокировки сервиса, bool, в этом оповещении всегда false
- cn - номер телефона пользователя (msisdn), строка
- access_token - инвалидированный access token, строка

10.2. Блокирование сервиса целиком

При блокировании сервиса целиком WebSSO отправляет соответствующее оповещение на callback URL. Защищаемый сервис, получив такое оповещение, должен немедленно инвалидировать сессии всех пользователей. Если защищаемый сервис использует кеширование результатов авторизации, кеш результатов авторизации всех пользователей также должен быть инвалидирован.

10.2.1. Формат оповещения о блокировке сервиса

```
POST <callback_url>
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
```

```
event=service_blocked&
global=true
```

Параметры

- callback_url - URL для оповещения из списка, заданного администратором WebSSO, строка
- event - имя события, строка
- global - флаг полной блокировки сервиса, bool, в этом оповещении всегда true

10.3. Понижение уровня авторизации токена

При понижении уровня авторизации токена конкретного пользователя WebSSO отправляет соответствующее оповещение на callback URL. Если защищаемый сервис использует кеширование результатов авторизации, кеш результатов авторизации всех пользователей также должен быть инвалидирован.

10.3.1. Формат оповещения о понижении уровня авторизации токена

```
POST <callback_url>
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded

event=token_auth_level_decreased&
global=false&
cn=<cn>&
access_token=<access_token>
```

Параметры

- callback_url - URL для оповещения из списка, заданного администратором WebSSO, строка
- event - имя события, строка
- global - флаг полной блокировки сервиса, bool, в этом оповещении всегда false
- cn - номер телефона пользователя (msisdn), строка
- access_token - измененный access token, строка

11. Использование параметра scope

Параметр scope (https://tools.ietf.org/html/rfc6749#section-3.3[RFC 6749 пункт 3.3]) может применяться в WebSSO для двух разных случаев:

- определение набора данных пользователя, к которым сервис получает доступ. По умолчанию любой запрос авторизации содержит scope cn, предоставляющий информацию о номере телефона. Для получения дополнительных атрибутов необходимо передать соответствующий список scope через пробел при аутентификации, например scope=networkAuthenticationType displayName. В этом случае при валидации токена будет возвращен расширенный список атрибутов. Возможные для запрашивания атрибуты конфигурируются администатором WebSSO отдельно для каждого сервиса.

Атрибуты могут включать:

название	**описание**
cn	Номер телефона пользователя (msisdn)
telephoneNumber	Номер телефона пользователя (msisdn), alias на cn
networkAuthenticationType	Тип аутентификации по сетевому устройству. AUTO - разрешен автоматический вход, NONE - запрещен
displayName	ФИО пользователя
contactEmail	Контактный email пользователя, указаный в контракте

- определение списка идентификаторов ресурсов защищаемых сервисов, которые могут быть использованы сервисом для данного пользователя. Часть методов может быть включена в список по умолчанию, остальные должны быть запрошены при аутентификации. Идентификатор метода может содержать ограничение на минимальный уровень авторизации пользователя. В этом случае соответствующий scope будет выдан только при наличии требуемого уровня авторизации.

Перечень доступных сервису scope и ограничения на минимальный уровень авторизации настраиваются администратором WebSSO.

scope - мнемоническое название ресурса защищаемого сервиса.

12. Диаграммы переходов для работы с WebSSO

12.1. Авторизация в WebSSO при существующей сессии

12.2. Аутентификация в WebSSO, отсутствие сессии, успешный автовход

12.3. Аутентификация и авторизация в WebSSO

12.4. Повышение уровня авторизации

12.5. Отзыв токена

13. Приложение 1. Базовые адреса конечных точек.

Раздел содержит значения для базовых адресов конечных точек, используемых в проекте.

Таблица 4. Базовые адреса конечных точек WebAPI управленя привязками

Окружение	Базовый адрес	Комментарий
vagrant	http://sso.ocb.hosted:8080/webapi-1.0	
интеграционное		
продуктив		