# Интеграция с WebSSO

Оглавление



---

Версия продукта: sso-server-ocb-{project-version}

# 1. История изменений

| Версия документа | Дата изменения | Комментарий |
|---|---|---|
| v1.0.0 | 2015-06-30 | Создание документа |
| v1.0.1 | 2015-07-10 | Изменения сценария revocation. |
| v1.0.2 | 2015-07-28 | Добавлено подробное описание авторизации доступа. |
| v1.0.3 | 2015-08-04 | Актуализация документа |
| v1.0.4 | 2015-08-19 | Добавлено предупреждение о возврате 503 ошибки в виде HTML. |
| v1.0.5 | 2015-08-20 | Добавлены заголовки Accept в запросах к API. |
| v1.0.6 | 2015-09-21 | Добавлено приложение с конечными точками WebSSO |
| v1.0.7 | 2016-04-08 | Добавлены рекомендации по мониторингу доступности WebSSO |

## # 2. Соглашения и терминология

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 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 минимальному. Подробнее об уровнях авторизации см. документ "SSO Auth Model".

токен - access token

access token - строковый идентификатор, предоставляющий внешнему сервису право выполнять действия от лица пользователя

идентификатор клиента - строка, идентифицирующая внешний сервис. Конфигурируется администратором SSO

scope - атрибут access token, определяющий либо набор данных пользователя, к которому получает доступ внешний сервис, либо 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 - HTTP URL с заменой зарезервированных URI символов на их представление в виде %XX

# 3. Введение

## 3.1. Назначение документа

Документ предназначен для разработчиков сервисов, интегрируемых с WebSSO, и предоставляет методические рекомендации по интеграции.

# 4. Общее описание WebSSO

WebSSO обеспечивает выполнение аутентификации абонентов МегаФон с авторизацией доступов абонентов к защищенным ресурсам сети МегаФон.

## 4.1. Общая последовательность действий при интеграции.

Процесс подключения сервиса к WebSSO описан в разделе 2.6. документа "Описание жизненного цикла внедрения SSO" (SSO Life Cycle)

### 4.1.1. Общая схема взаимодействия

Общая схема интеграции WebSSO и внешней системы:

1. Пользователь нажимает на странице системы поставщика услуг кнопку «Войти через МегаФон».
2. WebSSO аутентифицирует абонента.
3. WebSSO перенаправляет абонента на подключаемый ресурс и выдает access token.
4. Подключаемая система посылает запрос на получение пользовательской информации по access token.
5. WebSSO валидирует access token и возвращает пользовательскую информацию.

Сценарии взаимодействия с WebSSO более подробно рассмотрены в документе oauth2-integration.pdf

# 5. Общие положения

- При запросах к API, ошибки со статусом 503 всегда приходят в HTML.

# # 6. Аутентификация пользователей через WebSSO

## 6.1. Описание сценария аутентификации

### 6.1.1. Подготовка запроса на аутентификацию

Построив запрос аутентификации, сервис перенаправляет абонента в WebSSO для прохождения процесса аутентификации.

```
GET https://sso.megafon.ru/oauth2/authorize?
realm=%2Fcustomer&response_type=code&client_id=client&service=external&redirect_uri=https:
//client.example.org&scope=cn%20networkAuthenticationType
```

- realm - группа пользователей WebSSO, в качестве значения всегда используется %2Fcustomer (URL-encoded /customer)
- response_type - способ авторизации пользователя, всегда используется значение code
- client_id - идентификатор клиента
- service - механизм аутентификации в SSO, всегда используется значение external
- redirect_uri - URL обратного редиректа, URL-encoded. Адрес должен быть в списке разрешенных в WebSSO, список разрешенных адресов конфигурируется администратором WebSSO из опросника в AUTHN.16
- scope - список запрашиваемых scope через пробел (список разрешенных значений конфигурируется администратором WebSSO из опросника в AZSSO.1.*, AZRES.*), если параметр не передан, то будут использоваться предопределеный набор scope (настраивается в соответствии с AUTHZ.16 и AUTHZ.17 пунктами опросника)

ЗАМЕТКА

Нужно обратить внимание на то, что WebSSO позволяет задать перечень scope, которые будут выдаваться автоматически каждому пользователю, успешно прошедшему аутентификацию (пункт AUTHZ.17 опросника). Тогда параметер scope можно не передавать.

### 6.1.2. Аутентификация и авторизация пользователя

WebSSO взаимодействует с абонентом посредством прохождения процесса аутентификации с помощью ввода логина и пароля, если не осталось действующей сессии. При повышении уровня авторизации WebSSO может потребовать от абонента ввести код полученный с помощью SMS (в разделе повышения). После завершения процесса аутентификации абонент будет перенаправлен на адрес указанный в redirect_uri с успешным или неуспешным результатом.

#### Выдача авторизационного кода

Если пользователь прошел аутентификацию, то WebSSO направит пользователя по redirect_uri, добавив code:

```
https://client.example.org/?code=1d601e9a-992d-44a7-be21-bca07fbc762c
```

#### Ошибка аутентификации или недостаток прав

Если пользователю не удалось пройти процесс аутентификации, то WebSSO вернет пользователя в подключаемый сервис по redirect_uri с неуспешным результатом (с добавленным параметром error):

НАПОМИНАЕМ

Полный список кодов приведен в документе oauth2-integration.pdf

```
https://client.example.org/?
error=access_denied&error_description=The%20authorization%20server%20can%20not%20authorize
%20the%20resource%20owner.
```

#### Пользователь заблокирован

Если пользователь заблокирован, то в URL обратного редиректа будут добавлены параметры error, error_subtype с указанием блокировки пользователя и expires_in когда он будет разблокирован, если такая информация доступна:

```
https://client.example.org/?
error=access_denied&error_description=The%20authorization%20server%20can%20not%20authorize
%20the%20resource%20owner.&error_subtype=user_blocked&expires_in=3600
```

- error_subtype - статус токена, список возможных статусов приведен в документе oauth2-integration.pdf
- expires_in - время до истечения срока блокировки (в секундах). Если параметр отсутствует, то учетная запись сможет быть разблокирована только администратором системы.

#### 5.1.2.2.2 Подключаемый сервис заблокирован

Если сервис заблокирован, то WebSSO добавит в redirect URI параметры error, error_subtype с указанием блокировки сервиса и параметр expires_in, определяющий промежуток времени, через который сервис будет разблокирован, если такая информация доступна:

```
https://client.example.org/?
error=access_denied&error_description=The%20authorization%20server%20can%20not%20authorize
%20the%20resource%20owner.&error_subtype=system_blocked&expires_in=3600
```

- error_subtype - статус токена, список возможных статусов приведен в документе oauth2-integration.pdf
- expires_in - время до истечения срока блокировки (в секундах). Если параметр отсутствует, то подключаемая система может быть разблокирована только администратором системы.

### 6.1.3. Получение access токена

Для получения данных пользователя и возможности выполнять действия от его имени, подключаемый сервис должен получить access token. Для этого нужно выполнить POST запрос на соответствующий URL WebSSO с указанием данных клиента и кода, полученного в результате аутентификации.

Посылаемые параметры должны передаваться в теле HTTP-запроса с использованием типа сообщения "application/x-www-form-urlencoded", как это описано в HTML спецификации W3C.REC-html401-19991224

```
POST /sso/oauth2/access_token
Host: sso.megafon.ru
Content-Type: application/x-www-form-urlencoded
Accept: application/json

realm=%2Fcustomer&client_id=client&client_secret=password&redirect_uri=https://client.example.org&grant_type=authorization_code&code=1d601e9a-992d-44a7-be21-bca07fbc762c
```

- realm - группа пользователей WebSSO, всегда используется значение %2Fcustomer, которое является URL-encoded значением /customer
- client_id - идентификатор клиента, например client
- client_secret - пароль клиента, возможные значения зависят от конфигурации
- redirect_uri - URL обратного редиректа, который был указан при аутентификации, URL-encoded
- grant_type - способ авторизации пользователя, всегда используется значение authorization_code
- code - код авторизации, полученный в результате аутентификации, например 1d601e9a-992d-44a7-be21-bca07fbc762c

### Подключаемый сервис получает access token

Сервис получает ответ с HTTP кодом 200 "OK" от WebSSO со следующими парамерами:

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

Подключаемый сервис может использовать access token для доступа к защищённым ресурсам от имени абонента.

Пример ответа:

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

```
{
  "scope": [
    "cn",
    "networkAuthenticationType"
  ],
  "expires_in": 59,
  "token_type": "Bearer",
  "access_token": "7bdaeacc-3d80-415c-920f-a7c30ca5e743"
}
```

### Пользователь заблокирован

Сервис получает ответ с HTTP кодом 400 "Bad Request" от WebSSO со следующими парамерами:

- error - код ошибки согласно спецификации OAuth 2.0 RFC 6749 пункт 5.2

- error_description - текстовое описание ошибки
- error_subtype - статус токена, список возможных статусов приведен в документе oauth2-integration.pdf
- expires_in - время до истечения срока блокировки (в секундах). Если параметер отсутствует, то учетная запись сможет быть разблокирована только администратором системы.

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

```
{
  "error": "invalid_grant",
  "error_description": "User is blocked.",
  "error_subtype": "user_blocked",
  "expires_in": 3600
}
```

### Подключаемый сервис заблокирован

Сервис получает ответ с HTTP кодом 400 "Bad Request" от WebSSO со следующими парамерами:

- error - код ошибки согласно спецификации OAuth 2.0 RFC 6749 пункт 5.2
- error_description - текстовое описание ошибки
- error_subtype - статус токена, список возможных статусов приведен в документе oauth2-integration.pdf
- expires_in - время до истечения срока блокировки (в секундах). Если параметер отсутствует, то учетная запись сможет быть разблокирована только администратором системы.

```
[source,json]
HTTP/1.1 400 Bad Request

[source,json]
```

{ "error": "invalid_client", "error_description": "OAuth 2 client is blocked.", "error_subtype": "system_blocked", "expires_in": 3600 }

* +error_subtype+ - статус токена, список возможных статусов приведен в документе _oauth2-integration.pdf_
* +expires_in+ - время до истечения срока блокировки (в секундах). Если параметер отсутствует, то учетная запись сможет быть разблокирована только администратором системы.

===== Возможные ошибки
====== 5.1.3.4.1. Невалидный код или код с истекшим сроком действия

Сервис получает ответ с HTTP кодом 400 "Bad Request" от WebSSO со следующими парамерами:

* +error+ со значением +invalid_grant+
* +error_description+

[source,http]

HTTP/1.1 400 Bad Request

[source,json]

{ "error_description": "The provided access grant is invalid, expired, or revoked.", "error": "invalid_grant" }

====== 5.1.3.4.2. Ошибка аутентификации клиента - неверный идентификатор или пароль клиента

Сервис получает ответ с HTTP кодом 400 "Bad Request" от WebSSO со следующими парамерами:

* +error+ со значением +invalid_client+
* +error_description+

[source,http]

HTTP/1.1 400 Bad Request

[source,json]

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

====== 5.1.3.4.3. Переданный redirect_uri не совпадает с указанным при аутентификации

Сервис получает ответ с HTTP кодом 400 "Bad Request" от WebSSO со следующими парамерами:

* +error+ со значением +redirect_uri_mismatch+
* +error_description+

[source,http]

HTTP/1.1 400 Bad Request

[source,json]

{ "error_description": "The redirection URI provided does not match a pre-registered value.", "error": "redirect_uri_mismatch" }

====== 5.1.3.4.4. Указанный способ авторизации не поддерживается.

NOTE: Не должна возникать при использовании способа +authorization_code+

Сервис получает ответ с HTTP кодом 400 "Bad Request" от WebSSO со следующими парамерами:

* +error+ со значением +unsupported_grant_type+
* +error_description+

[source,http]

HTTP/1.1 400 Bad Request

[source,json]

{ "error_description": "Grant type is not supported: authorization_token", "error": "unsupported_grant_type" }

===== Описание параметров ответа WebSSO
* +error+ - код ошибки согласно спецификации OAuth 2.0 RFC 6749 пункт 5.2
* +error_description+ - текстовое описание ошибки
* +error_subtype+ - подтип ошибки, список возможных статусов приведен в документе _oauth2-integration.pdf_
* +expires_in+ - время до истечения срока блокировки (в секундах). Если параметр отсутствует, то учетная запись сможет быть разблокирована только администратором системы.

<<<
== Получение пользовательских данных

Подключаемый сервис при регистрации в системе WebSSO сообщает список атрибутов пользователя, которые ему необходимы.
Согласно этому списку атрибутов в системе WebSSO разрешается доступ к запрошенным _scope_.
Внешний сервис может использовать в качестве значения параметра _scope_ значения из опросника (список разрешенных scope конфигурируется администратором
WebSSO из опросника в AZSSO.1.*, AZRES.*) в запросах на получение пользовательских данных.
В случае, если _scope_ разрешен системой WebSSO и пользователь дал согласие на предоставление данных подключаемой системе,
WebSSO вернет значения атрибутов пользователя при успешной валидации токена.

=== Получение пользовательских данных

[source,http]

POST /sso/oauth2/tokeninfo?access_token=7bdaeacc-3d80-415c-920f-a7c30ca5e743 Host: sso.megafon.ru Content-Type: application/json Accept: application/json

* +access_token+ - полученный токен, например +7bdaeacc-3d80-415c-920f-a7c30ca5e743+

=== Подключаемый сервис получает пользовательские данные
[source,http]

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

[source,json]

{ "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": "client" }

Обязательные поля:

* +scope+ - список scope (в формате JSON Array) разрешенных для использования от имени пользователя
* +realm+ - группа пользователей WebSSO, всегда возвращается значение +/customer+
* +token_type+ - тип выданного токена. Всегда +Bearer+
* +expires_in+ - время до истечения срока действия токена в секундах
* +access_token+ - проверямый access token
* +auth_level+ - выданный уровень авторизации пользователя
* +client_id+ - идентификатор клиента, которому выдан токен

TIP: Подключаемая система обязательно должна проверять _expires_in_ (значение должно быть больше нуля), остальные поля являются информационными.

=== Невалидный токен

Если передан невалидный или токен с истекшим временем действия, будет возвращен следующий ответ:
[source,http]

HTTP/1.1 401 Unauthorized

[source,json]

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

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

=== Пользователь заблокирован

Сервис получает ответ с HTTP кодом 403 "Forbidden" от WebSSO со следующими парамерами:

* +error+
* +error_description+
* +error_subtype+
* +expires_in.+

[source,http]

HTTP/1.1 403 Forbidden

[source,json]

{ "error_subtype": "user_blocked", "expires_in": 3600, "access_token": "5fdfeafd-3061-4b1c-9076-0fe460f91fc8" }

* +error_subtype+ - статус токена, <<error_subtype, список возможных статусов приведен ниже>>
* +access_token+ - проверямый access token
* +expires_in+ - время до истечения срока блокировки (в секундах). Если параметер отсутствует, то учетная запись сможет быть разблокирована только администратором системы.

=== Подключаемый сервис заблокирован

Сервис получает ответ с HTTP кодом 403 "Forbidden" от WebSSO с парамерами error, error_description, error_subtype и expires_in.

[source,http]

HTTP/1.1 403 Forbidden

[source,json]

{ "error_subtype": "system_blocked", "expires_in": 3600, "access_token": "5fdfeafd-3061-4b1c-9076-0fe460f91fc8" }

* +error_subtype+ - статус токена, <<error_subtype,список возможных статусов приведен ниже>>
* +access_token+ - проверямый access token
* +expires_in+ - время до истечения срока блокировки (в секундах). Если параметер отсутствует, то учетная запись сможет быть разблокирована только администратором системы.

<<<
== Авторизация доступа к своим ресурсам

Подключаемый сервис должен проверять _access token_ на валидность и авторизовать запрос на WebSSO перед тем как
предоставлять доступ к своим защищенным ресурсам.

Подключаемый сервис не должен самостоятельно проверять наличие запрашиваемого scope среди разрешенных.
Проверка разрешения на доступ осуществляется вызовом метода WebSSO для авторизации доступа (tokeninfo).

=== Запрос на авторизацию доступа
[source,http]

POST /sso/oauth2/tokeninfo?access_token=7bdaeacc-3d80-415c-920f-a7c30ca5e743&scope=cn%20networkAuthenticationType Host: sso.megafon.ru Content-Type: application/json Accept: application/json

[source,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-MegaFon-Net": [ "1" ], "X-Nokia-MSISDN": [ "9876543210" ], "X-Forwarded-For": [ "10.20.30.40", "10.10.35.46", "192.168.12.74" ] } }

* +access_token+ - полученный access token, например +7bdaeacc-3d80-415c-920f-a7c30ca5e743+
* +scope+ - идентификатор ресурса защищаемого сервиса, подробное описание находится в документе oauth2-integration.pdf
* +httpMethod+ - HTTP метод для которого запрашивается доступ
* +url+ - URL для которого запрашивается доступ
* +headers+ - список HTTP заголовков из запроса пользователя, передаются "как есть" без дополнительной обработки или фильтрации

TIP: Подключаемая система обязательно должна проверять _expires_in_ (значение должно быть больше нуля), остальные поля являются информационными.

TIP: Если подлючаемый 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.
Подробности реализации представлены в документе SSO Auth Model.

=== Подключаемый сервис получает разрешение

[source,http]

HTTP/1.1 200 OK

[source,json]

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

* +scope+ - список scope (в формате JSON Array) разрешенных для использования от имени пользователя, возможные значения задаются конфигурацией
* +realm+ - группа пользователей WebSSO, всегда возвращается значение +/customer+
* +token_type+ - тип выданного токена. Всегда +Bearer+
* +expires_in+ - время до истечения срока действия токена в секундах
* +access_token+ - проверямый access token
* +auth_level+ - выданный уровень авторизации пользователя
* +client_id+ - идентификатор клиента, которому выдан токен, возможные значения зависят от конфигурации

=== Подключаемый сервис передал невалидный access token

Если передан невалидный токен или токен с истекшим сроком действия, будет возвращен ответ:

[source,http]

HTTP/1.1 401 Unauthorized

[source,json]

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

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

=== Подключаемый сервис не получает разрешение

[source,http]

HTTP/1.1 403 Forbidden

[source,json]

{ "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" }

* +scope+ - список scope (в формате JSON Array) разрешенных для использования от имени пользователя, возможные значения задаются конфигурацией
* +realm+ - группа пользователей WebSSO, всегда возвращается значение +/customer+
* +token_type+ - тип выданного токена. Всегда +Bearer+
* +expires_in+ - время до истечения срока действия токена в секундах
* +access_token+ - проверямый access token
* +auth_level+ - выданный уровень авторизации пользователя
* +required_auth_level+ - требуемый уровень авторизации для данного +scope+
* +client_id+ - идентификатор клиента, которому выдан токен, возможные значения зависят от конфигурации

==== Подключаемый сервис заблокирован

[source,http]

HTTP/1.1 401 Forbidden

[source,json]

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

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

<<<
== Обращение к подключенным ресурс серверам

Для выполнения бизнес-сценариев система *может* обращаться к другим ресурс серверам, защищенным WebSSO МегаФон.
Технология единого входа позволяет обращаться к таким ресурсам доверенным образом на основании пользовательского токена доступа.
Описание протокола доступа выходит за рамки ответственности WebSSO и нужно уточнять у разработчиков ресурс серверов, регламентируются только следующие требования:

a. *требуется* передавать токен доступа
b. *требуется* передавать оригинальный HTTP-контекст запроса клиента (заголовки запроса, путь, метод).

<<<
== Повышение уровня авторизации

=== Введение

Повышеный уровень авторизации требуется для доступа к платным защищенным ресурсам или персональным данных абонента.

=== Требования к повышению уровня авторизации
1. Повышение уровня авторизации может быть выполнено только после успешной аутентификации в WebSSO
2. Для повышения уровня авторизации требуется получить новый _access token_ с повышенными привилегиями. Ранее полученный access token продолжит действовать независимо от результата повышения уровня авторизации.

=== Сценарий повышения уровня авторизации
==== Подготовка запроса на повышение уровня авторизации
Построив запрос на повышение уровня авторизации, сервис перенаправляет абонента в WebSSO для прохождения процесса повышения уровня авторизации.

[source,http]

GET https://sso.megafon.ru/oauth2/authorize?realm=%2Fcustomer&response_type=code&client_id=client&service=external&redirect_uri=https://client.example.org&scope=cn%20networkAuthenticationType&auth_level=5

* +auth_level+ - желаемый уровень авторизации для текущего пользователя, *параметер обязателен*
* +realm+ - группа пользователей WebSSO, всегда используется значение +%2Fcustomer+, которое является uri-encoded значением +/customer+
* +response_type+ - способ авторизации пользователя, всегда используется значение +code+ (см. https://tools.ietf.org/html/rfc6749#page-19[RFC 6749])
* +client_id+ - идентификатор клиента (см. https://tools.ietf.org/html/rfc6749#section-2.3.1[RFC 6749])
* +service+ - механизм аутентификации в SSO, всегда используется значение +external+
* +redirect_uri+ - URL обратного редиректа, URL-encoded. Адрес должен быть в списке разрешенных в WebSSO, список разрешенных адресов конфигурируется администратором WebSSO из опросника в AUTHN.16
* +scope+ - URL-encoded список запрашиваемых scope, разделенных пробелами (список разрешенных scope конфигурируется администратором WebSSO из опросника в AZSSO.1.*, AZRES.*)

==== Получение кода повышенного уровня авторизации
WebSSO взаимодействует с абонентом для повышения уровня авторизации, например по одноразовому SMS-коду.
После завершения процесса аутентификации, WebSSO возвращает абонента с успешным или неуспешным результатом.

NOTE: WebSSO может не проводить проверку, если по ранее выданному токену. Т.е. если токен был выдан недавно,
то уровень авторизации пользователя будет повышен без дополнительных взаимодействий с ним.

===== Выдача нового авторизационного кода
Если процесс повышения уровня авторизации прошел успешно, то WebSSO добавит новый code повышенного уровня авторизации в redirect URI.

[source]

https://client.example.org/?code=1d601e9a-992d-44a7-be21-bca07fbc762c

===== Ошибка аутентификации или недостаток прав
Если пользователю не удалось пройти процесс повышения уровня авторизации, то WebSSO вернет пользователя в подключаемый сервис с неуспешным результатом.
[source]

https://client.example.org/?error=access_denied&error_description=The%20authorization%20server%20can%20not%20authorize%20the%20resource%20owner.

==== Получение access token для повышенного уровня авторизации
По выданному коду повышенного уровня авторизации можно получить токен повышенного уровня авторизации по протоколу OAuth 2 (см. документ _oauth2-integration.pdf_), что и для обычного _access token_

==== Использование токена повышенного уровня авторизации
_Access token_ с повышенным уровнем авторизации должен использоваться вместо обыного _access token_ при запросе
доступа к ресурсам, требующим повышенного уровня авторизации.

<<<
== Завершение сессии и инвалидация токенов

=== Инициирование завершения сессии и инвалидации токенов

Завершение сеанса на UIDM МОЖЕТ быть технически инициировано одним из двух способов:

1. Вызовом AJAX-эндпоинта
2. Переходом на URL выхода

IMPORTANT: Следует использовать AJAX, если нет технических ограничений.

==== Вызов AJAX-эндпойнта

CAUTION: Содержимое раздела может измениться

Выданный ранее токен может быть инвалидирован отправкой серверного запроса на соответствующий адрес UIDM.
Подключаемый сервис должен поддержать AJAX endpoint, выполняющий серверный запрос к UIDM.
Дополнительная аутентификация для этого метода не требуется, достаточно передать действующий _access token_.
Для этого нужно выполнить POST запрос на соответствующий URL UIDM с указанием _access token_ пользователя.

[source,http]

POST /sso/oauth2/revoke HTTP/1.1 Host: sso.uidm.ru Content-Type: application/x-www-form-urlencoded Accept: application/json

token=7bdaeacc-3d80-415c-920f-a7c30ca5e743&token_type_hint=access_token

* +‹token›+ — токен для инвалидации
* +‹token_type_hint›+ — тип токена, всегда +access_token+

===== Успешное завершение сессии
В случае успешной инвалидации пользовательской сессии, UIDM вернет следующий ответ с пустым телом:

[source,http]

HTTP/1.1 200 OK

===== Неуспешное завершение сессии

В случае ошибки инвалидации пользовательской сессии, UIDM сообщит об ошибке в теле ответа с параметрами:

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

[source,http]

HTTP/1.1 400 Bad Request

[source,json]

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

==== Вызвать сценарий завершения сессии через редирект

Завершение сессии из UIDM может быть осуществлено переходом аутентифицированного пользователя на URL выхода.
При этом UIDM завершит текущую сесиию 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.

[source]

https://sso.uidm.ru/UI/Logout?goto=https://client.example.org

===== Завершение сессии

Пользователь будет перенаправлен на адрес, указанный в параметре +goto+

=== Вызов callback-URL при инвалидации токенов

При инвалидации токена клиента UIDM может выполнить запрос на URL подключаемого сервиса с оповещением о факте инвалидации.
Список URL для оповещения регистрируется в опроснике в AUTHZ.15 и настраивается администратором UIDM при подключении сервиса. Внешняя система, получив такое оповещение, должна немедленно инвалидировать локальную сессию пользователя.

IMPORTANT: Недопустимо удаление атрибутов из локальной http-сессии вместо полной ее инвалидации.

<<<
== Требования к реализации ПО

=== Общие требования

НЕОБХОДИМО синхронизировать системные часы ресурс сервера со службой точного времени NTP.

НЕДОПУСТИМО передавать секретный ключ ресурс сервера в сторонние IT-системы кроме WebSSO в запросах на /token.

НЕДОПУСТИМО протоколировать секретный ключ ресурс сервиса.

НЕДОПУСТИМО передавать токен доступа в любые другие IT-системы кроме WebSSO и другие Resource Server.

НЕДОПУСТИМО протоколировать токен в немаскированном виде.

НЕДОПУСТИМО отключать проверку SSL-сертификатов при любых обращениях к WebSSO.

Система МОЖЕТ поддерживать другие способы аутентификации кроме входа через WebSSO. В таких случаях ТРЕБУЕТСЯ указание этого факта в спецификации интеграции. Система проверяет доступ к своим ресурсам самостоятельно.
При аутентификации через WebSSO НЕДОПУСТИМО запрашивать персональные данные пользователей (в том числе логин, пароль) на страницах ресурс сервера.

=== Рекомендации по мониторингу

Рекомендуется реализовать мониторинг доступности WebSSO непосредственно с серверов приложений. Для этого можно регулярно выполнять обращения на

‹базовый адрес WebSSO›/sso/isAlive.jsp
, например
https://id.megafon.ru:7443/sso/isAlive.jsp

Проверка должна выполняться не чаще, чем 1 раз в 10 секунд. В случае, если запрос возвращает HTTP код, отличный от 200, система мониторинга должна уведомлять о проблемах с коннективностью до WebSSO.

Технически проверка доступности WebSSO может быть реализована любым удобным способом, важно, чтобы проверка выполнялась с того же сервера, на котором расположеное приложение-клиент WeSSO.

<<<
== Ссылки

rfc6749.pdf - базовая спецификация протокола OAuth2.0.

rfc7009.pdf -  расширение базовой спецификации протокола OAuth2.0, описывающее token revocation.

SSO Auth Model_ v 1.4.doc - описание модели аутентификации и авторизации WebSSO

oauth2-integration.pdf -

http://www.w3.org/TR/1999/REC-html401-19991224/ - HTML спецификация версии 4.01

<<<
== Приложение 1. Базовые адреса конечных точек.
Раздел содержит значения для базовых адресов конечных точек, используемых в проекте.

.Базовые адреса конечных точек WebAPI управленя привязками
[options="header"]
|===
|Окружение|Базовый адрес|Комментарий
|vagrant|http://sso.ocb.hosted:8080/webapi-1.0|
|интеграционное||
|продуктив||
|===