Интеграция с WebSSO по OAuth 1.0

Получение токена сервера (request token)

Для возможности аутентификации пользователя по протоколу OAuth 1.0 прежде всего необходимо получить токен сервера (request token). Токен сервера может быть получен только зарегистрированными в WebSSO и не заблокированными клиентами. Для получения токена сервера необходимо выполнить запрос к WebSSO, подписанный секретным ключом клиента, открытый и секретный ключи клиента выдаются при регистрации клиента в WebSSO.

Формат запроса

```
POST <sso_host>/sso/resources/1/oauth/get_request_token
Content-Type: application/x-www-form-urlencoded
Accept: application/json
Authorization: OAuth oauth_callback="<oauth_callback>",
    oauth_signature="<oauth_signature>",
    oauth_nonce="<oauth_nonce>",
    oauth_version="1.0",
    oauth_signature_method="HMAC-SHA1",
    oauth_consumer_key="<oauth_consumer_key>",
    oauth_timestamp="<oauth_timestamp>",
    realm="%2Fcustomer"
```

Параметры

- <sso_host> - базовый адрес сервера WebSSO, например sso.rooxteam.com, может содержать порт, например sso.rooxteam.com:8080
- Authorization - название заголовка в запросе
- <oauth_callback> - URL обратного редиректа, адрес страницы, на которую должен быть перенаправлен ответ, например https://sso.rooxteam.com/sso/secure/lk.jsp
- <oauth_signature> - цифровая подпись текущего запроса согласно спецификации RFC 5849 пункт 3.4
- <oauth_nonce> - случайный набор символов, генерируемый клиентом, должен быть уникален среди запросов с совпадающими параметрами oauth_timestamp и oauth_consumer_key
- <oauth_version> - версия протокола, всегда 1.0
- <oauth_signature_method> - алгоритм цифровой подписи, всегда HMAC-SHA1
- <oauth_consumer_key> - открытый ключ клиента, выдается при регистрации клиента в WebSSO
- <oauth_timestamp> - время запроса в формате Unix time - число секунд с 1 января 1970

00:00:00 GMT

- <realm> - группа пользователей WebSSO, всегда используется значение %2Fcustomer, которое является uri-encoded значением /customer

Формат успешного ответа

```
HTTP/1.1 200 OK

oauth_token=<oauth_token>&
oauth_token_secret=<oauth_token_secret>&
oauth_callback_confirmed=true
```

Параметры

- <oauth_token> - выданный токен сервера, URL-encoded, например, https%3A%2F%2Fsso.rooxteam.com%3A443%2Fsso%2Fresources%2F1%2Foauth%2Frtoken%2F18f5401a7f034e6caf7c301164c97c7b
- <oauth_token_secret> - выданный секретный ключ сервера, например, 9dc531be995c434eb11926e7142a3078
- <oauth_callback_confirmed> - параметр для отличия от предыдущих версий протокола OAuth 1.0, всегда true

TIP	полученный ключ oauth_token_secret, используется для цифровой подписи всех последующих запросов со стороны платформы.

Формат неуспешного ответа

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

```
{
  "code": 400,
  "message": "<message>"
}
```

Параметры

- <code> - код ошибки, повторяет HTTP статус ответа
- <message> - текстовое описание ошибки, например, Callback URL is missing., Signature invalid.

Аутентификация

Для выполнения аутентификации нужно направить пользователя на соответствующий адрес WebSSO с указанием полученного на предыдущем шаге токена сервера. WebSSO выполнит аутентификацию пользователя по логину-паролю или другому способу, если это необходимо, после чего выполнит обратный переход на адрес, указанный в запросе на получение токена сервера. К адресу обратного перехода WebSSO добавит параметры по которым можно будет получить токен доступа.

Формат запроса

```
GET
<sso_host>/sso/oauth/userconsole.jsp?oauth_token=<oauth_token>&logout_reason=<logout_r
eason>&return_strategy=<return_strategy>
Content-Type: application/x-www-form-urlencoded
```

Параметры

- <oauth_token> - значение токена сервера (request token), полученного после предыдущего запроса
- <logout_reason> - код причины разрыва сессии. **опционально**
- <return_strategy> - код стратегии возврата при блокировке. **опционально**

Формат успешного ответа

```
GET <redirect_url>?oauth_token=<oauth_token>&oauth_verifier=<oauth_verifier>
```

Параметры

- <redirect_url> - URL обратного редиректа, например https://sso.rooxteam.com/sso/secure/lk.jsp
- <oauth_token> - токен сервера из запроса, URL-encoded, например, https%3A%2F%2Fsso.rooxteam.com%3A443%2Fsso%2Fresources%2F1%2Foauth%2Frtoken%2F18f5401a7f034e6caf7c301164c97c7b
- <oauth_verifier> - проверочный код для получения токена доступа, например, 6e5aa56e040f44c49d29869bf526bd7d

Формат неуспешного ответа

Для пользователя может быть заблокирован конкретный ресурс по client_id. В этом случае вместо проверочного кода для получения токена доступа в GET-параметре будет описание ошибки.

```
GET <redirect_url>?error=<error>&error_description=<error_description>
```

Параметры

- <redirect_url> - URL обратного редиректа, например https://sso.rooxteam.com/sso/secure/lk.jsp
- <error> - код ошибки, например, 401
- <error_description> - текстовое описание ошибки, URL-encoded, например, Consumer%20is%20blocked%20for%20current%20resource%20owner.

Формат ответа при блокировке пользователя

IMPORTANT	В текущей реализации процесса аутентификации не поддерживается.

Пользователь может быть заблокирован. В этом случае вместо проверочного кода для получения токена доступа в GET-параметре будет описание ошибки.

```
GET
<redirect_url>?error=401&error_description=The%20authorization%20server%20can%20not%20
authorize%20the%20resource%20owner.&error_subtype=user_blocked&expires_in=3600
```

Параметры

- <redirect_url> - URL обратного редиректа, например https://sso.rooxteam.com/sso/secure/lk.jsp
- <error> - код ошибки, соответствующий HTTP статусу, например 401.
- <error_description> - текстовое описание ошибки, например The%20authorization%20server%20can%20not%20authorize%20the%20resource%20owner.
- <error_subtype> - статус токена, список возможных статусов приведен ниже
- <expires_in> - время до истечения срока блокировки (в секундах). Если параметр отсутствует, то учетная запись сможет быть разблокирована только администратором системы.

Формат ответа при блокировке OAuth 1.0 клиента

OAuth 1.0 клиент может быть заблокирован. В этом случае вместо проверочного кода для получения токена доступа в GET-параметре будет описание ошибки.

```
GET
<redirect_url>?error=401&error_description=The%20authorization%20server%20can%20not%20
authorize%20the%20resource%20owner.
```

Параметры

- <redirect_url> - URL обратного редиректа, например https://sso.rooxteam.com/sso/secure/lk.jsp
- <error> - код ошибки, соответствующий HTTP статусу, например 401.
- <error_description> - текстовое описание ошибки, например The%20authorization%20server%20can%20not%20authorize%20the%20resource%20owner.

Получение токена доступа (access token)

Для получения данных пользователя и возможности выполнять действия от его имени защищаемый сервис должен получить access token. Для этого нужно выполнить запрос на соответствующий URL WebSSO с указанием данных клиента и кода, полученного после аутентификации. Результатом будет новый токен, либо сообщение об ошибке.

Формат запроса

```
POST <sso_host>/sso/resources/1/oauth/get_access_token
Content-Type: application/x-www-form-urlencoded
Accept: application/json
Authorization: OAuth oauth_signature="<oauth_signature>",
    oauth_nonce="<oauth_nonce>",
    oauth_signature_method="HMAC-SHA1",
    oauth_consumer_key="<oauth_consumer_key>",
    oauth_token="<oauth_token>",
    oauth_verifier="<oauth_verifier>",
    oauth_timestamp="<oauth_timestamp>"
```

Параметры

- <sso_host> - базовый адрес сервера WebSSO, например sso.rooxteam.com, может содержать порт, например sso.rooxteam.com:8080
- Authorization - название заголовка в запросе
- <oauth_signature> - цифровая подпись текущего запроса согласно спецификации RFC 5849 пункт 3.4
- <oauth_nonce> - случайный набор символов, генерируемый клиентом, должен быть уникален среди запросов с совпадающими параметрами oauth_timestamp и oauth_consumer_key
- <oauth_signature_method> - алгоритм цифровой подписи, всегда HMAC-SHA1
- <oauth_consumer_key> - открытый ключ клиента, выдается при регистрации клиента в WebSSO
- <oauth_token> - токен сервера (request token)
- <oauth_verifier> - проверочный код, полученный после аутентификации

- <oauth_timestamp> - время запроса в формате Unix time - число секунд с 1 января 1970 00:00:00 GMT

Формат успешного ответа

```
HTTP/1.1 200 OK

oauth_token=<oauth_token>&oauth_token_secret=<oauth_token_secret>
```

Параметры

- <oauth_token> - токен доступа (access token), например, https://sso.rooxteam.com:443/sso/resources/1/oauth/atoken/3c3e6d4cec214b0b8f4baee5b030e501
- <oauth_token_secret> - ключ для подписи запросов, например, 8588e4d7a8cc4c17a651691a7934d001

Формат неуспешного ответа

```
HTTP/1.1 401 Unauthorized
```

```
{
  "code": 401,
  "message": "<message>"
}
```

Параметры

- <code> - код ошибки, повторяет HTTP статус ответа
- <message> - текстовое описание ошибки, например, Request token invalid., Signature invalid.

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.

Формат запроса

```
POST <sso_host>/sso/oauth-status
Content-Type: application/x-www-form-urlencoded
Accept: application/json
Authorization: OAuth oauth_signature="<oauth_signature>",
    oauth_nonce="<oauth_nonce>",
    oauth_signature_method="HMAC-SHA1",
    oauth_consumer_key="<oauth_consumer_key>",
    oauth_token="<oauth_token>",
    oauth_timestamp="<oauth_timestamp>"
```

- <sso_host> - базовый адрес сервера WebSSO, например sso.rooxteam.com, может содержать порт, например sso.rooxteam.com:8080
- Authorization - название заголовка в запросе

Параметры

- <oauth_signature> - цифровая подпись текущего запроса согласно спецификации RFC 5849 пункт 3.4
- <oauth_nonce> - случайный набор символов, генерируемый клиентом, должен быть уникален среди запросов с совпадающими параметрами oauth_timestamp и oauth_consumer_key
- <oauth_signature_method> - алгоритм цифровой подписи, всегда HMAC-SHA1
- <oauth_consumer_key> - открытый ключ клиента, выдается при регистрации клиента в WebSSO
- <oauth_token> - токен доступа (access token), полученный на предыдущем шаге
- <oauth_timestamp> - время запроса в формате Unix time - число секунд с 1 января 1970 00:00:00 GMT

Формат успешного ответа

```
HTTP/1.1 200 OK
```

```
{
  "resources": {
    "BAL": 1,
    "SUB": 1,
    "MSISDN": 1
  },
  "msisdn": "79876543210",
  "resultDetails": "",
  "result": 200,
  "client_id": "selfcare"
}
```

Параметры

- <resources> - список scope - разрешенных операций для пользователя
- <msisdn> - номер телефона пользователя
- <resultDetails> - дополнительное описание ответа
- <result> - статус ответа, повторяет HTTP статус
- <client_id> - идентификатор OAuth 1.0 клиента, которому был выдан токен

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": {
    "code": 401,
    "message": "Access token is invalid."
  }
}
```

Параметры

- <code> - код ошибки, повторяет HTTP статус ответа
- <message> - текстовое описание ошибки, например, Access token is invalid., Signature is invalid.

Формат ответа при блокировке OAuth 1.0 клиента

```
HTTP/1.1 403 Forbidden
```

```
{
  "error": {
    "code": 403,
    "message": "Client is blocked."
  }
}
```

Параметры

- <code> - код ошибки, повторяет HTTP статус ответа

- <message> - текстовое описание ошибки, например, OAuth client is blocked.

Глобальный выход из WebSSO по ссылке

Глобальный выход из WebSSO может быть осуществлен переходом аутентифицированного пользователя на URL <sso_host>/sso/UI/Logout. При этом WebSSO завершит текущую сессию пользователя и удалит все выданные в рамках данной сессии токены. Желательно добавить в URL параметр goto с адресом обратного перехода, на этот адрес пользователь будет перенаправлен сразу после выхода.

Формат запроса

```
GET <sso_host>/sso/UI/Logout?goto=<goto>
```

- <sso_host> - базовый адрес сервера WebSSO, например sso.rooxteam.com, может содержать порт, например sso.rooxteam.com:8080
- <goto> - URL обратного редиректа, URL-encoded, например https%3A%2F%2Fsso.rooxteam.com%2Fsso%2Findex.html для адреса https://sso.rooxteam.com/index.html

Формат успешного ответа

Пользователь будет перенаправлен на адрес, указанный в параметре goto

Формат неуспешного ответа

В случае ошибки выхода из WebSSO пользователь будет перенаправлен на адрес, указанный в параметре goto, при этом будут добавлены GET-параметры с кодом и описанием ошибки.

```
GET <goto>?error=<error>&error_description=<error_description>
```

Параметры

- <goto> - URL обратного редиректа, указанный в запросе
- <error> - код ошибки, например internal_error
- <error_description> - текстовое описание ошибки, например Cannot%access%20session%20store.

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.

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>
```

Параметры

- <sso_host> - базовый адрес сервера WebSSO, например sso.rooxteam.com, может содержать порт, например sso.rooxteam.com:8080
- <token> - токен для инвалидации, например https://sso.rooxteam.com:443/sso/resources/1/oauth/atoken/3c3e6d4cec214b0b8f4baee5b030e501
- <token_type_hint> - тип токена, всегда access_token

Формат успешного ответа

```
HTTP/1.1 200 OK
```

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_description - текстовое описание ошибки
- error - код ошибки согласно спецификации OAuth 2.0 Token Revocation RFC 7009 пункт 4.1.1

Кеширование токенов

Если защищаемый сервис находится под общим доменом второго уровня с WebSSO, то он может использовать кеширование результатов валидации токенов, чтобы не делать запрос на валидацию токена при каждом действии пользователя. В этом случае сервис должен отслеживать значение cookie sh, как только значение cookie изменится, необходимо сбросить кеш.

Если защищаемый сервис не находится под доменом второго уровня WebSSO, кешировать результаты валидации токенов запрещено, необходимо всегда делать запрос валидации в соответствии с пунктом Валидация токена доступа.

Подробные требования к кешированию результатов авторизации описаны в документе "SSO Auth Model" пункт 4.3

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.

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.

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
- event - имя события
- global - флаг полной блокировки сервиса, в этом оповещении всегда false
- cn - номер телефона пользователя (msisdn)
- access_token - инвалидированный access token

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.

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
- event - имя события
- global - флаг полной блокировки сервиса, в этом оповещении всегда true

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.

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
- event - имя события
- global - флаг полной блокировки сервиса, в этом оповещении всегда false
- cn - номер телефона пользователя (msisdn)
- access_token - измененный access token

Справочник возможных причин разрыва сессии

Значения параметра logout_reason из пункта Аутентификация приведены в следующей таблице

код	описание
idle_timeout	Бездействие пользователя в течение определенного времени
session_timeout	Истечение времени сессии

Справочник возможных стратегий возврата при блокировке

Значения параметра return_strategy из пункта Аутентификация приведены в следующей таблице

код	описание
off	Отображение ошибки на виджете логина без перехода на redirect_uri. Используется по умолчанию
auto	Автоматический переход на redirect_uri при блокировке пользователя

код	описание
manual	Отображение в виджете кнопки "Назад" при блокировке пользователя, переход переход на redirect_uri при нажатии этой кнопки

Справочник возможных статусов токена

код	описание
revoked	Пользователь аннулировал токен
user_blocked	Пользователь заблокирован

Диаграммы переходов для работы с WebSSO

Аутентификация и авторизация в WebSSO

```
@startuml
' Generic Attributes
title Аутентификация и авторизация в WebSSO
autonumber
hide footbox

' Actors
actor Пользователь as customer #3F9C35
participant "Веб-портал" as wp #7B7BC0
participant "WebSSO Server \n (основное серверное приложение WebSSO)" as sso #F78F1E
participant "Audit Storage" as audit #F78F1E

' Use Case
customer  -> wp : запрашивает защищенный ресурс
             wp  -> sso : запрашивает токен сервера
             wp <-- sso : возвращает токен сервера
customer <-- wp : перенаправляет на логин в WebSSO
customer  -> sso : запрашивает страницу логина WebSSO
customer <-- sso : возвращает страницу логина
customer  -> sso : отправляет логин+пароль
             sso -> sso : проверяет логин+пароль
             sso -> audit : протоколирует факт аутентификации
             sso -> sso : генерирует код авторизации
customer <-- sso : перенаправляет на веб-портал с кодом авторизации
customer  -> wp : запрашивает защищенный ресурс, передает код авторизации
             wp  -> sso : запрашивает токен доступа по коду авторизации
                    sso -> sso : создает токен
             wp <-- sso : возвращает токен
             wp  -> sso : валидирует токен\n обращением на интерфейс валидации
             wp <-- sso : возвращает результат валидации
             wp  -> wp : инициирует собственную сессию\n на основе данных из токена
WebSSO
customer <-- wp : устанавливает собственный сессионный токен,\n возвращает защищенный
ресурс
customer  -> wp : дальнейшие запросы авторизуются\n на основании токена Веб-портала

@enduml
```

Отзыв токена

```
@startuml
' Generic Attributes
title Отзыв токена через API
autonumber
hide footbox

' Actors
actor Пользователь as customer #3F9C35
participant "Веб-портал" as wp #7B7BC0
participant "WebSSO Server \n (основное серверное приложение WebSSO)" as sso #F78F1E

' Use Case
customer -> wp : выполняет Ajax-запрос на отзыв токена
            wp  -> sso: передает access_token для инвалидации
                   sso -> sso: удаляет токен

@enduml
```

Logout через URL

```
@startuml
' Generic Attributes
title Logout через URL
autonumber
hide footbox

' Actors
actor Пользователь as customer #3F9C35
participant "WebSSO Server \n (основное серверное приложение WebSSO)" as sso #F78F1E
participant "Веб-портал" as wp #7B7BC0

' Use Case
customer  -> sso : выполняет запрос на logout
             sso -> sso: удаляет внутреннюю сессию пользователя и все привязанные к
ней токены
             sso -> sso: сбрасывает cookie "sl", чтобы клиенты под тем же доменом
инвалидировали кеш.
             sso -> wp: отправляет уведомление на callback URL
                    wp -> wp: инвалидирует сессию пользователя
customer <-- sso: перенаправляет на страницу веб-портала
customer  -> wp: запрашивает защищенный ресурс
             wp  -> wp: проверяет cookie "sl", сбрасывает кеш
             wp  -> sso: проверяет токен
             wp <-- sso: отвечает, что токен не действителен
customer <-- wp: отказывает в доступе к защищенному ресурсу

@enduml
```

Связанные документы

Название документа	Описание
rfc5849.pdf	базовая спецификация протокола OAuth1.0.
rfc7009.pdf	расширение базовой спецификации протокола OAuth2.0, описывающее token revocation.