

# CA API Management OAuth Toolkit - 3.4

## CA API Management OAuth Toolkit - Home

Date: 21-Apr-2016





This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2016 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

# Table of Contents

---

<b>Release Notes 3.4.x .....</b>	<b>12</b>
OTK Version Support Matrix .....	12
What's New in 3.4.00 .....	13
CA API MAS 1.1 Release Support .....	13
CA API SaaS Portal Integration .....	13
API Change .....	13
New WADL Files .....	14
Known Issues .....	14
Resolved Issues .....	14
 <b>Supporting Files .....</b>	 <b>15</b>
Create a New Database .....	15
Upgrade an Existing Database .....	15
Install Test Data .....	16
WADL Files .....	16
Cron Jobs .....	17
 <b>CA API Gateway OAuth Toolkit .....</b>	 <b>18</b>
Compliance .....	18
 <b>Installation Workflow .....</b>	 <b>20</b>
Create or Upgrade the OTK Database .....	20
MySQL Database .....	20
Before You Begin .....	21
Create the OTK Database .....	21
Upgrade an OTK Database .....	21
Oracle Database .....	22
Create an OTK Database .....	22
Upgrade an OTK Database .....	23
Apache Cassandra Database .....	23
Create an OTK Database .....	24
Create the Database Connection .....	24

Upgrade a Cassandra OTK Database .....	25
Manage Data Sources .....	25
Manage JDBC Connections .....	25
MySQL Database Connection Properties .....	26
Oracle Database Connection Properties .....	26
Support for Multiple Local Databases .....	27
Manage Cassandra Connections .....	27
The Real Cassandra Connection .....	27
The Empty JDBC Connection .....	28
Install the OAuth Solution Kit .....	28
Before you Begin .....	28
Launch the OAuth Solution Kit Installer .....	28
Select and Install Specific Solution Kits .....	29
Resolve Entity Conflicts .....	30
Identify the JDBC Connection for the OAuth Entity .....	30
Multiple Gateway Scenario .....	31
Install DMZ Solution Kit Components .....	32
Modify Policies on the DMZ Gateway .....	33
Export the SSL Certificate from the Internal Gateway .....	33
Import the SSL Certificate into the DMZ Gateway .....	33
Configure the Internal Gateway .....	34
Installing on a Remote Database .....	36
Configure Authentication .....	36
OAuth Validation and Storage .....	37
Create a FIP .....	38
Create a Federated User .....	38
Add Authentication Against FIP .....	39
SAML Grant Type Support .....	40
Select SAML Options for SSL Certificates .....	40
Create a FIP .....	41
Enable the SAML Grant Type .....	42
User Authentication Options .....	43
Authenticate against a Custom Identity Provider .....	44
Authenticate against CA SiteMinder .....	44
Post-Installation Tasks .....	45
Restart the Gateway .....	45
Set the Database Type .....	45
Import the Public Certificate .....	46
Set a UUID Value for CookieKey .....	46
Configure ID_TOKEN Attributes .....	47
Verify the Installation .....	47
Run the OAuth 1.0 Test Client .....	48

Security Precautions .....	48
Run the Test Client .....	48
Further Configuration .....	48
Run the OAuth 2.0 Test Client .....	49
Security Precautions .....	49
Enable the Client .....	49
Run the Client .....	50
Restrict the OAuth 2.0 Grant Types .....	52
Change Resource Owner Authentication .....	52
Add SLA Rules .....	53
Verify the OAuth Infrastructure .....	53
Verify the OAuth Infrastructure .....	53
<b>Upgrade and Uninstall .....</b>	<b>55</b>
Uninstall Previously Installed Solution Kits .....	55
Uninstalling an Entire Solution Kit with Instance Modifiers .....	55
Uninstall Installations Prior to OTK Version 3.2 .....	56
Upgrade an Existing OTK Installation .....	57
<b>Prepare JSON Message for Export .....</b>	<b>58</b>
Include the OAuth Server Certificate .....	58
Assign OAuth Server Details .....	58
JSON Export Endpoint .....	59
Enable the Endpoint .....	59
Configure Endpoint Access .....	60
JSON Message Example .....	60
Server .....	60
MAG .....	61
OAuth .....	62
Custom .....	64
<b>Using the OAuth Manager .....</b>	<b>65</b>
Log into OAuth Manager .....	65
Register a Client .....	65
Master Key .....	66
Environment and Scope .....	66
Callback URI .....	67
Manage Clients .....	67

List Client Keys .....	68
Available Actions .....	69
Field Information .....	69
Manage Tokens .....	70
Available Actions .....	70
<b>Secure an API Endpoint with OAuth .....</b>	<b>71</b>
OTK Require OAuth 1.0 Token .....	71
OTK Require OAuth 2.0 Token .....	71
<b>OAuth Request Scenarios .....</b>	<b>74</b>
Requests Specifying response_type .....	74
response_type=token .....	74
response_type=token id_token .....	75
response_type=code .....	77
Requests Specifying grant_type .....	78
grant_type=password .....	78
grant_type=client_credentials .....	79
grant_type=authorization code .....	80
grant_type=urn:ietf:params:oauth:grant-type:jwt-bearer .....	80
grant_type=urn:ietf:params:oauth:grant-type:saml2-bearer .....	81
grant_type=refresh_token .....	82
<b>Customizing the OAuth ToolKit .....</b>	<b>83</b>
Configure Token Lifetime Properties .....	83
Customize Token Lifetimes .....	83
Customize Caches .....	84
OTK Client DB .....	84
OTK Session DB .....	85
OTK Session GET .....	85
OTK Require OAuth 2.0 Token .....	86
/auth/oauth/v1/authorize .....	86
/auth/oauth/v2/authorize .....	87
/oauth/manager .....	89
/oauth/manager/clients .....	89
/oauth/manager/tokens .....	89
/oauth/validation/validate/v1/signature .....	90
Customize the OAuth 2.0 Authorization Server Website .....	90

Customizing Website Template .....	90
Hosting Website on an External Web Server .....	90
Manage API Keys with CA API Portal .....	91
Before You Begin .....	92
Replace the OTK Client DB Get Policy .....	92
Register OAuth Test Clients .....	92
Create the Request .....	93
Optimization .....	93
Remove Expired Tokens .....	94
Run Cron Jobs .....	94
Modify Variables .....	94
Configure Local Cache Lifetimes .....	94
Configure Session Lifetime Properties .....	95
Automate Database Maintenance .....	95
Delete Expired Sessions and Tokens .....	95
Delete Expired OAuth Clients .....	96
Delete Expired Sessions .....	96
Delete Temporary Sessions .....	96
Delete Long Living Tokens .....	97
OTK User Role Configuration .....	97

<b>OpenID Connect Implementation .....</b>	<b>99</b>
Import Certificates .....	99
Configure the Callback URL of the Test Client .....	99
Run the Test Client .....	100
Implementation Details .....	101
Supported Features .....	101
Valid OpenID Connect Requests .....	101
Persistence .....	101
Endpoints .....	101
Using the OpenID Connect Assertions .....	102
Decode ID Token .....	102
Context Variables Created by this Assertion .....	103
Configure Assertion Properties .....	103
Generate ID Token .....	103
Context Variables Created by this Assertion .....	104
Assertion Properties .....	104
Configuration Required .....	104
Configuration Required for OAuth 2.0 Implicit Flow Only .....	104
Optional Configuration .....	105

<b>APIs and Assertions .....</b>	<b>106</b>
APIs .....	106
OAuth API Endpoints .....	106
OAuth Validation Point (OVP) API .....	185
Endpoints .....	185
Associated Tasks .....	207
OAuth Protected APIs .....	211
Clientstore API .....	215
Request values of a given client_key .....	284
Request values of a given client, client_key at once .....	285
Tokenstore API .....	285
Register a Token .....	344
Update a Token .....	345
Revoke a Token .....	346
Delete a Token .....	346
Retrieve Token Values .....	346
Retrieve Temporary Token Values .....	347
Sessionstore API .....	348
Portal Storage API .....	357
OAuth Client Assertions .....	364
Retrieve OAuth 1.0 Token Assertion .....	364
Context Variables Set .....	365
Context Variables Used .....	365
Consume OAuth 1.0 Resource .....	365
Context Variables Set .....	365
Context Variables Used .....	365
Retrieve OAuth 2.0 Token Assertion .....	366
Context Variables Set .....	366
Refresh OAuth 2.0 Token Assertion .....	366
Context Variables Set .....	366
Encapsulated Assertions .....	366
OTK Require OAuth 2.0 Token .....	367
Context Variables .....	368
OTK Access Token Retrieval .....	368
Context Variables .....	369
OTK Client Persist .....	369
Create a New Client .....	370
Create a New Client ID .....	370
Create a New Client with a New Client ID .....	371
Context Variables .....	371
OTK SCOPE Verification .....	371

Context Variables .....	372
Error Codes .....	372
How to Add Error Codes to a Policy .....	372
Error Handling .....	373
Error Codes .....	374

# CA API Management OAuth Toolkit - Home

---

# Release Notes 3.4.x

These release notes contain the following 3.4.x version details for the CA API Management OAuth Toolkit (OTK):

- [OTK Version Support Matrix \(see page 12\)](#)
- [What's New in 3.4.00 \(see page 13\)](#)
- [Known Issues \(see page 14\)](#)
- [Resolved Issues \(see page 14\)](#)

## OTK Version Support Matrix

The following table shows the OAuth Toolkit version and CA API Gateway version compatibility.

	7.0.0	7.1.0	8.0.0	8.1.0	8.1.1	8.1.02	8.2.0	8.3.0	8.4.0	9.0.0	9.1.0
CA API Gateway Version											
OTK Released Versions											
1.0.											
2.0											
2.1											
2.1.1											
2.1.2											
3.0											
3.1.1											
3.2.0											
3.3.01											
3.4											

CA API Gateway 9.1 supports OTK version 3.4 on:

- Oracle 11g
- MySQL 5.5.39 Enterprise edition
- Apache Cassandra 2.1.7

# What's New in 3.4.00

The OTK 3.4.00 contains the following new items:

- [CA API MAS 1.1 Release Support \(see page 13\)](#)
- [CA API SaaS Portal Integration \(see page 13\)](#)
- [API Change \(see page 13\)](#)
- [New WADL Files \(see page 14\)](#)

## CA API MAS 1.1 Release Support

OTK 3.4 supports the CA API Mobile Application Service (MAS) 1.1 release.

Access the latest CA API MAS release documentation at [wiki.ca.com/mas](http://wiki.ca.com/mas) (<http://wiki.ca.com/>).

## CA API SaaS Portal Integration

Two new solution kits located within the OAuth solution kit sskar file facilitate CA API SaaS Portal Integration with the OTK:

- Internal, Portal
- Shared Portal Resources

After installation, no further policy configuration is required.

Clients are registered and configured through the SaaS portal. A JSON sync message provides configuration to the Portal API Key Sync encapsulated assertion. Keys are then mapped to values in the portal\_apikey table of the OTK database.



Integration with the CA API Developer Portal (on-premise) still requires manual configuration. See [Manage API Keys with CA API Portal \(see page 91\)](#).

Installation of the CA API Portal solution kits creates the /portal/storage API.

For installation instructions, see [Install the OAuth Solution Kit \(see page 28\)](#).

## API Change

The resource\_owner\_session\_status API at /connect/session/status no longer returns **expired** as a session status. The session status is returned as either **active** or **none**.

## New WADL Files

Changes are captured in new 3.4.00 WADL files.

See [Supporting Files \(see page 15\)](#).

## Known Issues

Reference	Description
MST-202	Storage endpoint request fails when OTK is hosted on Cassandra database. The OTK Client Get encapsulated assertion does not support passing only the Client ID and get operation when using Cassandra.
MST-247	OAuth 1.0 issue. MySQL and Cassandra database return different status codes when accessing /auth/oauth/v1/authorize with expired session. MySQL returns a 401 status and "session expired" message. Cassandra returns a status 200 and a message "Session has expired, please start again."
MST-147	The Upgrade button in the solution kit installer is inutile. Do not use it.
MST-50	OTK Token DB Get returns all tokens if token_status is set to empty. Expected behavior is an empty result.

## Resolved Issues

The following table lists issues that existed in previous releases and are fixed in release OTK 3.4.

Reference	Description
MST-141	Unable to do anonymous client export. The auth/oauth/v2/client/export api was missing the OTK Variable Configuration assertion and was unable to find the variable \${enable_anonymous_client_export}. This issue has been resolved.
MST-196	MAS Messaging fails to enforce client id structure: <device_specific_id>::<client_id>::<SCIM userID> . This issue has been resolved.
MST-149	Invalid SAML assertion returns incorrect error code and message and stops the policy from processing. When an invalid assertion is passed to the /auth/oauth/v2/token endpoint the expected server is: Status 400 Error Code 3003103. This issue has been resolved.

# Supporting Files

To download a file, click the filename of the script file.

- [Create a New Database \(see page 15\)](#)
- [Upgrade an Existing Database \(see page 15\)](#)
- [Install Test Data \(see page 16\)](#)
- [WADL Files \(see page 16\)](#)
- [Cron Jobs \(see page 17\)](#)



## Create and Upgrade Database File Collection

Click the image below to download the complete create and upgrade file collection.



## Create a New Database

To create a new OTK database, you need both the create and the testdata scripts. For instructions see [Create or Upgrade the OTK Database \(see page 20\)](#).

MySQL	Oracle	Cassandra
<a href="#">otk_db_schema.sql</a>	<a href="#">otk_db_schema_oracle.sql</a>	<a href="#">otk_db_schema_cassandra.cql</a>
<a href="#">otk_db_testdata.sql</a>	<a href="#">otk_db_testdata_oracle.sql</a>	<a href="#">otk_db_testdata_cassandra.cql</a>

## Upgrade an Existing Database

Start with the script corresponding to your current OTK version, then work up the list, executing all scripts until you reach the latest version.

For instructions see [Create or Upgrade the OTK Database \(see page 20\)](#).

MySQL Database	Oracle Database	Cassandra Database
----------------	-----------------	--------------------

MySQL Database	Oracle Database	Cassandra Database
<a href="#">upgrade_otk3.3.01-otk3.4.00.sql</a>	<a href="#">upgrade_otk3.3.01-otk3.4.00.oracle.sql</a>	<a href="#">otk_db_schema_cassandra_update_3.01-3.4.00.cql</a>
<a href="#">upgrade_otk3.2.00-otk3.3.01.sql</a>	<a href="#">upgrade_otk3.2.00-otk3.3.01.oracle.sql</a>	<a href="#">otk_db_schema_cassandra_update_3.00-3.3.01.cql</a>
<a href="#">upgrade_otk3.1.1-otk3.2.0.sql</a>	<a href="#">upgrade_otk3.1.1-otk3.2.0.oracle.sql</a>	<a href="#">otk_db_schema_cassandra_update_3.00-3.3.01.cql</a>
<a href="#">upgrade_otk3.0-otk3.1.1.sql</a>	<a href="#">upgrade_otk3.0-otk3.1.1.oracle.sql</a>	<a href="#">upgrade_otk3.0-otk3.1.1-otk3.2.0_cassandra.cql</a>
<a href="#">upgrade_otk2.1-otk3.0.0.sql</a>	<a href="#">upgrade_otk2.1-otk3.0.0.oracle.sql</a>	
<a href="#">upgrade_otk2.0-otk3.0.0.sql</a>	<a href="#">upgrade_otk2.0-otk3.0.0.oracle.sql</a>	
<a href="https://wiki.ca.com/download/attachments/220332718/upgrade_otk1.0-otk2.0.sql?version=1&amp;modificationDate=1430913947645&amp;api=v2">upgrade_otk1.0-otk2.0.sql (<a href="https://wiki.ca.com/download/attachments/220332718/upgrade_otk1.0-otk2.0.sql?version=1&amp;modificationDate=1430913947645&amp;api=v2">https://wiki.ca.com/download/attachments/220332718/upgrade_otk1.0-otk2.0.sql?version=1&amp;modificationDate=1430913947645&amp;api=v2</a>)</a>		

## Install Test Data

Any test data from the previous release is persisted. In the event you need to re-install the OTK test data, use one of the following files.

MySQL	Oracle	Cassandra
<a href="#">otk_db_testdata.sql</a>	<a href="#">otk_db_testdata_oracle.sql</a>	<a href="#">otk_db_testdata_cassandra.cql</a>

## WADL Files

The WADL files describe the APIs used in the OAuth Toolkit. The files are only updated to a newer version when API changes occur.

WADL Files
------------

---

**WADL Files**[otk\\_portalstorage\\_wadl.xml](#)[otk\\_server\\_wadl.xml](#)[otk\\_sessionstorage\\_wadl.xml](#)[otk\\_tokenstorage\\_wadl.xml](#)[otk\\_clientstorage\\_wadl.xml](#)[otk\\_oauth\\_apis\\_wadl.xml](#)[otk\\_ovp\\_wadl.xml](#)

---

## Cron Jobs

The sample cron jobs in the cronjob.zip file are for deleting tokens and sessions from the database. These examples are for MySQL databases but can be modified for Oracle. See [Optimization \(see page 93\)](#).

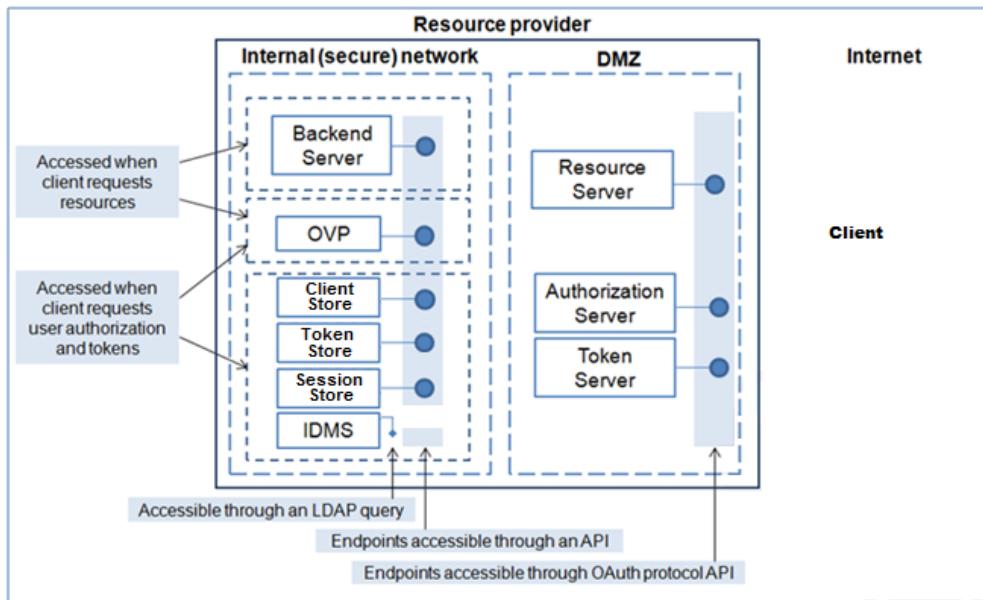


# CA API Gateway OAuth Toolkit

The CA API Gateway OAuth Toolkit is separated in the following logically different components.

Component	Notes
OAuth Validation Point (OVP)	An endpoint that validates incoming requests for OAuth 1.0 and OAuth 2.0. It is accessed via a REST API.
API Proxy	The CA API Gateway holding the OAuth installation enforcing the OAuth token requirement.
Clientstore	All oauth_consumer_keys (OAuth 1.0) and client_ids (OAuth 2.0) are stored here. The clientstore is accessible via a REST API.
Tokenstore	All tokens are stored here. The tokenstore is accessible via a REST API.
Sessionstore	An endpoint that provides caching and session services to the OTK components. This allows OTK components to avoid going to the database in calls to clientstore and tokenstore APIs.
Resource Server	Provides endpoints to access resources. These endpoints require a valid OAuth token.

The following graphic displays the components within their preferred network zones.



## Compliance

The CA OAuth Toolkit provides a full featured and standards compliant OAuth 1.0 and 2.0 solution.

OAuth is an authorization standard that allows one service to integrate with another service on behalf of a user. Instead of exposing user credentials, an OAuth access token is issued and accepted for user authentication. The OAuth authorization framework permits a user to grant an application (consumer) access to a protected resource without exposing the user's password credentials.

This implementation conforms to the following specifications:

- **OAuth 1.0:** <http://tools.ietf.org/html/rfc5849>
- **OAuth 2.0:** <http://tools.ietf.org/html/rfc6749>

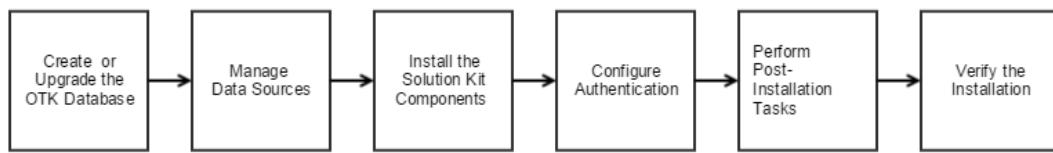
This implementation may provide incomplete support for the following draft specifications:

- **MAC:** <https://tools.ietf.org/html/draft-ietf-oauth-v2-http-mac-01#section-3.2>
- **Base 64:** <https://tools.ietf.org/html/rfc4648#section-5>



Specifications can change without notice, possibly causing the OAuth Toolkit to produce incorrect results.

# Installation Workflow



To create the OAuth Tool Kit database and install OTK policy components, follow the OTK installation workflow. The workflow provides instructions for performing both new installations and upgrades.

Perform the following tasks to install and configure the OAuth Toolkit:

- [Create or Upgrade the OTK Database \(see page 20\)](#)
- [Manage Data Sources \(see page 25\)](#)
- [Install the OAuth Solution Kit \(see page 28\)](#)
- [Configure Authentication \(see page 36\)](#)
- [Post-Installation Tasks \(see page 45\)](#)
- [Verify the Installation \(see page 47\)](#)

## Create or Upgrade the OTK Database

The following databases are supported for the OTK:

- [MySQL Database \(see page 20\)](#)
- [Oracle Database \(see page 22\)](#)
- [Apache Cassandra Database \(see page 23\)](#)

The database scripts are available on [Supporting Files \(see page 15\)](#).

For the compatibility of supported versions see the release notes.



Whether you are creating or upgrading, backup any existing database version before proceeding.

## MySQL Database

- [Before You Begin \(see page 21\)](#)

- [Create the OTK Database \(see page 21\)](#)
- [Upgrade an OTK Database \(see page 21\)](#)

## Before You Begin

Before creating or upgrading the MySQL database, perform the following tasks:

- Ensure that MySQL is installed on a database host machine.
- Prepare the OTK .sql scripts you require on your local machine.

The .sql script files referenced by these instructions are available for download from this site. See [Supporting Files \(see page 15\)](#).

## Create the OTK Database

To create the OTK database:

1. Use the mysql program to connect to the server as the MySQL root user. The following script example creates a database called "otk\_db".

```
root@machine_name> mysql
mysql> CREATE DATABASE otk_db;
mysql> GRANT SELECT,UPDATE,DELETE,INSERT ON otk_db.* TO '<db_user>'@'localhost'
identified by '<db_user_password>';
mysql> flush privileges;
mysql> exit;
```

2. Now run the script to create the schema.

```
root@machine_name> mysql -u root otk_db < otk_db_schema.sql
```

Running otk\_db\_schema.sql creates the schema files and installs test data. If you need to re-install the otk test data, use the file otk\_db\_testdata.sql

3. **Save.**

You have created the OTK database.

## Upgrade an OTK Database

Perform the following tasks to upgrade an existing MySQL OTK database:

- [Back Up Your Database \(see page 21\)](#)
- [Delete Existing OAuth Tokens \(see page 22\)](#)
- [Determine your Current OTK Database Version \(see page 22\)](#)
- [Run the Upgrade Scripts \(see page 22\)](#)

### Back Up Your Database

As a precaution, back up your OTK database before running the upgrade scripts.

The following commands back up an existing database named otk\_db:

```
[root@ssg]# cd /home/ssgconfig
[root@ssg]# mkdir dbbackup
[root@ssg]# cd dbbackup
[root@ssg]# mysqldump otk_db > otk_backup.sql
```

## Delete Existing OAuth Tokens

Optional. Reduce the upgrade script execution time by deleting any existing OAuth tokens. Note that deleting OAuth tokens causes client application users to re-authenticate.

```
mysql> DELETE FROM oauth_token;
```

## Determine your Current OTK Database Version

To determine the version of your existing OTK database:

1. Connect to your database.
2. As the root MySQL user, run the following command from the mysql shell:

```
mysql> use <dbname>;
mysql> select * from otk_version;
```

## Run the Upgrade Scripts

As the database user, run the upgrade scripts from the mysql command line.

For example:

```
mysql> source <location>/upgrade_otk3.3.01-otk3.4.00_mysql.sql
```

# Oracle Database

The following tasks apply to an Oracle database:

- [Create an OTK Database \(see page 22\)](#)
- [Upgrade an OTK Database \(see page 23\)](#)

Before creating or upgrading the Oracle database, perform the following tasks:

- Ensure that Oracle 11g is installed on a database host machine
- Prepare the OTK .sql scripts you require on your local machine.

The .sql script files referenced by these instructions are available for download from this site. See [Supporting Files \(see page 15\)](#).

## Create an OTK Database

To create an OTK database on Oracle 11g:

1. Connect to the Oracle server and start SQL Plus.

For example:

```
cd /u01/app/oracle/product/11.2.0/xe/bin/
source oracle_env.sh
sqlplus
Username: SYSTEM
Password: mypassword
```

2. In SQL> mode create a database user.

```
SQL> create user <db_user> identified by <db_user_password>;
SQL> grant connect, resource to <db_user>;
--- Note: CONNECT role enables user to connect to the database
--- Note: RESOURCE role enables user to create certain types of schema
objects in that user's own schema (ie. it grants the create table, but not
create view)
SQL> exit
```

3. Run the following command. In this example the sql script is stored in the /temp directory.

```
sqlplus db_user/db_user_password @/temp/otk_db_schema_oracle.sql
```

Running otk\_db\_schema\_oracle.sql creates the schema files and installs test data.  
If you need to re-install the otk test data, use the file otk\_db\_testdata\_oracle.sql

4. **Save.**

You have created the OTK database.

## Upgrade an OTK Database

Download the SQL scripts you require based on your existing OTK version. SQL scripts are available on the [Supporting Files \(see page 15\)](#) page.

To upgrade an OTK database on Oracle, connect to the Oracle server and start SQL Plus.

## Apache Cassandra Database

Apache Cassandra™ is an open source non-relational NoSQL database. The following tasks are for a Cassandra database running on a Linux machine:

- [Create an OTK Database \(see page 24\)](#)
- [Create the Database Connection \(see page 24\)](#)
- [Upgrade a Cassandra OTK Database \(see page 25\)](#)

Before creating the Cassandra database, perform the following tasks:

- Ensure that a running Cassandra instance exists.
- Prepare the OTK .cql scripts you require on your local machine.

The .cql script files referenced by these instructions are available for download from the [Supporting Files \(see page 15\)](#) page.

## Create an OTK Database

To create an OTK database on Cassandra:

1. Download the schema and testdata cql files from the [Supporting Files \(see page 15\)](#) page.  
Store them locally.
2. Log in as the root user to the database node:

```
$ ssh root@yourCassandraDatabase
```

3. Launch the cqlsh shell and create the otk\_db keyspace from the prompt:

```
$ cqlsh
cqlsh> CREATE KEYSPACE otk_db WITH replication = {'class' : 'SimpleStrategy',
'replication_factor' : 1};
```

4. From the UNIX command line, run the scripts to create the schema and populate the tables with test data:

```
$ cqlsh -k otk_db -f otk_db_schema_cassandra.cql
$ cqlsh -k otk_db -f otk_db_testdata_cassandra.cql
```

### Additional Notes

Refer to the Apache Cassandra documentation for how to run external files.

For example, you can specify the IP Address and port to start cqlsh on a different node.  
You may need to provide user credentials if authentication is required.

```
$ cqlsh 123.123.123.123 9042 -u [username] -p [password] -f otk_db_schema_cassandra.cql
```

## Create the Database Connection

Apache Cassandra does not use JDBC connections. Create the Cassandra database connection before policy selection using the following procedure. The connection must be named "OAuth\_Cassandra". Later, when you install policies, you create an empty JDBC connection.

To create a Cassandra connection:

1. Navigate to **Tasks > Manage Data Sources**.
2. Select **Manage Cassandra Connections**.
3. Click **Add**.

4. Configure connection properties as shown in the following table:

Property	Value	Notes
Connection Name	<b>OAuth_Cassandra</b>	You must use this value as the connection name.
Contact Points	myCassandra. myCorp.com	Cassandra nodes separated by commas. IP address or DNS
Port	9042	Default value.
Keyspace	otk_db	Default value. Existing keyspaces can be viewed in cqlsh using "DESCRIBE keyspaces".
Username	root	
Password	dbpassword	

## Upgrade a Cassandra OTK Database

Find the scripts to upgrade a Cassandra OTK database on the [Supporting Files \(see page 15\)](#) page. You may need to run multiple scripts sequentially to upgrade your current otk version to the most recent version.

To upgrade an OTK database on Cassandra:

1. Open an ssh window to a Cassandra node:

```
$ssh root@node.cassandra.myDomain.com
```

2. From the UNIX command line, run the scripts found on the to update the otk data. The following example upgrades a Cassandra database from version 3.3.01 to 3.4.00:

```
$ cqlsh -k otk_db -f otk_db_schema_update_3.3.01-3.4.00.cql
```

## Manage Data Sources

Configure the data source connection to the database then identify this connection when installing the shared resources solution kit.

There are two types of data source connections you can manage:

- [Manage JDBC Connections \(see page 25\)](#)
- [Manage Cassandra Connections \(see page 27\)](#)

## Manage JDBC Connections

JDBC connections are used with MySQL and Oracle databases.



To create a JDBC connection:

1. From the Policy Manager, navigate to **Tasks, Sources, Manage JDBC Connections**.
2. Click **Add** or select an existing connection and click **Clone**.
3. Configure connection properties based on database type. Refer to the sections below.
4. Click **Test** to verify the JDBC connection works.

## MySQL Database Connection Properties

The default OAuth connection uses the properties shown in the following table. Provide your own URL value.

Property	Value	Notes
Connection Name	<b>OAuth</b>	The name of the JDBC Connection that will be created. Maximum 128 characters.
Driver Class	com.mysql.jdbc.Driver	Select from the driver classes or provide your own. A support description appears after you select.
JDBC URL	jdbc:mysql://localhost: 3306/otk_db	
User Name	otk_user	
Password	password	Replace with a more secure password.

## Oracle Database Connection Properties

If you are using an Oracle 11g database, create a connection using the values shown in the following tables.

Property	Value	Notes
Connection Name	<b>OAuth</b>	The name of the JDBC Connection that will be created. Maximum 128 characters.
Driver Class	com.l7tech.jdbc.oracle.OracleDriver	Select from the driver classes or provide your own. A support description appears after you select.
JDBC URL	jdbc:l7tech: oracle://<yourOracleDBServer>:1521	
User Name	db_user	
Password	db_user_password	

### Additional Properties

Property	Value
Database	<yourDatabaseName>

## Support for Multiple Local Databases

Multiple local databases for distinct transactions can be supported. For example, you can dedicate one database for client configurations, one for token management, and another for session handling. To support such a configuration, create one JDBC connection per database and modify the policies to adjust the JDBC assertions to use a non-default JDBC connection. Policies containing JDBC assertions are found in the subfolders of Policy Fragments/persistence/.

## Manage Cassandra Connections

Apache Cassandra™ does not use JDBC connections, however, the component installer requires a JDBC connection to be selected. You therefore need to create a real Cassandra connection, as well as an empty JDBC connection.

- [The Real Cassandra Connection \(see page 27\)](#)
- [The Empty JDBC Connection \(see page 28\)](#)

### The Real Cassandra Connection

Create the real Cassandra database connection using the following procedure. The connection name must be "OAuth\_Cassandra". This name is hardcoded in policies.



To create the real Cassandra connection:

1. Navigate to **Tasks, Data Sources > Manage Cassandra Connections**.
2. Click **Add**.
3. Configure connection properties as shown in the following table.

Property	Value	Notes
Port	9042	Default value.
Password	dbpassword	
Contact Points	myCassandra. myCorp.com	Cassandra nodes separated by commas. IP address or DNS
Connection Name	<b>OAuth_Cassandra</b>	You must use this value as the connection name. The value is hardcoded in the policies.
Keyspace	otk_db	Default value. Existing keyspaces can be viewed in cqlsh using "DESCRIBE keyspaces".
Username	root	

For more connection configuration details, see [Apache Cassandra Database \(see page 23\)](#).

## The Empty JDBC Connection

This empty connection satisfies the requirements of the solution kit installer, is subsequently ignored, and the real Cassandra connection is used.



To create the real Cassandra connection:

1. Navigate to **Tasks, Data Sources, Manage JDBC Connections**.
2. Click **Add**.
3. Configure connection properties as shown in the following table.

Property	Value	Notes
Connection Name	forCassandra(empty)	Any name works
Driver Class	com.l7tech.jdbc.oracle.OracleDriver	Select any available value
JDBC URL	whatever	Does not have to be a real URL
User Name		Leave empty
Password		Leave empty

## Install the OAuth Solution Kit

The OAuth Solution Kit contains the policies, endpoints, and assertions that create the OAuth Toolkit (OTK). From the Policy Manager, install the single OAuth Solution Kit .sskar file. This file contains multiple solution kits that provide specific OAuth functionality.

## Before you Begin

- Install the CA API Gateway.  
The CA API Gateway license enables installation of the OAuth Solution Kit.
- Download the OAuth Solution Kit File.  
The OAuth Solution kit is distributed as a .sskar file: OAuthSolutionKit-3.4.00-52.sskar.
- Configure JDBC and Cassandra database connections.  
See [Manage Data Sources \(see page 25\)](#).

## Launch the OAuth Solution Kit Installer

This is step one of the Solution Kit Installation Wizard.

1. In Policy Manager, go to Tasks, Extensions and Add-Ons, Manage Solution Kits.

2. Click **Install**.

3. Identify the **Solution Kit File** to use.

Click **File** and locate the signed skar file (.sskar) for the OAuth Solution Kit.

For example: OAuthSolutionKit-3.4.00-52.sskar

The path to the solution kit file appears, click **Next**.

## Select and Install Specific Solution Kits

This is step two of the Solution Kit Installation Wizard. The OAuth Solution kit includes multiple solution kits.

Name	Version	Instance Modifier	Description
DMZ, OAuth 1.0	3.4.00-52		OAuth 1.0 protocol endpoints
DMZ, OAuth 2.0 and OpenID Connect endpoints	3.4.00-52		OAuth 2.0 protocol endpoints
Internal, Endpoint to access the client persistence layer	3.4.00-52		Storage, database access to client configuration storage
Internal, Endpoint to access the session persistence layer	3.4.00-52		Storage, database access to session storage
Internal, Endpoint to access the token persistence layer	3.4.00-52		Storage, database access to token storage
Internal, OAuth Validation Point	3.4.00-52		Endpoints that handle protocol related validations
Internal, Portal	3.4.00-52		Portal integration pieces (internal)
Internal, Server Tools	3.4.00-52		Tools, OAuth Manager and test clients
Shared OAuth Resources	3.4.00-52		Resources shared across OTK bundles
Shared Portal Resources	3.4.00-52		Portal integration pieces (common)

The solution kit includes DMZ, Internal, and Shared kits. In a single server scenario, install all kits on the same server. In a split OTK scenario, install the DMZ solution kits on the exposed server, and the Internal solution kits on the protected server. Shared OAuth Resources must be installed on both servers.

If you intend to integrate with the SaaS CA API Portal, install the **Internal Portal** and **Shared Portal Resources** kits.

To select and install specific solution kits:

1. Select one or more of the available solution kits listed.

See Solution Kit Selection note.

2. Are you upgrading?

If so, see the Upgrade Instructions note to set the Instance Modifier value.

3. Click **Next**.

The installer tests each solution kit for potential conflicts in the following areas:

- Service routing conflicts
- Policy conflicts
- Certificate conflicts
- Encapsulated Assertion conflicts
- Missing JDBC connections

- Missing assertions



#### Solution Kit Selection

Are you installing the OTK on two different servers?

See [Multiple Gateway Scenario \(see page 31\)](#)



#### Upgrade Instructions

Are you upgrading an existing OTK installation?

If so, select all solution kits and click **Set Instance Modifier**. Type a string value, then click **OK**. The instance modifier value must be different for each installation. The value is added to service resolution URIs, folders, policy names, and other components.

After comparing old and new policies, then manually merging any customizations to the new policies, further configuration is required to include the instance modifier in endpoint paths.

See [Upgrade and Uninstall \(see page 55\)](#).

## Resolve Entity Conflicts

This is step three of the Solution Kit Installation Wizard.

The installation test opens each solution kit and displays the entities. If an error is detected in any of the entities, the **Finish** button is grayed out and not available.

To resolve entity conflicts:

1. Click each solution kit tab and scan the Error Type column for any text.
2. Select the entity containing the error and click **Resolve**. A dialog box offers you actions to resolve the conflict.  
The Resolved column indicates when the conflict is resolved.

## Identify the JDBC Connection for the OAuth Entity

Found in the Shared OAuth Resources solution kit, the JDBC connection OAuth entity commonly requires conflict resolution. Identify the connection to resolve the conflict.

Internal, Endpoint to access the session persistence layer		Internal, Endpoint to access the token persistence layer		Internal, Endpoint to access the client persistence layer			Resolve
DMZ, OAuth 1.0		DMZ, OAuth 2.0 and OpenID Connect endpoints		Internal, OAuth Validation Point			
Internal, OAuth Validation Point		Internal, Server Tools		Shared OAuth Resources			
Name	Type	Action	Action Taken	Error Type	Resolved	Source ID	
id_token	FOLDER	AlwaysCreate...	CreatedNew	---	---	e001ctdUc1c1ffa...	
manage	FOLDER	AlwaysCreate...	CreatedNew	---	---	e001fd0c1c1ffa...	
OAuth	JDBC_CONNECTION	NewOrExisting	--	InvalidResource	No	4432207d16a1b5...	

To identify the JDBC connection:

1. Double-click the entity. The Resolve Entity Conflict dialog appears.
2. In the Action section, select an existing defined connection from the **Change to use this entity** selection box.  
If you have the connection does not exist, click **Manage** and **Add** to create a new data source connection. For more information, see [Manage Data Sources \(see page 25\)](#).



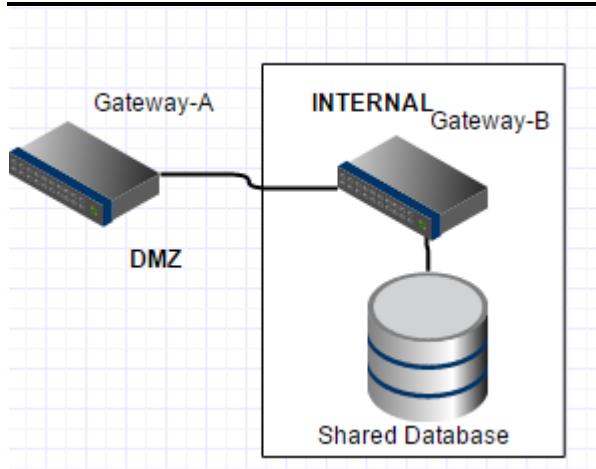
Cassandra databases do not use JDBC connections, however, definition of an empty JDBC connection is required.

After a new JDBC connection is defined you can select it from the Action section.

3. After selecting the JDBC connection to use, click **OK**.

When all conflicts are resolved, click **Finish** to start the installation. Finish can only be clicked if all conflicts are resolved.

## Multiple Gateway Scenario



The multiple gateway scenario configuration can be used to support the CA Mobile API Gateway (MAG) installed on two servers and includes the following tasks:

- Installation of the DMZ and Internal components on corresponding servers.
- Installation of the Shared components on both servers.
- Configuration of the hostname variable within policies on the DMZ server
- SSL certificate importation between gateways.

Configure the DMZ Server

The gateway on the DMZ is the client-facing OAuth server. These endpoints are exposed to the Internet. Perform the following tasks to configure the OTK on the DMZ server:

- [Install DMZ Solution Kit Components \(see page 32\)](#)
- [Modify Policies on the DMZ Gateway \(see page 33\)](#)
- [Export the SSL Certificate from the Internal Gateway \(see page 33\)](#)
- [Import the SSL Certificate into the DMZ Gateway \(see page 33\)](#)

## Install DMZ Solution Kit Components

Name	...	Description
<input checked="" type="checkbox"/> DMZ, OAuth 1.0	...	OAuth 1.0 protocol endpoints
<input checked="" type="checkbox"/> DMZ, OAuth 2.0 and OpenID Connect endpoints	...	OAuth 2.0 protocol endpoints
<input type="checkbox"/> Internal, Endpoint to access the client persistence layer	...	Storage, database access to client configuration storage
<input type="checkbox"/> Internal, Endpoint to access the session persistence layer	...	Storage, database access to session storage
<input type="checkbox"/> Internal, Endpoint to access the token persistence layer	...	Storage, database access to token storage
<input type="checkbox"/> Internal, OAuth Validation Point	...	Endpoints that handle protocol related validations
<input type="checkbox"/> Internal, Portal	...	Portal integration pieces (internal)
<input type="checkbox"/> Internal, Server Tools	...	Tools, OAuth Manager and test clients
<input checked="" type="checkbox"/> Shared OAuth Resources	...	Resources shared across OTK bundles
<input checked="" type="checkbox"/> Shared Portal Resources	...	Portal integration pieces (common)

To install the OTK components for the DMZ Gateway:

1. Open the Policy Manager and connect to the DMZ server.
2. Go to Tasks, Extensions and Add-Ons, Manage Solution Kits.
3. Click **Install** and locate the OAuth solution kit sskar file. Click **Next**.
4. Select solution kit components for installation:

Solution Kit Components	Notes
DMZ, OAuth 2.0 and OpenID Connect endpoints	Required.
DMZ, OAuth 1.0 protocol endpoints	Optional. Provides support for the older OAuth protocol.
Shared OAuth Resources	Required.
Shared Portal Resources	Optional. Only required if you intend to integrate with CA API Portal.

Click **Next**.

5. Resolve entity conflicts.  
Your JDBC connection is not set.  
Click the Shared OAuth Resources tab, select the **OAuth** entity, and click **Resolve**.  
Even though no database is co-located, create a fake JDBC connection. This does not need to point to an existing database.
6. Click **Finish**.

The DMZ folder under Server is populated. The Internal folder and SecureZone - Storage folder are installed, but remain empty.

## Modify Policies on the DMZ Gateway

By default, OTK policies support a single gateway scenario and are set to localhost. For the multiple gateway scenario, certain policies must point to the additional gateway.

Go to OTK-version/Policy Fragments/configuration and perform modifications as described in the following table.

Policy	Modifications
OTK OVP Configuration	Locate the variable host_oauth_ovp_server. Replace "localhost" with the hostname of the internal gateway.
OTK Storage Configuration	Modify the following context variables containing "localhost" with the hostname of the internal gateway: <ul style="list-style-type: none"> <li>▪ host_oauth_tokenstore_server</li> <li>▪ host_oauth_clientstore_server</li> <li>▪ host_oauth_session_server</li> </ul>

## Export the SSL Certificate from the Internal Gateway

To export the SSL certificate of the internal gateway:

1. With the Policy Manager connected to the internal gateway, go to Tasks, Certificates, Keys and Secrets, Manage Certificates.
2. Select the certificate with the server host name.
3. Select Properties.
4. Click Export. Save the certificate using .pem format.

## Import the SSL Certificate into the DMZ Gateway

1. With the Policy Manager connected to the DMZ gateway, go to Tasks, Certificates, Keys and Secrets, Manage Certificates.
2. Click **Import**. Locate the saved certificate of the internal gateway.
3. Click **Load**. The import certificates dialog box appears with the certificate highlighted.
4. In Certificate import options, click Import as Trust Anchor. Click OK.

The imported certificate must have **Trusted Anchor** selected.

To verify, select the imported certificate in the certificate list, click **Properties**, and select the Validation tab. The **Certificate is a Trust Anchor** check box should be selected. If it is not, select it now.

## Configure the Internal Gateway

The internal gateway hosts all validation and storage related endpoints. The OTK database is created on the shared database. The user has access to the database from localhost only.

- [Install Internal Solution Kit Components \(see page 34\)](#)
- [Export the SSL Certificate of the DMZ Gateway \(see page 35\)](#)
- [Import the DMZ Gateway Certificate \(see page 35\)](#)
- [Modify Policies \(see page 35\)](#)
- [Remove Unused Folders \(see page 36\)](#)

### Install Internal Solution Kit Components

Name	Description
<input type="checkbox"/> DMZ, OAuth 1.0	OAuth 1.0 protocol endpoints
<input type="checkbox"/> DMZ, OAuth 2.0 and OpenID Connect endpoints	OAuth 2.0 protocol endpoints
<input checked="" type="checkbox"/> Internal, Endpoint to access the client persistence layer	Storage, database access to client configuration storage
<input checked="" type="checkbox"/> Internal, Endpoint to access the session persistence layer	Storage, database access to session storage
<input checked="" type="checkbox"/> Internal, Endpoint to access the token persistence layer	Storage, database access to token storage
<input checked="" type="checkbox"/> Internal, OAuth Validation Point	Endpoints that handle protocol related validations
<input checked="" type="checkbox"/> Internal, Portal	Portal integration pieces (internal)
<input checked="" type="checkbox"/> Internal, Server Tools	Tools, OAuth Manager and test clients
<input checked="" type="checkbox"/> Shared OAuth Resources	Resources shared across OTK bundles
<input checked="" type="checkbox"/> Shared Portal Resources	Portal integration pieces (common)

To install the OTK components for the internal gateway:

1. Open the Policy Manager and connect to the Internal server.
2. Go to Tasks, Extensions and Add-Ons, Manage Solution Kits.
3. Click **Install** and locate the OAuth solution kit sskar file. Click **Next**.
4. Select components for your installation:

Solution Kit Components	Notes
Internal, Endpoint to access the client persistence layer	Database access to client configuration storage
Internal, Endpoint to access the session persistence layer	Database access to session storage
Internal, Endpoint to access the token persistence layer	Database access to token storage
Internal, OAuth Validation Point	Endpoints that handle protocol related validations
Internal, Portal	Optional. Required for CA API Portal integration
Internal, Server Tools	OAuth Manager for OAuth Client and token administration. Test clients.
Shared OAuth Resources	Resources shared across OTK bundles
Shared Portal Resources	Optional. Required for CA API Portal integration

5. Click **Next**.
6. Time to resolve entity conflicts.  
Your JDBC connection is not set.  
Click the Shared OAuth Resources tab, select the **OAuth** entity, and click **Resolve**.  
Select a JDBC connection that points to the local database. If you have not created one, see [Manage Data Sources \(see page 25\)](#).
7. Click **Finish**.

The Internal folder under Server is populated.

## Export the SSL Certificate of the DMZ Gateway

To export the SSL certificate:

1. Use Policy Manager to connect to the DMZ gateway.
2. Go to Tasks, Certificates, Keys and Secrets, Manage Certificates.
3. Select the certificate with the server host name.
4. Select **Properties**.
5. Click **Export**.  
Save the certificate using .pem format.

## Import the DMZ Gateway Certificate

Import the SSL certificate

1. Use Policy Manager to connect to the internal gateway.
2. Go to Tasks, Certificates, Keys and Secrets, Manage Certificates.
3. Click **Import**.  
Locate the saved certificate of the internal gateway.
4. Click **Load**.  
The import certificates dialog box appears with the certificate highlighted.
5. In Certificate import options, click **Import as Trust Anchor**. Click **OK**.

The imported certificate must have **Trusted Anchor** selected.

## Modify Policies

By default, OTK policies support a single gateway scenario and are set to localhost. In a multiple gateway scenario, certain policies must point to the additional gateway.

To support the OAuth 2.0 test clients, perform the following policy modifications:

Policy	Modifications
OpenIDConnectConfigServer	Set the host_web_authserver variable to the DMZ hostname value.
OpenIDConnectConfigClient	Set the following variables to the DMZ hostname value: <ul style="list-style-type: none"><li>▪ host_authserver</li><li>▪ host_tokenserver</li><li>▪ host_userinfoendpoint</li><li>▪ host_resourceendpoint</li></ul>
OTK Client Context Variables	Set host_oauth2_auth_server to the DMZ hostname value.

To support SAML Token authentication, perform the following policy modifications:

Policy	Modifications
OTK Client Context Variables	Set location_saml_token_server to the Internal hostname. This is used with the test clients.
/oauth/v2/samlTokenServer*	Set audience_recipient_restriction to point to the DMZ gateway. For example, <code>https://MAG-A hostname</code>

To support OAuth 1.0, perform the following policy configuration:

Policy	Modifications
OTK OAuth 1.0 Test Client Configuration	Set the following variables to point to the DMZ gateway: <ul style="list-style-type: none"><li>▪ host_oauth_endpoint</li><li>▪ host_api_endpoint</li></ul>

## Remove Unused Folders

To remove unused folders:

- Delete the folder MAG-version/Server/DMZ/OAuth 1.0.
- Delete the folder DMZ if it is empty.

## Installing on a Remote Database

When the database is not co-located with the Internal gateway, the same procedure can be followed, however, a JDBC connection must be made between the Internal gateway and the database.

# Configure Authentication

By default, OAuth policies related to client-certificate validation and SAML token-signing validation include Stop assertions that require manual configuration. Attempting to use the endpoints affected by these assertions fails unless manual validation configuration is completed.

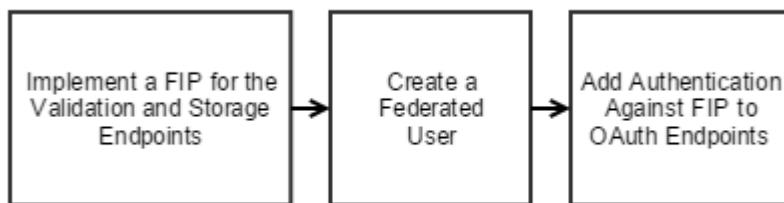
To configure validation:

1. Read the SAML Token Validation Configuration and the OAuth Storage and Validation Endpoints Configuration sections below.
2. If the scenario applies, click the workflow image to access configuration instructions. Both scenarios may apply, in which case, complete both configuration instruction sets.

#### OAuth Storage and Validation Endpoints Configuration

Policies provided with the CA Mobile API Gateway include endpoints that are used to access storage locations and execute validations. To access these endpoints, the policies require a mutual SSL connection (SSL with client authentication) and verify that the SSL handshake includes a client certificate. Additional manual configuration verifies the client certificate.

**Click the workflow image to access configuration instructions.**



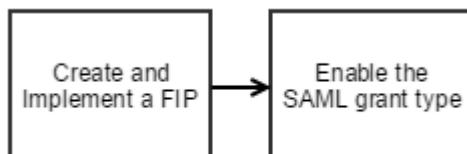
(see page 37)

Selection of the OAuth validation and storage endpoints during installation is optional. However, if selected, you must [configure client certificate verification \(see page 36\)](#).

#### SAML Token Validation Configuration

Policies provided with the CA Mobile API Gateway support the SAML 2.0 Bearer Assertion grant type, which uses a SAML token to authenticate users. By default, the OAuth policies validate the SAML token signature. Additional manual configuration verifies that the signature was generated by a trusted party.

**Click the workflow image to access configuration instructions.**



(see page 40)

If you do not intend to support SAML token, no further action is required.

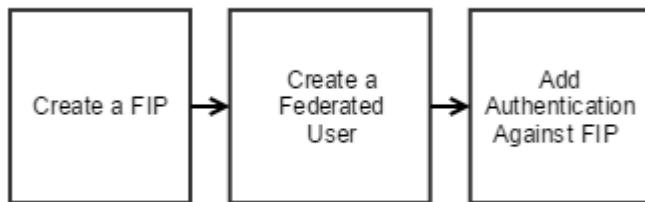
## OAuth Validation and Storage

An X.509 FIP must be implemented to validate client certificates.

Before implementing a FIP, ensure that any needed certificates have been imported.

In Policy Manager, go to **Tasks, Certificates, Keys and Secrets, Manage Certificates**. Certificates that must be imported include:

- The Gateway's own default SSL certificate
- The SSL certificate of any Gateway that is connecting as a client.

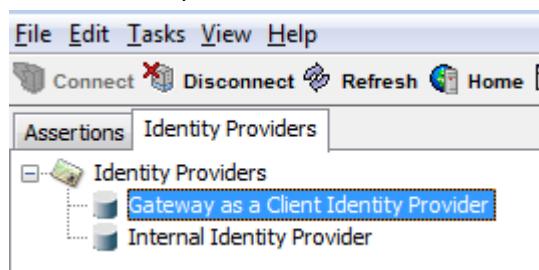


## Create a FIP

To create a FIP:

1. Navigate to **Tasks, Identity Providers, Create Federated Identity Provider**.
2. Click **Provider Name** and type a name. For example: "Gateway as a Client Identity Provider".
3. For Credential Source Type Allowed, select only the **X.509 Certificate** checkbox. Leave the SAML Token checkbox unchecked.  
Click **Next**.
4. Do not add any trusted certificates to this FIP. Leave the box blank.  
Click **Next**. A warning box appears. Click **OK**.
5. For **Validation**, select **Validate Certificate Path**.
6. Click **Finish**.

Click the Identity Providers tab to see the created FIP.

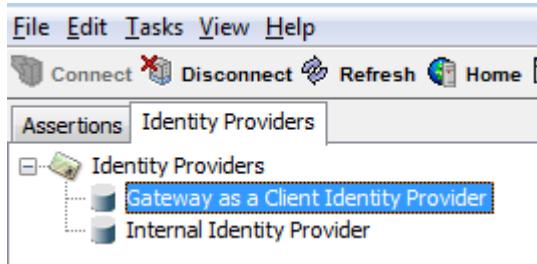


## Create a Federated User

For each client connecting to the validation and storage endpoints (possibly including the Gateway itself), create a Federated User within this new FIP. Identify the client's certificate for its outbound TLS connection.

To create a federated user:

1. In Policy Manager, select the Identity Providers tab.



2. Right-click the "Gateway as a Client Identity Provider" FIP you created and select **Create User**. The Create Federated User dialog appears.
3. Click **X.509 Subject DN** and enter the complete DN of the client certificate that will be imported for this user.  
For example: CN=gateway.example.com  
A default user name value is generated.
4. Select **Define Additional Properties** and click **Create**. The user properties dialog box appears.
5. Click the Certificate tab and click **Import**.
6. Import the SSL certificate of the client gateway as this user's certificate.  
If the gateway is connecting to itself, select **Import from Private Key Certificate Chain** and choose the default SSL key.  
If an external client (for example, a MAG in the DMZ) is connecting to these endpoints, select **Retrieve via SSL Connection (HTTP or LDAPS URL)**. Type a URL that leads to a listen port on the external client. For example: https://clientgateway.example.com:8443/
7. Click **Next**. View certificate details.
8. Click **Finish**.

## Add Authentication Against FIP

This procedure uses the following resources:

Resource	Location
OTK FIP Client Authentication	Encapsulated assertion located in OTK-version/Policy Fragments /authentication/
Authenticate Against Identity Provider	Assertion located in Policy Assertions > Access Control

To add authentication against a FIP:

1. Open the OTK FIP Client Authentication encapsulated assertion.
2. Drag the Authenticate Against Identity Provider assertion onto the following line in the OTK FIP Client Authentication encapsulated assertion:  
==== Drag the 'Authenticate Against Identity Provider' assertion onto this comment  
A selection window appears when you drop the assertion.

3. Select the OAuth Client Identity Provider you created, for example "Gateway as a Client Identity Provider".  
Click **OK**.

4. **Save and Activate** the encapsulated assertion.

No further configuration is required. By default, the OAuth storage and validation endpoints use the OTK FIP Client Authentication encapsulated assertion.

## SAML Grant Type Support

By default, support for the SAML token grant type is disabled in the policies delivered in the OAuth Toolkit. These tasks are required only if you intend to support the SAML grant type.



CA Mobile API Gateway installations: In a dual MAG scenario, perform the following tasks on the Gateway in the DMZ.



### Select SAML Options for SSL Certificates

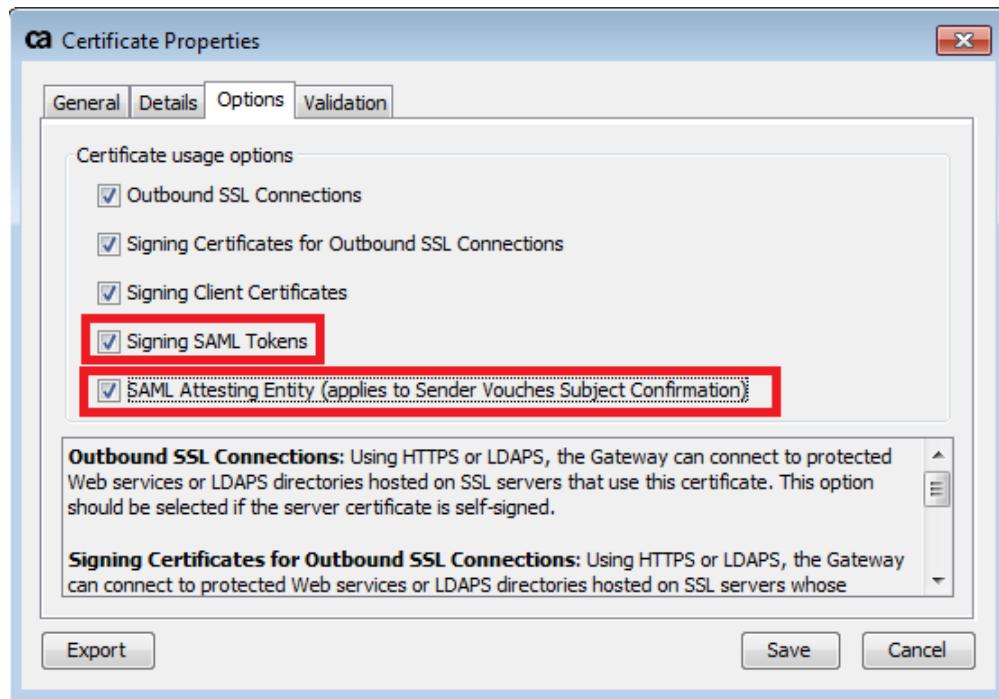
Certificates imported into the FIP include:

- The default certificate of the Gateway.
- The SSL certificate of any Gateway that is connecting as a client.

These certificates must have the additional options for SAML checked.

To select SAML options for SSL certificates:

1. In Policy Manager, go to **Tasks > Certificates, Keys and Secrets > Manage Certificates**.
2. Double-click the certificate to view properties. Select the Options tab.
3. Ensure that the two SAML options are checked. If they are not, select them and click **Save**.



## Create a FIP

To create a FIP:

1. Navigate to **Tasks > Identity Providers > Create Federated Identity Provider**.
2. Click **Provider Name** and type "OAuth SAML Identity Provider".
3. For **Credential Source Type Allowed** select only the **SAML Token** checkbox. Leave the X.509 Certificate checkbox unchecked.

Provider Name:	OAuth SAML Identity Provider
Credential Source Type Allowed:	<input type="checkbox"/> X.509 Certificate <input checked="" type="checkbox"/> SAML Token

4. Click **Next** to add trusted certificates.  
Trusted certificates include the gateway's own default SSL certificate, and the certificate of any Gateway that is connecting as a client.
5. Click **Add**.  
Click **Search**.  
A list of certificates appears. Click the gateway's own default SSL certificate. To add any

additional certificates, use Ctrl-click.

Click **Select**.

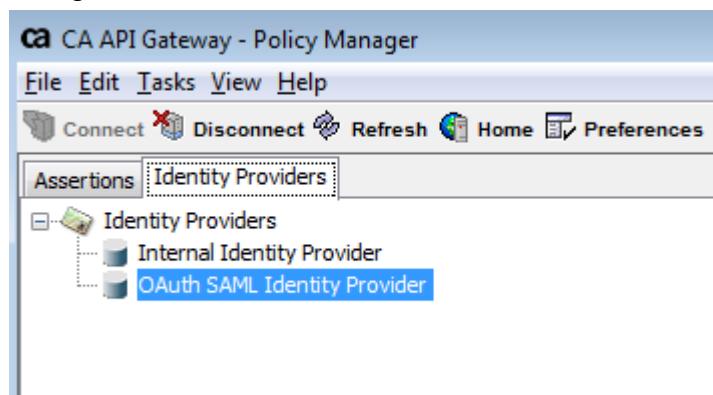
The certificates appear on the Trusted Certificates list.

Click **Next** to set certificate validation options.

6. For Validation, select **Validate Certificate Path**.

7. Click **Finish**.

Verify the FIP was created by clicking the Identity Providers tab in the upper left panel of the Policy Manager.



Now enable the SAML grant type.

## Enable the SAML Grant Type

This task is required if you intend to support the SAML grant type. Before you can enable the SAML grant type, you must create a FIP.

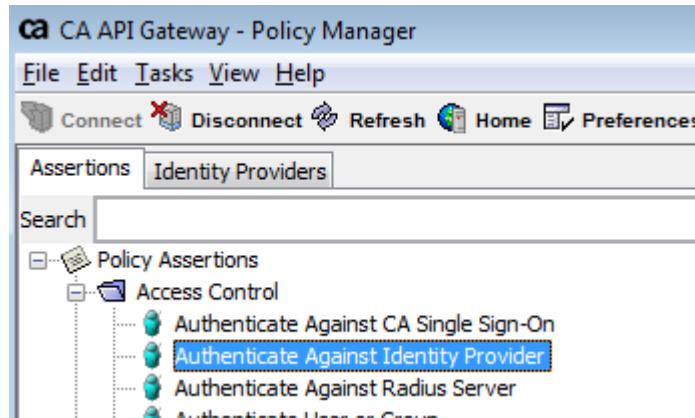
The following instructions add validation of the SAML Token Signer to SAML Token Grant Type policies. The SAML grant type is not supported until the Stop assertion is replaced with an Authenticate Against Identity Provider assertion. The FIP is selected as the provider.

To enable the SAML grant type:

1. Navigate to **OAuth-version/Policy Fragments/grant\_types** and open the **OTK grant\_type=SAML** policy.
2. Locate the Stop Processing assertion.

12	Comment: ===== Validate the trusted issuer by adding a FIP here =====
13	Comment: A FIP has to be introduced in order to accept only trusted clients
14	Comment: Remove the Stop assertion once the FIP is configured
15	Stop Processing

3. Replace the Stop Processing assertion in the policy with an **Authenticate Against Identity Provider** assertion. Find the assertion under Policy Assertions/Access Control.



Drag the Authenticate Against Identity Provider assertion and drop it directly on the Stop Processing assertion.

4. A "Change Authentication Identity Provider" dialog box asks you to select an Identity Provider. Select the **OAuth SAML Identity Provider** FIP you created. Click **OK**.
5. Right-click the Request: Authenticate against OAuth SAML Identity Provider. Choose **Select Target Message**. Select **Other Context Variable** and type "bearerToken". Click **OK**.

6. Verify that the assertion label is now:  
\${bearerToken}: Authenticate against OAuth SAML Identity Provider

```

13  Comment: ====== Validate the trusted issuer by adding a FIP hei
14  Comment: A FIP has to be introduced in order to accept only trusted client
15  Comment: Remove the Stop assertion once the FIP is configured
16  Stop Processing
17  ${bearerToken}: Authenticate against OAuth SAML Identity Provider
18  Comment: ======
19  Comment: ====== Get Subject Name ID from SAML token =====

```

7. Disable or delete the Stop Processing assertion.

8. **Save** the policy.

## User Authentication Options

The policy responsible for user authentication is **OTK User Authentication** located at OTK-version /Policy Fragments/authentication. By default, username password authentication is supported. However, you can configure support for the following optional authentication mechanisms:

- [Authenticate against a Custom Identity Provider \(see page 44\)](#)
- [Authenticate against CA SiteMinder \(see page 44\)](#)

Customization of the OTK User Authentication policy to support these mechanisms is optional.

## Authenticate against a Custom Identity Provider

To authenticate against a custom identity provider:

1. In Policy Manager, click the Identity Provider tab and create the identity provider.
2. In the **OTK User Authentication** policy fragment, disable any branches that you do not intend to support.
3. Search for **Request: Authenticate against Internal Identity Provider**.
4. Double-click and select the custom identity provider.
5. **Save** the policy.

If you change the authentication to use a custom identity provider rather than the Internal Identity Provider, further modification may be required.

For example, if a policy is modified not to use the Authenticate Against Identity Provider or Authenticate User or Group assertions, the value of the \${request.authenticatedUser} context variable (which captures the name of the authenticated user) requires updating.

## Authenticate against CA SiteMinder

You must have an existing SiteMinder installation running and configured to work with the CA API Gateway. For configuration details, refer to SiteMinder information in the CA API Gateway documentation.

In the Policy Manager, in the OAuth folder under **Policy Fragments > authentication**:

1. Open the OTK User Authentication policy fragment.
2. Search for the comment: "Enable this block if a SiteMinder configuration has been configured and update appropriately". Right-click the disabled assertion folder under the comment and select **Enable Assertion**.
3. Within the policy fragment, search for the "Check Protected Resource Against CA Single Sign-On (SiteMinder)" assertion. Configure the assertion parameters to match those registered in the SiteMinder Configuration Properties.

No further configuration for the OTK User Authentication policy fragment is required. However, this policy can be customized. By default, the username context variable string is set as: \${siteminder.smcontext.attributes.ATTR\_USERNAME}.

You can set \${current.username}, by replacing ATTR\_USERNAME with any of the SiteMinder variables shown in the following table.

Variable	Notes
ATTR_USERDN	The user's distinguished name as recognized by SiteMinder.
ATTR_USERUNIVERSALID	This is the user's universal ID. It could be the name from the LDAP.
ATTR_AUTH_DIR_OID	The object ID of the directory where the user has been authenticated.
ATTR_AUTH_DIR_NAME	

Variable	Notes
	The name specification of the directory where the user has been authenticated
ATTR_AUTH_DIR_SERVER	The server specification of the directory where the user has been authenticated
ATTR_AUTH_DIR_NAMESPACE	The namespace specification of the directory where the user has been authenticated.
ATTR_USERNAME	The user's display name.

## Post-Installation Tasks

The following post-installation tasks are required:

- [Restart the Gateway \(see page 45\)](#)
- [Set the Database Type \(see page 45\)](#)
- [Import the Public Certificate \(see page 46\)](#)
- [Set a UUID Value for CookieKey \(see page 46\)](#)
- [Configure ID\\_TOKEN Attributes \(see page 47\)](#)

## Restart the Gateway

After installing the OAuth Solution Kit, you must restart the CA API Gateway.

## Set the Database Type

The database type is identified by the dbsystem context variable. The default setting for dbsystem is "mysql".

To set the database type:

1. Open the OTK Storage Configuration encapsulated assertion located in OTK-version/Policy Fragments/configuration.
2. Set the dbsystem context variable to one of the following values:
  - oracle
  - mysql
  - cassandra

## Import the Public Certificate

The Gateway needs to trust its own SSL certificate before you can use the test clients. To do this, import the Gateway public certificate into the certificate store of the Gateway.

To import the public certificate:

1. In the Policy Manager, choose **Tasks, Certificates, Keys, and Secrets, Manage Certificates**.  
The Manage Certificate dialog appears.
2. Click **Add**. The Add Certificate Wizard appears.
3. Select **Retrieve via SSL Connection (HTTPS or LDAPS URL)** and enter `https://hostname:8443` where hostname is the name of your server.
4. Click **Next**.  
The certificate details are displayed.
5. Click **Next**.  
On the Specify Certificate Options page, select the following:
  - Outbound SSL Connections
  - Signing Certificates for Outbound SSL Connections
  - Signing Client Certificates
  - Signing SAML Tokens
6. Click **Next**.  
Select **Certificate is a Trust Anchor**.
7. Click **Finish** and **Close**.

## Set a UUID Value for CookieKey

The context variable cookieKey is used to sign cookies in the oauth manager config policy.

Set Cookie configuration in oauth manager config policy fragments.

By default, the cookieKey value is set to "I\_HAVE\_TO\_BE\_CHANGED".

To set the UUID value for the cookieKey context variable:

1. Use an online UUID generator such as <https://www.uuidgenerator.net/> to create a UUID value.  
For example: 6f4874df-378a-4dde-bb7a-93c09b994ab0.

2. Navigate to OTK-version/Policy Fragments/configuration/oauth manager/  
Open the **oauth manager config** policy.
3. Search for cookieKey and set the value to the generated UUID value.
4. Click **Save and Activate**.

## Configure ID\_TOKEN Attributes

The ID\_TOKEN has attributes that require configuration.

To configure ID\_TOKEN attributes:

1. Navigate to OTK-version/Policy Fragments/configuration.  
Open the **OTK id\_token Configuration** policy.
2. Modify the following context variable:

Context Variable	Description
iss	The URL of your gateway that is issuing the id_token. For example: <a href="http://yourServer.com/connect">http://yourServer.com/connect</a>

3. Click **Save and Activate**.

## Verify the Installation

Verify the following components of your OAuth Tool Kit:

- [Run the OAuth 1.0 Test Client \(see page 48\)](#)
- [Run the OAuth 2.0 Test Client \(see page 49\)](#)
- [Verify the OAuth Infrastructure \(see page 53\)](#)

The test clients are used to verify installation changes and to access secured API endpoints of platforms.

Note the following security precautions when using the test clients:

- Do not install the test client on product systems.
- Do not install the test client on a Gateway that is available on the Internet.
- Modify the test client to use your own specific client credentials.
- Remove the test client from the OAuth Manager when it is no longer needed.

# Run the OAuth 1.0 Test Client

The test client is used to verify installation changes and to access OAuth 1.0/1.0a-secured API endpoints of platforms. This section describes how it works and how it can be configured.

- [Security Precautions \(see page 48\)](#)
- [Run the Test Client \(see page 48\)](#)
- [Further Configuration \(see page 48\)](#)

## Security Precautions

Note the following security precautions when using the test client:

- Do not install the test client on product systems.
- Do not install the test client on a Gateway that is available on the Internet.
- Modify the test client to use your own specific client credentials.
- Remove the test client from the OAuth Manager when it is no longer needed.

## Run the Test Client

To run the test client, open a browser and navigate to this URL:

`https://<Gateway_host>:8443/oauth/v1/client`

The OAuth V1 Client welcome screen is displayed.

The following table describes each setting on the test client.

Setting	Description
<b>Your name</b>	The client will use this name to associate received tokens with the current user. This can be any name; it is not verified and is easily changed. This makes it useful if more than one developer is using the client at the same time. The default user name "system" is used if none was provided.
<b>AccessReso urces</b>	The client will request a <i>request_token</i> at the authorization server.
<b>Reuse existing access_tok en if it exists?</b>	Select this check box to reuse an <i>access_token</i> for the current <i>consumer_key</i> , if the user has already received one from an earlier session. If the token exists, the client will directly access the resources. Clear this check box to issue a new <i>access_token</i> instead of reusing an existing one.

## Further Configuration

The following configurations can be modified:

- Application name
- consumer\_key

- consumer\_key\_secret
- resource endpoint
- request, authorize and token endpoint

All modifications can be made in the policy using the Policy Manager in the /oauth/v1/client policy.

## Run the OAuth 2.0 Test Client

The test client is used to verify installation changes and to access OAuth 2.0-secured API endpoints of platforms. This section describes how it works and how it can be configured.

- [Security Precautions \(see page 49\)](#)
- [Enable the Client \(see page 49\)](#)
- [Run the Client \(see page 50\)](#)
- [Restrict the OAuth 2.0 Grant Types \(see page 52\)](#)
- [Change Resource Owner Authentication \(see page 52\)](#)
- [Add SLA Rules \(see page 53\)](#)

### Security Precautions

Note the following security precautions when using the test client:

- ▪ Do not install the test client on product systems.
- Do not install the test client on a Gateway that is available on the Internet.
- Modify the test client to use your own specific client credentials.
- Remove the test client from the OAuth Manager when it is no longer needed.

### Enable the Client

To enable the OAuth 2.0 test client:

1. Navigate to the following URL in a browser:  
`https://<Gateway_host>:8443/<prefix>/oauth/manager`
2. Log in with an administrator account:  
Username: Admin  
Password: <yourAdminPassword>
3. Click **Clients**.  
No clients appear? See the troubleshooting section below.
4. Click **List Keys** of the client with the name "**TestClient2.0**"
5. Click **Edit**

6. Edit **Callback URL** and replace "<YOUR\_SSG>" with the protocol, hostname, port, and prefix of your Gateway; for example: <https://acmecorp.com:8443/myPrefix>  
Note that there are two replacements to be performed.

7. Click **Save**.

8. Click **Client > List Keys** and verify that the client key has been updated.

Click **Clients** to list, delete, and register client applications.

Click **Tokens** to list, and revoke access tokens granted to client applications.

## Troubleshooting

When you click Clients, no clients are listed? The most common reasons are:

- You must sign in with the username "Admin".

- The OTK is integrated with the CA API Portal.

In this case, clients are managed in the API Portal, not the OAuth Manager. You can disable API portal integration by opening the OTK Client DB Get policy and setting `usePortal` to false. For more information, see [Manage API Keys with CA API Portal \(see page 91\)](#).

## Run the Client

To run the OAuth 2.0 test client:

1. Navigate to the following URL in a browser:

[https://<Gateway\\_host>:8443/<prefix>/oauth/v2/client/authcode](https://<Gateway_host>:8443/<prefix>/oauth/v2/client/authcode)  
The OAuth Client Test Application screen is displayed.

2. Navigate between the other OAuth 2.0 Test Clients:

- Authorization Code
- Implicit
- Resource Owner Password Credentials
- Client Credentials
- SAML

Each OAuth 2.0 Test Client tests its own grant type. If you are only using a subset of the available OAuth grant types, you can ignore the other test clients.



### Known Issue

There is a known issue with the **OAuth V2 Clients** navigation tab.

If your OAuth Toolkit is installed with a prefix, a browser error occurs when you click the the OAuth V2 Clients page tab. No clients are displayed and a "service not found" error appears. The URL is missing the prefix value.

Workaround: Manually add the prefix to the URL and refresh.

Each client app maintains its own token. Each time you initiate a new OAuth session, the current access token is overwritten.

The access token in memory is used to call the API of your choice. In the case of SAML, a SAML token is also maintained in memory and overwritten each time you initiate a new one.

The screenshot shows the CA technologies Authorization Test Clients interface. At the top, there are tabs for 'Authorization Test Clients', 'OAuth V2 Clients', 'OAuth V1 Client', and 'OpenID Connect Clients'. Below these, a black bar contains five buttons: 'AUTHORIZATION CODE' (selected), 'IMPLICIT', 'RESOURCE OWNER PASSWORD CREDENTIALS', 'CLIENT CREDENTIALS', and 'SAML BEARER'. A red header 'Grant type: Authorization code' is displayed above a text input field labeled 'Current Access Token:'. Below this is a horizontal line. Underneath the line, there are two buttons: 'INITIATE' and 'REFRESH'. Further down, another button 'CALL API Using Current Access Token' is shown above a text input field 'Target: https://yourGateway.com:8443/oauth/v2/protectedapi/resourcefoo' and a blue 'CALL API' button.

## Get an Access Token

To get an access token before calling an API:

1. Click any of the OAuth Test clients identified by grant type on the black bar.
2. Click **Initiate**. The OAuth 2.0 Authorization Server page is displayed. (Resource Owner Password Credentials skips this step).
3. Enter your credentials and then click **Grant**. You will be redirected back to the client application with an access token and a refresh token.

## Test the Client

To test using the access token to call an API on the CA API Gateway:

1. Enter a target URL in the **Target** field.
2. Click **Call API**. The client app will use the access token currently residing in memory as a credential to call the target API.
3. View the response for this call below the **Target** field.

## Refresh a Token

Certain grant types support refresh tokens. This is indicated by the presence of a **Refresh** button.

To refresh an existing OAuth access token, click the **Refresh** button.

## Clear the Current Session

Click the **Clear Session** button on the OAuth client page. This starts a new test and clears all the parameters in the clients.

## Restrict the OAuth 2.0 Grant Types

By default, the OAuth Authorization Server enables the following OAuth grant types:

- Authorization code
- Implicit
- Client Credentials
- Resource owner password credentials
- SAML

To restrict the different grant types that you want to support for your use cases, disable the branches that implement the ones you do not wish to support in the token endpoint policy. Set this to "At least one assertion must evaluate to true" with the comment "grant types" in the endpoint /auth/oauth/v2 /token.

In the Policy Manager, click **Show Comments** in the policy tool bar to see comments in the policy window.

## Change Resource Owner Authentication

The authorization and token endpoint policies authenticate the resource owner. In the initial installation, the Internal Identity Provider is used to achieve this authentication. You may want to attach a different authentication source for your purpose.

For more information, refer to the following sections in [wiki.ca.com/gateway](https://wiki.ca.com/gateway) (<https://wiki.ca.com/gateway>):

- Authenticate User or Group Assertion
- Working with Identity Providers

If you change the way resource owner authentication is done by either the OAuth authorize or token endpoint policies, there may be downstream consequences. By default, these policies are set to authenticate against the Gateway's Internal Identity Provider using the Authenticate Against Identity Provider assertion. This assertion sets the \${request.authenticatedUser} context variable at runtime and this variable is subsequently used by both the authorize and token endpoint policies to set another context variable representing the resource owner. If you change the resource owner authentication to another method that does not set the \${request.authenticatedUser} context

variable, then the OAuth authorize and token endpoint policies must be adjusted to set the resource owner context variables appropriately. (Tip: Use the Find command (Ctrl+F) in the policy window to search for "authenticatedUser" to find where these changes may need to be made.)

## Add SLA Rules

In addition to authenticating these various identities, you may want to add SLA rules associated with their use of the Authorization Server.

The following example shows the Apply Throughput Quota assertion being used to prevent a client from refreshing the access token more than 10 times per second.

The screenshot shows a policy editor interface for the 'oauth/v2 [/auth/oauth/v2/token] (v204, active)' policy. The policy tree on the left lists various assertions and configurations. A red box highlights a specific section of code starting at line 194:

```

194  Comment: ===== Set Context Variable scope as String to: ${request.http.parameter.scope}
195  Compare Expression: ${[request.http.parameter.grant_type]} is equal to refresh_token
196  Comment: ===== Each client can refresh access token maximum of 10 times/second =====
197  At least one assertion must evaluate to true
198    Apply Throughput Quota: ${client_id}-refresh_token: 10 per second
199      All assertions must evaluate to true
200        400 Return Template Response to Requestor
201          Response: Add Header Pragma: no-cache
202          Response: Add Header Cache-Control: no-store
203          Stop Processing
204
205  At least one assertion must evaluate to true load session
206  At least one assertion must evaluate to true Validate refresh token

```

This section of code uses the 'Compare Expression' assertion to check if the grant type is 'refresh\_token'. It then applies a throughput quota of 10 refresh tokens per second for the client ID. If the quota is exceeded, it returns a 400 error response with 'no-cache' and 'no-store' headers and stops processing.

For more information, see the Apply Throughput Quota assertion description in [wiki.ca.com/gateway](https://wiki.ca.com/wiki.ca.com/gateway) (<https://wiki.ca.com/wiki.ca.com/gateway>).

## Verify the OAuth Infrastructure

The following instructions are for OAuth 2.0 installations only.

### Verify the OAuth Infrastructure

1. Navigate to <https://<your-ssg>:8443/<Prefix>/oauth/v2/client/bcp>  
The browser should display a simple OpenID Connect test client.

2. Click **Send**.

The OAuth 2.0 Authorization Server (shown at right) should appear.  
If not, an error message lets you know where a problem has occurred.



## OAuth 2.0 Authorization Server

Mismatching redirect uri. Given: 'https://[REDACTED]:8443/oauth/v2/client/bcp?auth=done'

3. In the table listing client properties, verify that the scope values do not include "oob".  
If "oob" is listed as a scope type, you must restart the CA API Gateway, then restart this verification procedure:

```
service ssg restart
```

4. Enter valid gateway user credentials and click **Grant**.  
After being directed back to the Open ID Connect Test Client page, click either the **Resources** or **Claims** buttons.  
A JSON result should be displayed.

# Upgrade and Uninstall

## Uninstall Previously Installed Solution Kits

The Solution Kit manager allows you to uninstall the entire OTK Solution kit, or individual solution kits within the parent solution kit.

To uninstall solution kits:

1. Go to **Tasks, Extensions and Add-Ons, Manage Solution Kits**.
2. Click to select the component you want to uninstall. If you want to remove the entire OTK, select the top level OTK Solution Kit.
3. Click **Uninstall** and confirm.
4. Click **OK** when the task is completed.

The uninstall process leaves empty folders in the policy manager.

To remove the empty folders:

1. In Policy Manager, locate the root folder that you want to uninstall.
2. Right-click the folder in policy manager and select **Delete Folder**.
3. Click the check box to activate the OK button, then click **OK**.  
Any empty sub-folders are also deleted.

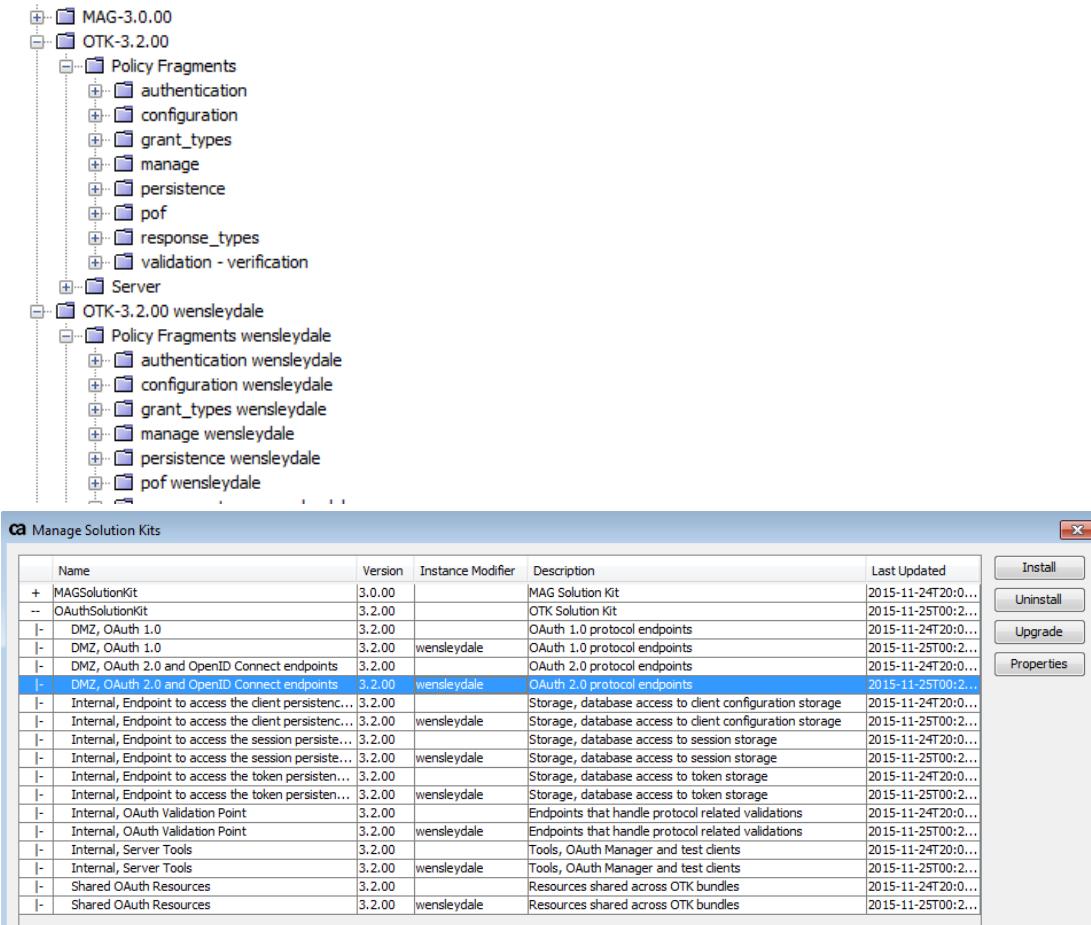


When you uninstall all components, the Shared OAuth Resources solution kit may remain. If this occurs, manually select the Shared OAuth Resources solution kit and click Uninstall.

## Uninstalling an Entire Solution Kit with Instance Modifiers

If the OTK solution kit has been installed with an instance modifier, you must uninstall each child solution kit individually.

Version 3.2 is shown below.



## Uninstall Installations Prior to OTK Version 3.2

OTK versions prior to OTK version 3.2 were not installed using the solution kit installer. Subsequently, uninstalling the OAuth Toolkit is a manual task.

To uninstall OTK versions that were not installed as solution kits :

1. Login into Policy Manager as an administrator.
2. Select the main OAuth folder.
3. Right-click and select **Delete Folder**.
4. Click the check box to activate the OK button, then click **OK**.

If the folder is not deleted, delete components within the folder first, then delete the empty OTK folders.

# Upgrade an Existing OTK Installation

To upgrade an existing OTK installation:

- Run the database upgrade scripts.
- Install the solution kit using a unique Instance Modifier.
- Merge any customization from the old policies into the new policies.
- Modify the API endpoint paths for clients. The new paths must include the new Instance Modifier value.

The instance modifier value must be different than any prefix value used for the prior installation.



Do not click **Upgrade** from the solution kit installer. A critical issue exists with this functionality.

Instead, perform an upgrade by providing an Instance Modifier and clicking **Install**.

# Prepare JSON Message for Export

Exporting the current OAuth configuration in JSON format provides convenient policy-driven setting of attributes for communication between the CA API Gateway and the OAuth Client. The exported JSON message provides a snapshot of the values assigned to context variables. It also includes endpoint paths and certificate information.

Configuration is exported for a specific client id.

Perform the following tasks to prepare the JSON content and make it accessible:

- [JSON Export Endpoint \(see page 59\)](#)
- [JSON Message Example \(see page 60\)](#)

# Include the OAuth Server Certificate

The OAuth server certificate is the SSL certificate on the server. The certificate name must match the oauth2\_server\_certificate variable in the policy used to create the JSON message.

To include the OAuth server certificate in the JSON message:

1. In Policy Manager, go to **Tasks, Certificates, Keys and Secrets, Manage Certificates**
2. If the SSL certificate is already listed, copy the Certificate Name value.  
If the certificate is not listed, you must create one by clicking **Add**.
3. Open the OTK Variable Configuration policy.
4. Assign the certificate is value to the oauth2\_server\_certificate variable. See [Assign OAuth Server Details \(see page 58\)](#).

The name used in the JSON content variable must match the alias of the certificate.

# Assign OAuth Server Details

To assign OAuth server details:

1. Open the OTK Variable Configuration policy.
2. Locate the Export section shown below.

```

22 | // Comment: PLEASE REVIEW THE VALUES BELOW AND CONFIGURE THEM TO YOUR NEEDS
23 | // Export All assertions must evaluate to true Configure values related to the OAuth Manager client_id export feature
24 | ✓ Set Context Variable oauth2_server_hostname as String to: com Match the hostname that will be used by
25 | ✓ Set Context Variable oauth2_server_certificate as String to: com Match the name of the cert that is used
26 | ✓ Set Context Variable oauth2_server_port as String to: 8443 Match the port through which oauth clients connect
27 | ✓ Set Context Variable oauth2_server_url_prefix as String to empty Set the url prefix if a prefix has been used when the
28 | ✓ Set Context Variable expose_client_secret as String to: false
29 | ✓ Set Context Variable enable_anonymous_client_export as String to: false
30 | ⚡ Comment: Modify the value below ONLY if the path has been modified
31 | ✓ Set Context Variable userSession_logout_path as String to: /connect/session/logout
32 | ✓ Set Context Variable userinfo_path as String to: /openid/connect/v1/userinfo
33 | ✓ Set Context Variable userSession_status_path as String to: /connect/session/status
34 | ⚡ Comment: The content below will be added to the configuration export JSON message as is
35 | ✓ Set Context Variable export_custom as String to: "oauth_demo_protected_api_endpoint_path": "/oauth/v2/protect...

```

3. Assign values for the following variables found in the policy.

Context Variable	Notes
oauth2_server_hostname	Hostname of the OAuth server.
oauth2_server_certificate	OAuth server certificate name. This name must match the assigned Certificate Name value found in <b>Tasks, Certificates, Keys and Secrets, Manage Certificates</b> .
oauth2_server_port	Port of the OAuth server.
oauth2_server_url_prefix	Prefix of the OAuth server.

4. Optionally configure the OAuth Server behavior.

Context Variable	Type	Notes
expose_client_secret	Boolean	Default: false. If true, the client secret is included in the exported JSON message.
enable_anonymous_client_export	Boolean	Default: false. If true, the endpoint allows access to users without authentication.

## JSON Export Endpoint

### Enable the Endpoint

The endpoint that provides MAG configuration values in JSON format is located in the OAuth/DMZ /OAuth 2.0 folder:

/auth/oauth/v2/client/export



By default this endpoint is disabled. The preferred method of exporting the JSON file is through the Export button in OAuth Manager. You do not need to enable this endpoint to export the JSON file through OAuth Manager.

To enable the server endpoint:

1. Locate the /auth/oauth/v2/client/export endpoint. By default, this is located in Server/DMZ /OAuth 2.0/
2. Right-click the endpoint, and select **Service Properties**.
3. Select **Enable** and click **OK**.

## Configure Endpoint Access

By default the JSON export endpoint allows access to users authenticated using HTTP Basic Authentication only.

To allow anonymous access to the JSON export endpoint:

1. Open the **OTK Variable Configuration** policy:  
OAuth/Policy Fragments/configuration/OTK Variable Configuration
2. Set the enable\_anonymous\_client\_export\_endpoint context variable to true.

Both authorized and anonymous users require a secure SSL connection to access the endpoint.

## JSON Message Example

The JSON message contains the current server configuration values used to initialize the SDK for a client. The message comprises four sections:

- [Server \(see page 60\)](#)
- [MAG \(see page 61\)](#)
- [OAuth \(see page 62\)](#)
- [Custom \(see page 64\)](#)

### Server

The server section contains server details and the SSL certificate required to establish communication between the SDK and the MAG. It also contains the OAuth server SSL certificate information in the server\_certs attribute. The name of the certificate is identified in the OTK Variable Configuration policy. See [Prepare JSON Message for Export \(see page 58\)](#).

**JSON Server Section Example**

```

"server": {
    "hostname": "example.com",
    "port": 8443,
    "prefix": "myPrefix",
    "server_certs": [
        [
            "-----BEGIN CERTIFICATE-----",
            "MIIC9TCCA...","bi5...","aW4....",
            "-----END CERTIFICATE-----"
        ]
    ]
}

```

## MAG

MAG and the OAuth Manager Extension must be installed for the Mobile API Gateway section to be included in the JSON configuration file.

**JSON MAG Section Example**

```

"mag": { "system_endpoints": {
    "device_register_endpoint_path": "/connect/device/register",
    "device_remove_endpoint_path": "/connect/device/remove",
    "client_credential_init_endpoint_path": "/connect/client/initialize" },
  "oauth_protected_endpoints": {
    "enterprise_browser_endpoint_path": "/connect/enterprise/browser",
    "device_list_endpoint_path": "/connect/device/list" },
  "mobile_sdk": {
    "sso_enabled": true,
    "location_enabled": true,
    "location_provider": "network",
    "msisdn_enabled": true,
    "trusted_public_pk": true,
    "trusted_cert_pinned_public_key_hashes": [],
    "client_cert_rsa_keybits" : 1024
  }
}

```

No explicit configuration of server variables for export is required. However, you can override existing values by editing the MAG Variable Configuration policy found in /Policy Fragments/configuration.

Exported Attributes	Description	Type
device_register_endpoint_path	URL suffix for device registration endpoint	String
device_remove_endpoint_path	URL suffix for token server's remove_device_x509 endpoint	String

Exported Attributes	Description	Type
client_credential_i	URL Suffix for initialize client credential endpoint	String
nit_endpoint_path		
enterprise_brows	URL suffix for server's enterprise apps endpoint.	String
er_endpoint_path		
device_list_endpo	URL suffix for device list	String
int_path		
sso_enabled	Indicates whether single sign on is allowed for this app.	Boolean
location_enabled	Indicates whether location information is allowed in outbound requests	Boolean
location_provider	The location provider to use if location is enabled	String
msisdn_enabled	MSISDN information should be included in the outbound requests	Boolean
trusted_public_pkki	Indicates whether public CAs recognized by the OS are accepted as TLS server certificates in addition to the list returned by server_certs	Boolean
trusted_cert_pinn	Controls whether TLS server certificate public key pinning is in ed_public_key_ha use and, if so, what pinned public key hashes to permit within shes server cert chains	JSON Array of String
client_cert_rsa_ke	The size in bits of the RSA keypair to generate for the client certificate	Number
ybits		

## OAuth

For the OAuth section to be included in the JSON configuration file, OAuth Manager must be installed.

### JSON OAuth Section Example

```
"oauth": {
    "client": {
        "organization": "CA Technologies",
        "description": "Example application for Mobile SSO demonstrations",
        "client_name": "AppA",
        "client_type": "confidential",
        "registered_by": "admin",
        "client_ids": [
            {
                "client_id": "12341234b4f-aaaa-aaab-aaab-123412345377a",
                "client_secret": "abababa25-1239-1232-1232-12345678999a",
                "scope": "openid mssso phone profile address email mssso_register",
                "redirect_uri": "https://android.ssosdk.ca.com/android",
                "environment": "Android"
            }
        ]
    }
}
```

```

        "status": "ENABLED",
        "registered_by": "admin",
        "service_ids": "",
        "account_plan_mapping_ids": "",
        "client_key_custom": "{}"
    }
]
},
"system_endpoints": {
    "authorization_endpoint_path": "/auth/oauth/v2/authorize",
    "token_endpoint_path": "/auth/oauth/v2/token",
    "token_revocation_endpoint_path": "/auth/oauth/v2/token/revoke",
    "usersession_logout_endpoint_path": "/connect/session/logout",
    "usersession_status_endpoint_path": "/connect/session/status"
},
"oauth_protected_endpoints": {
    "userinfo_endpoint_path": "/openid/connect/v1/userinfo",
    "usersession_status_endpoint_path": "/connect/session/status"
}
}

```

You are required to configure certain variables in the OTK Variable Configuration policy. See [Prepare JSON Message for Export \(see page 58\)](#).

Open the OTK Variable Configuration policy found in the OAuth/Policy Fragments/configuration/ to further customize values for the JSON message.

Exported Attributes	Description	Type
organization	The organization name to include in the client cert DN	String
description	App description	String
name	Client name	String
type	Client type	String
registered_by	User who registered the client	String
client_ids	List of client ids, only the first client_id will be used by the SDK and the rest will be ignored by the SDK	JSON Array
client_id	The application's client id for the initial OAuth token request	String
client_secret	The application's client secret for the initial OAuth token request	String
scope	The OAuth scope string that should be requested when obtaining an access token that will be used to consume service from an API endpoint	String
redirect_uri	The redirect URI provided to the third-party-login platform.	String
environment	The environment of the client	String
status	Status of the client	String
registered_by	User who registered the client	String
authorization_end_point_path	System endpoint for authorization	String
	System endpoint to acquire token	String

Exported Attributes	Description	Type
token_endpoint_path	System endpoint to revoke token	String
usersession_logout_endpoint_path	System endpoint to perform logout user session	String
userinfo_endpoint_path	Endpoint to retrieve user info	String
usersession_status_endpoint_path	Endpoint to check user session status	String

## Custom

The custom section allows you add additional attributes to pass values to the SDK.

### JSON Custom Section Example

```
"custom": {
    "oauth_demo_protected_api_endpoint_path": "/oauth/v2/protectedapi/foo",
    "mag_demo_products_endpoint_path": "/protected/resource/products"
}
```

To add new attributes to the Custom section of the JSON content, set the `export_custom` variable value in the OAuth Variable Configuration policy.

# Using the OAuth Manager

The OAuth Manager displays information about registered OAuth clients and associated OAuth tokens used to access OAuth protected resources.

The OAuth Manager provides the following functions:

- [Log into OAuth Manager \(see page 65\)](#)
- [Register a Client \(see page 65\)](#)
- [Environment and Scope \(see page 66\)](#)
- [Callback URI \(see page 67\)](#)
- [Manage Clients \(see page 67\)](#)
- [List Client Keys \(see page 68\)](#)
- [Manage Tokens \(see page 70\)](#)

## Log into OAuth Manager

To log into OAuth Manager:

1. Open a browser and go to:  
`https://<yourgatewayURL>:8443/prefix/oauth/manager`
2. Provide a username and password. The type of access you are granted depends on your user role. See [OTK User Role Configuration \(see page 97\)](#).
3. Click **Clients** to list, delete, and register OAuth clients.

## Register a Client



REGISTER A NEW CLIENT

To register a client:

1. Open the OAuth Manager.
2. Click **Clients**.
3. Click **Register a New Client**.
4. Provide values for the client fields, then click **Register**.

The registration page fields are used for both OAuth 1.0 and OAuth 2.0 clients.

## Master Key

This setting applies only to clients using the CA Mobile API Gateway (MAG).

A master key is a special client\_id (also referred to as key, client\_key, oauth\_consumer\_key).

Behavior is as follows:

- Only master keys can be used to access the /connect/client/initialize API used to initialize the Mobile SDK with the MAG server. The API endpoint issues unique client credentials (non master keys)
- Only non master keys can request OAuth tokens.
- If a master key is deleted, all keys that were issued based on that master key are also deleted. Any mobile app configured with a deleted key is disabled.

Click the check box to indicate the key for this client is a master key. The option to specify a client secret is greyed out. The OAuth Manager automatically sets the client secret to the client\_id value, which indicates this key is a master key.

## Environment and Scope

The **Environment** value is not validated by the OTK by default. Set the value to the client's associated platform such as iOS, Android, or Web. You can enter the environment value to filter search results from the List Keys page. Additionally, you can customize OTK policies to take advantage of environment information during OAuth related requests.

The OAuth **Scope** is a space separated list of values that apply to OAuth 2.0 clients only. The scope limits the authorization granted to the client by the resource owner. For OAuth 1.0 clients, the scope defaults to "oob".

If a client sends a token request and includes scope, the OTK can only issue scope values that have been registered for that client. If none of the requested scope values match the registered values, the request fails. An OAuth protected API can require specific scope values. Any access\_token used at that API must be granted for all required scope values, otherwise the request fails. If no scope is registered and no scope requested, scope is set to 'oob'.

Default supported scope values are as follows:

Scope	Notes
Value	
openid	Enables clients to send requests to the /userinfo endpoint. Additionally this SCOPE causes the server to issue an id_token which can be used within the context of Mobile SSO.
address	Returns address information of the current user. Must be requested with 'openid'
phone	Returns telephone information of the current user. Must be requested with 'openid'

Scope Notes
Value
email Returns email information of the current user. Must be requested with 'openid'
profil Returns profile information of the current user. Must be requested with 'openid'
e
user_r Returns the role of the resource_owner. By default it will be <b>user</b> or <b>admin</b> . The role <b>admin</b> is used in the OAuth Manager and MAG Manager to identify an administrator. It has to be requested with 'openid'. This SCOPE is an OTK extension; it is not part of OpenID Connect.

## Callback URI

The callback\_uri is also known as the redirect\_uri in OAuth 2.0.

The OTK uses the callback URI value to return response parameters back to a requesting client. This occurs if the client uses one of the following token request parameters:

- oauth\_callback (OAuth 1.0 clients)
- redirect\_uri (OAuth 2.0 clients)

For OAuth 2.0, the redirect\_uri parameter may be used with response types

The value of the callback\_uri is a comma separated list of absolute URLs. You must include the scheme.

Valid callback_uri	Invalid callback_uri
https://example.ca.com/ ( <a href="https://example.ca.com/">https://example.ca.com/</a> )/callback,https://another-callback. example.ca. com/granted	example.ca. com
https://example.ca.com:9876/callback?key=value	
myscheme://for.my.mobile.native.app	

## Manage Clients

In the OTK, clients must be registered in order to request oauth tokens and consume oauth protected APIs. The Manage Clients page allows you to add, edit, or delete clients.

## CA API Management OAuth Toolkit - 3.4

This is an overview of all registered client applications.

Filter search results

index	client_ident	name	type	description	organization	registered_by	created	custom	action
1	mssso-clientAppA	AppA	confidential	Example application for Mobile SSO demonstrations	CA Technologies	admin	0		<button>DELETE</button> <button>EDIT</button> <button>LIST KEYS</button>
2	3e74-mag-test-mssso-clientAppB	AppB	confidential	Example application for Mobile SSO demonstrations	CA Technologies	admin	0		<button>DELETE</button> <button>EDIT</button> <button>LIST KEYS</button>

**Warning:** Client management is not available via the OAuth Manager if the CA OAuth Toolkit has been integrated with the CA API Portal.

### Available Actions

action
<button>DELETE</button>
<button>EDIT</button>
<button>LIST KEYS</button>

The available actions are:

- **Delete** – If you delete the client application, all client\_ids issued for the client are also deleted.
- **Edit** – Edit the client name, organization, description, and client type.  
Also allows you to set a value in JSON format for the client\_custom field.  
Click **Update Client** to save any changes.
- **List Keys** – Displays individual client information.  
The table below shows additional information for selected fields.

## List Client Keys

All registered client\_keys for the given client application appear on the list client keys page.

Click **Add Client Key** to create additional keys for this client.

## Available Actions



Perform any of the following available actions for a specific client key:

- **Revoke** – Deletes the client key. All tokens for this client key are also revoked.
- **Edit** – Edit properties for the key such as changing the status, disabling the key, adding scopes, and providing a Callback URL.  
Disabling a client key prevents the client key from being used for any future tokens, however it does not disable the tokens for that client key.
- Allows you to set a value in JSON format for the client\_key\_custom field.
- Allows you to set the Service ID's and Account Plan Mapping ID's values used in the CA API Portal.
- **Disable Tokens** – Disables all tokens for the client key.
- **Export** – Used to export client information in JSON message format. Accesses the current server configuration values. Use the JSON message to initialize OAuth clients or the CA Mobile API Gateway SDK for mobile applications.

## Field Information

Field	Notes
<b>client_ident</b>	The identifier for the client.
<b>client_key</b>	In OAuth 1.0, the client_key is the oauth_consumer_key. In OAuth 2.0 the client_key is the client_id.
<b>secret</b>	The client secret. If the client key and the secret are the same value, the client key is used as the master key in other endpoints/policies.
<b>scope</b>	The allowed scope for the client.
<b>environment</b>	Identifies the client platform.
<b>callback</b>	In OAuth 2.0, call back is the redirect_uri. Multiple URIs are supported. In Oauth 1.0 it can either hold "oob" or a single valid URI.
<b>expiration</b>	The date until this key is valid. A value of 0 indicates the key never expires.
<b>status</b>	Either "ENABLED" or "DISABLED". Disabled tokens will cause resource requests to be denied
<b>client_key_cu</b>	A custom field associated with the client key.
<b>stom</b>	The custom value must be a valid JSON object and cannot contain the following characters: < > &
<b>serviceIds</b>	A comma separated list of key strings that identify API services registered with the CA API Portal.

Field	Notes
<b>accountPlan</b>	A comma separated list of key strings that identify account plans registered with the app. <b>appId</b> CA API Portal.

## Manage Tokens

To manage tokens using the OAuth Manager:

1. Open a browser and navigate to this URL:  
[https://<Gateway\\_host>:8443/oauth/manager](https://<Gateway_host>:8443/oauth/manager)  
The home page of the OAuth Manager appears.
2. Click **Tokens** to view values of issued tokens, and to disable or revoke tokens.

The table below shows additional information for selected fields.

Field	Description
<b>rtoken</b>	A <i>refresh_token</i> if available
<b>expiration</b>	The expiration date of the refresh token
<b>client_key</b>	In OAuth 1.0 it is the "oauth_consumer_key"; in OAuth 2.0 it is the "client_id"
<b>status</b>	Disabled tokens cause resource requests to be denied

## Available Actions



Perform any of the following available actions for a specific token:

- **Revoke** – Deletes the token.
- **Disable/Enable** – Toggles the current state of the token between invalid and valid.

# Secure an API Endpoint with OAuth

---

To add OAuth authorization to an existing API, perform one of the following tasks:

- For OAuth 1.0 authentication, use the OTK Require OAuth 1.0 Token policy fragment.
- For OAuth 2.0 authentication, use the OTK Require OAuth 2.0 Token encapsulated assertion.

You can add global OAuth Authorization rules directly within the Require OAuth 1.0 Token or Require OAuth 2.0 Token policy fragments. In this case, these rules apply the same way to all APIs that are protected with this OAuth fragment.

## OTK Require OAuth 1.0 Token

Use the OTK Require OAuth 1.0 Token policy fragment to allow access to an API only when a valid OAuth 1.0 access\_token is presented by the client. Place the fragment as early as possible in an API policy.

It takes the http request, including the OAuth header, and outputs the following:

- oauth\_consumer\_key
- oauth\_token
- oauth\_verifier
- authorizationHeader

The policy fragment includes the OTK Require OAuth 1.0 Parameter policy fragment located in OTK-version/Policy Fragments/authentication.

## OTK Require OAuth 2.0 Token

Use the OTK Require OAuth 2.0 Token encapsulated assertion to allow access to an API only when a valid OAuth 2.0 access\_token is presented by the client. It includes the OTK Access Token Retrieval assertion to find the incoming OAuth 2.0 access\_token.

Place the assertion as early as possible in an API policy.

The screenshot shows the 'Identity Providers' tab selected in the top navigation bar. A search bar contains the text 'require oau'. Below the search bar is a list of assertions. The 'OTK Require OAuth 2.0 Token' assertion is highlighted with a blue selection bar.

Drag the assertion into a policy and configure the properties shown in the table below.

Properties	Parameter	Type	Notes
Name			
Required SCOPE(s)	scope_req	String	If SCOPE is not required, this value can be empty.
	uired	g	A space separated list of required SCOPES. An access_token is only accepted if it has been granted with those SCOPE values.
Cache validation result (s)	cache_lifet ime	Integ er	This value cannot be empty. Represents the time in seconds for which an access_token is cached. The assertion initially validates an access_token. The validation result is then cached until the cache period expires. This increases performance, but also enables clients to use potentially expired access_tokens. The cache_lifetime value extends the lifetime of the token. A value of 0 indicates no caching is performed.
Is this a one-time access-token?	onetime	Bool	Default value: <b>false</b> . To allow an access_token to be considered valid only once for this endpoint, set this value to <b>true</b> . This setting is rare, but enables special use cases.
Fail if this SCOPE was granted?	scope_fail	Bool	Default value: <b>false</b> . Set this value to "true" if a request should fail in the case that an access_token has been granted for at least one the specified SCOPE values listed above
Access Token	given_acce ss_token	String	Optional. The hardcoded value of an access token or a context variable representing an access_token. Use this property if an access_token is made available, but not by the client, or if the access_token is passed using a non-standard mechanism.

The encapsulated assertion sets the following context variables:

Output	Type
access_token	string
content-type	string
error.code	string
error.msg	string

Output	Type
session.client_id	string
session.expires_at	string
session.scope	string
session.subscriber_id	string
status	string

# OAuth Request Scenarios

---

The following scenarios examine OAuth client requests for an access\_token.

- [Requests Specifying response\\_type \(see page 74\)](#)
- [Requests Specifying grant\\_type \(see page 78\)](#)

## Requests Specifying response\_type

The following scenarios describe requests for access\_token using a specified response\_type:

- [response\\_type=token \(see page 74\)](#)
- [response\\_type=token id\\_token \(see page 75\)](#)
- [response\\_type=code \(see page 77\)](#)

### response\_type=token

The 'token' response\_type can be used when the client should not have access to user credentials. It includes a redirect that involves a browser or a web view on a mobile device. Using the token response\_type is not secure. It should be used only if an exposed access\_token is not an issue. Clients using this response\_type are considered to be 'public' clients and do not receive a refresh\_token. This type of client may be implemented in JavaScript.

Request
<b>Me</b> GET <b>tho</b> <b>d:</b>
<b>En</b> /auth/oauth/v2/authorize <b>dp</b> <b>oin</b> <b>t:</b>
<b>Par</b> response_type=token&client_id=a-client_id&redirect_uri=a-redirect_uri&scope=a-list-of-scope- <b>am</b> values&state=a-state-value <b>ete</b> <b>rs:</b>
<b>Op redirect_uri:</b> If provided only requests using a registered redirect_uri of this client will be granted <b>tio</b> by the OAuth server. If the parameter is not included the OAuth server will use the registered <b>nal</b> redirect_uri. If multiple redirect_uris have been registered the request will fail. At least one <b>: redirect_uri</b> MUST have been registered!
<b>Op scope:</b> Only SCOPE values that have been registered for the client will be granted by the OAuth <b>tio</b> server <b>nal</b> <b>:</b>
<b>state:</b> This value is opaque to the OAuth server and will be passed back unmodified to the client

**Request**

```
Op
tio
nal
:
```

---

**Response**

**He** status: 200

```
ad
er:
```

---

**He** content-type: text/html

```
ad
er:
```

---

**Bo** The user-agent will receive a login page. This page will request user credentials and the consent of **dy** the user. If the user denies the request the client will receive an error. If the user grants the client : it will receive the access\_token attached to the redirect\_uri

**Ne** The OAuth server will redirect the user-agent back to the client:

```
xt:
```

---

**He** 302

```
ad
er:
```

---

**He** Location: the-redirect-uri?state=the-given-state#access\_token=an-  
**ad** access\_token&expires\_in=lifetime-in-seconds&token\_type=Bearer&scope=granted-scope  
**er:**

---

The receiving user-agent (browser, JavaScript client) can now extract the parameters from the redirect\_uri fragment portion. The fragment value will only be available in the browser.

## response\_type=token id\_token

The 'token id\_token' response\_type can be used when the client should not have access to user credentials. It includes a redirect that involves a browser or a web view on a mobile device. Using the token response\_type is not secure. It should be used only if an exposed access\_token is not an issue. Clients using this response\_type are considered to be 'public' clients and do not receive a refresh\_token. This type of client may be implemented in JavaScript.

**Request**

```
Me GET
tho
d:
```

---

**En** /auth/oauth/v2/authorize

```
dp
oin
t:
```

---

---

**Request**

**Par** response\_type=token%20id\_token&client\_id=a-client\_id&redirect\_uri=a-redirect\_uri&state=a-am state-value&scope=a-list-of-scope-values (SCOPE MUST be included and it MUST include **ete** 'openid')

**rs:**

**Op redirect\_uri:** If provided only requests using a registered redirect\_uri of this client will be granted **tio** by the OAuth server. If the parameter is not included the OAuth server will use the registered **nal** redirect\_uri. If multiple redirect\_uris have been registered the request will fail. At least one : redirect\_uri MUST have been registered!

**Op state:** This value is opaque to the OAuth server and will be passed back unmodified to the client

**tio**

**nal**

:

---



---

**Response**

**H** status: 200

**ea**

**de**

**r:**

**H** content-type: text/html

**ea**

**de**

**r:**

**B** The user-agent will receive a login page. This page will request user credentials and the consent of **o** the user. If the user denies the request the client will receive an error. If the user grants the client **dy** it will receive the access\_token attached to the redirect\_uri

:

**N** The OAuth server will redirect the user-agent back to the client:

**ex**

**t:**

**H** 302

**ea**

**de**

**r:**

**H** Location: the-redirect-uri?state=the-given-state#access\_token=an-  
**ea** access\_token&expires\_in=lifetime-in-seconds&token\_type=Bearer&scope=granted-  
**de** scope&id\_token=an-id-token-represented-as-jwt&id\_token\_type=urn%3Aietf%3Aparams%  
**r:** 3Aoauth%3Agrant-type%3Ajwt-bearer

---

The receiving user-agent (browser, JavaScript client) can now extract the parameters from the redirect\_uri fragment portion. The fragment value will only be available in the browser.

## response\_type=code

This is the safer response\_type to use because it is the most secure with regards to visibility of issued tokens. The flow involves multiple steps that are required between sending the initial request to receiving an access\_token

A client is requesting an access\_token using response\_type=code . This response\_type can be used if the client should not have access to user credentials. This response\_type includes a redirect that involves a browser or a web view on a mobile device.

---

### Request

---

**M** GET

**et**

**ho**

**d:**

**En** /auth/oauth/v2/authorize

**dp**

**oi**

**nt:**

**Pa** response\_type=code&client\_id=a-client\_id&redirect\_uri=a-redirect\_uri&scope=a-list-of-scope-  
**ra** values&state=a-state-value

**m**

**et**

**er**

**s:**

**O redirect\_uri:** If provided only requests using a registered redirect\_uri of this client will be granted by the OAuth server. If the parameter is not included the OAuth server will use the registered redirect\_uri. If multiple redirect\_uris have been registered the request will fail. If a redirect\_uri is included and none was registered the OAuth server will use the one included in the request

**O scope:** Only SCOPE values that have been registered for the client will be granted by the OAuth server

**on**

**al:**

**O state:** This value is opaque to the OAuth server and will be passed back unmodified to the client

**pti**

**on**

**al:**

---



---

### Response

---

**He** status: 200

**ad**

**er:**

**He** content-type: text/html

**ad**

**er:**

---

**Response**

**B** The user-agent will receive a login page. This page will request user credentials and the consent of **dy** the user. If the user denies the request the client will receive an error. If the user grants the client : it will receive an authorization code attached to the redirect\_uri

**N** The OAuth server will redirect the user-agent back to the client:

**ex**  
**t:**

**He** 302  
**ad**  
**er:**

**He** Location: the-redirect-uri?code=an-authorization-code&state=the-given-state  
**ad**  
**er:**

The receiving client can now extract the code (authorization\_code) from the redirect\_uri and exchange it for an access\_token (using grant\_type=authorization\_code).

## Requests Specifying grant\_type

The following scenarios describe requests for access\_token using a specified grant\_type:

- [grant\\_type=password \(see page 78\)](#)
- [grant\\_type=client\\_credentials \(see page 79\)](#)
- [grant\\_type=authorization code \(see page 80\)](#)
- [grant\\_type=urn:ietf:params:oauth:grant-type:jwt-bearer \(see page 80\)](#)
- [grant\\_type=urn:ietf:params:oauth:grant-type:saml2-bearer \(see page 81\)](#)
- [grant\\_type=refresh\\_token \(see page 82\)](#)

### grant\_type=password

This grant\_type can be used if the client was built by the enterprise that also implements the OAuth token server.

**Request**

**Meth** POST  
**od:**

**Heade** content-type: application/x-www-form-urlencoded  
**r:**

**Heade** authorization: Basic base64(client\_id:client\_secret) (This header can only be used if 'client\_id' r: and 'client\_secret' are **NOT** found within the message body and vice versa!)

**Endpo** /auth/oauth/v2/token  
**int:**

**Request**

**Param** grant\_type=password&username=a-username&password=a-users-password&client\_id=a-  
**eters:** client\_id&client\_secret=a-client\_secret&scope=a-list-of-scope-values

**Optio scope:** Only SCOPE values that have been registered for the client will be granted by the  
**nal:** OAuth server

**Response**

**Hea** status: 200  
**der:**

**Hea** content-type: application/json  
**der:**

**Body** Example: { "access\_token":"115b8c ... 11a5", "token\_type":"Bearer", "expires\_in":3600,  
**y:** "refresh\_token":"74b29d19-8b ... 7bb6bd1", "scope":"openid email" }

## grant\_type=client\_credentials

This grant\_type can be used if the client is acting on its own behalf. No user consent is required.

**Request**

**Meth** POST  
**od:**

**Heade** content-type: application/x-www-form-urlencoded  
**r:**

**Heade** authorization: Basic base64(client\_id:client\_secret) (This header can only be used if 'client\_id'  
**r:** and 'client\_secret' are **NOT** found within the message body and vice versa!)

**Endpo** /auth/oauth/v2/token  
**int:**

**Param** Parameters: grant\_type=client\_credentials&client\_id=a-client\_id&client\_secret=a-  
**eters:** client\_secret&scope=a-list-of-scope-values

**Optio scope:** Only SCOPE values that have been registered for the client will be granted by the  
**nal:** OAuth server

**Response**

**Heade** status: 200  
**r:**

**Heade** content-type: application/json  
**r:**

**Body:** { "access\_token":"115b8c ... 11a5", "token\_type":"Bearer", "expires\_in":3600, "scope":"  
 openid email" }

## grant\_type=authorization code

Exchange the authorization\_code for an access\_token. A client has received the authorization\_code attached to a redirect URI. The client now exchanges the authorization\_code for an access\_token by using grant\_type 'authorization\_code'.

### Request

**Meth** POST

**od:**

**Heade** content-type: application/x-www-form-urlencoded

**r:**

**Heade** authorization: Basic base64(client\_id:client\_secret) (This header can only be used if 'client\_id' r: and 'client\_secret' are NOT found within the message body and vice versa!)

**Endpo** /auth/oauth/v2/token

**int:**

**Param** grant\_type=authorization\_code&code=the-received-authorization-code&client\_id=a-  
eters: client\_id&client\_secret=a-client\_secret&redirect\_uri

**Optio** redirect\_uri: The value has to be included if it has been used in the initial request. It also has  
nal: to match the original value

### Response

**Hea** status: 200

**der:**

**Hea** content-type: application/json

**der:**

**Bod** { "access\_token":"115b8c ... 11a5", "token\_type":"Bearer", "expires\_in":3600, "refresh\_token":"  
y: 74b29d19-8b ... 7bb6bd1", "scope":"openid email" }

If the client included 'openid' as SCOPE in his request, additional keys are included in the response:

... "id\_token": "eyJ0eXAiO1v8 ... JZu\_LsN851VtfC5pclqJc", "id\_token\_type": "urn:ietf:params:oauth:  
grant-type:jwt-bearer" ...

The id\_token (JWT) can be used with grant\_type=urn:ietf:params:oauth:grant-type:jwt-bearer.

## grant\_type=urn:ietf:params:oauth:grant-type:jwt-bearer

This grant\_type can be used if the client is in possession of an id\_token (represented as JWT) of an authenticated user. Only id\_token (JWT) that were issued by the OAuth server are accepted.

### Request

**Meth** POST

**od:**

---

Request**Heade** content-type: application/x-www-form-urlencoded**r:****Heade** authorization: Basic base64(client\_id:client\_secret) (This header can only be used if 'client\_id' and 'client\_secret' are **NOT** found within the message body and vice versa!)**Endpo** /auth/oauth/v2/token**int:****Para** grant\_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Ajwt-bearer&assertion=a-meter jwt&client\_id=a-client\_id&client\_secret=a-client\_secret&scope=a-list-of-scope-values**s:****Optio** scope: Only SCOPE values that have been registered for the client will be granted by the **onal:** OAuth server

---

Response**Hea** status: 200**der:****Hea** content-type: application/json**der:****Bod** { "access\_token": "115b8c ... 11a5", "token\_type": "Bearer", "expires\_in": 3600, "refresh\_token": "y: 74b29d19-8b ... 7bb6bd1", "scope": "openid email" }

## grant\_type=urn:ietf:params:oauth:grant-type:saml2-bearer

This grant\_type can be used if the client is in possession of a SAML 2.0 token of an authenticated user. This scenario is useful in cases of federation where the SAML 2.0 token was signed by a trusted party.

---

Request**Meth** POST**od:****Head** content-type: application/x-www-form-urlencoded**er:****Head** authorization: Basic base64(client\_id:client\_secret) (This header can only be used if 'client\_id' and 'client\_secret' are **NOT** found within the message body and vice versa!)**Endp** /auth/oauth/v2/token**oint:****Para** grant\_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Asaml2-bearer&assertion=a-meter base64-encoded-saml-token&client\_id=a-client\_id&client\_secret=a-client\_secret&scope=a-list-rs: of-scope-values**Opti** scope: Only SCOPE values that have been registered for the client will be granted by the **onal:** OAuth server

---

Response**Heade** status: 200

r:

**Heade** content-type: application/json

r:

**Body:** { "access\_token":"115b8c ... 11a5", "token\_type":"Bearer", "expires\_in":3600, "scope":"openid email" }

## grant\_type=refresh\_token

This grant\_type can be used if the client is in possession of a refresh\_token. The request will only be successful if the refresh\_token has not expired. The parameter 'SCOPE' can only include the same or a subset of values that were originally requested. The refresh\_token can only be used once.

---

Request**Meth** POST**od:****Heade** content-type: application/x-www-form-urlencoded

r:

**Heade** authorization: Basic base64(client\_id:client\_secret) (This header can only be used if 'client\_id' r: and 'client\_secret' are NOT found within the message body and vice versa!)**Endpo** /auth/oauth/v2/token**int:****Param** Parameters: grant\_type=refresh\_token&refresh\_token=a-refresh-token&client\_id=a-  
**eters:** client\_id&client\_secret=a-client\_secret&scope=a-list-of-scope-values**Optio** scope: Only SCOPE values that have been registered for the client will be granted by the  
**nal:** OAuth server

---

Response**Hea** status: 200**der:****Hea** content-type: application/json**der:****Bod** { "access\_token":"115b8c ... 11a5", "token\_type":"Bearer", "expires\_in":3600, "refresh\_token":  
**y:** 74b29d19-8b ... 7bb6bd1", "scope":"openid email" }

# Customizing the OAuth ToolKit

All OAuth Toolkit features are implemented using policies in the Policy Manager, making it possible to customize the entire OAuth process.

- [Configure Token Lifetime Properties \(see page 83\)](#)
- [Customize Caches \(see page 84\)](#)
- [Customize the OAuth 2.0 Authorization Server Website \(see page 90\)](#)
- [Manage API Keys with CA API Portal \(see page 91\)](#)
- [Optimization \(see page 93\)](#)
- [OTK User Role Configuration \(see page 97\)](#)

## Configure Token Lifetime Properties

### Customize Token Lifetimes

The token lifetime is the time in seconds before the token expires. You can use the default values for the OAuth access\_token and refresh\_token, or set your own lifetime value.

Temporary tokens have short lifetimes for security reasons. We recommend you do not extend the default value for temporary tokens.

To customize token lifetimes:

1. In the Policy Manager, navigate to OAuth-version/PolicyFragments/configuration.
2. Open the OTK Token Lifetime Configuration policy.
3. Set the access\_token, refresh\_token, and id\_token. These lifetime values should be set so that the access\_token expires before the refresh\_token.

By default, client credentials have a lifetime value of 0, indicating no expiration date.

The following table describes the token lifetime variables.

OAuth 1.0 Variables	Description
oauth_v1_access_token_lifetime_s	Controls the lifetime of access tokens. Set to 0 to make the token invalid immediately. Default: 86400 seconds = 1 day.
oauth_v1_request_token_lifetime_s	Controls the lifetime of request tokens. Set to 0 to make the token invalid immediately. Default: 300 seconds

OAuth 1.0 Variables	Description
oauth_v1_consumer_key_lifetime_m	Controls the lifetime of OAuth consumer keys. Default: 0 minutes = Unrestricted lifetime
OAuth 2.0 Variables	Description
oauth2_auth_code_lifetime_sec	Controls the lifetime of issued OAuth codes. Set to 0 to make the code invalid immediately. Default: 600 seconds
oauth2_access_token_lifetime_sec	Controls the lifetime of issued access tokens. Set to 0 to make the token invalid immediately. Default: 3600 seconds
oauth2_refresh_token_lifetime_sec	Controls the lifetime of issued refresh tokens. Default: 604800 seconds
oauth2_client_id_lifetime_m	Default: 0 minutes
oauth2_client_id_lifetime_SDK_m	Default: 10080 minutes = 7 days

## Customize Caches

OTK policies take advantage of caching to avoid database calls and improve performance. The policies leverage local caches (visible on a single node only) and database backed caches (visible throughout all cluster nodes).

Default configuration within these caches is optimized for performance, however, you can customize the cache configuration to your specifications.

Click a policy or endpoint to view the related caches and customization notes.

- [OTK Client DB \(see page 84\)](#)
- [OTK Session DB \(see page 85\)](#)
- [OTK Session GET \(see page 85\)](#)
- [OTK Require OAuth 2.0 Token \(see page 86\)](#)
- [/auth/oauth/v1/authorize \(see page 86\)](#)
- [/auth/oauth/v2/authorize \(see page 87\)](#)
- [/oauth/manager \(see page 89\)](#)
- [/oauth/manager/clients \(see page 89\)](#)
- [/oauth/manager/tokens \(see page 89\)](#)
- [/oauth/validation/validate/v1/signature \(see page 90\)](#)

## OTK Client DB

---

allClientValuesCache

---

**Properties:** Stores client values when `access_token` requests are made.

---

**allClientValuesCache**

- lifetime of cache entries: **Customization Notes**  
300s The cache lifetime value determines how often the client configuration is looked up from the database. Reduce the cache lifetime for more frequent calls. Increase the lifetime for less frequent calls.
- content-type: text/xml
- max number of entries: 10,000 To modify the cache lifetime value, edit the cache assertions within any of the following policies:
  - OTK Client DB GET
  - OTK Client DB Revoke Key
  - OTK Client DB Update
- max size of entries in bytes: 100,000 bytes

## OTK Session DB

**defaultCache****Properties** Caches the OTK session information.**S:****Customization Notes**

Policy: OTK Session DB

The cache name can be configured when using any encapsulated assertion named "OTK Session".

## OTK Session GET

**openIDConnectCache**

- Properties:** Used to retrieve values associated with an authorization\_code in the context of OpenID Connect when the client exchanges the code for an access\_token.
- lifetime of cache entries: **Customization Notes**  
undefined
  - max number of entries: OTK grant\_type=AUTHORIZATION\_CODE  
undefined
  - max size of entries in bytes: Encapsulated Assertion: OTK Session GET  
undefined Customize the lifetime by modifying the assertions "OTK Session - Store" commented with "Cache Pollkey" on line 100. The value is expressed in seconds.

## OTK Require OAuth 2.0 Token

### accessTokenValidation

<b>Properties:</b>	Caches access_token validation results at OAuth 2.0 protected endpoints. This value is a compromise between performance and accepted lifetime of invalid tokens. The default cache time for saving client information is set to 30 seconds. Increasing this time elevates the risk of unauthorized access.
▪ lifetime of cache entries:	30s.
▪ content-type:	<b>Customization Notes</b>
text /xml	Policy: OTK Require OAuth 2.0 Token
▪ max number of entries:	The lifetime value is passed in through the interface of the encapsulated assertion wherever access_token_validation is used. Modify the cache lifetime to lookup access_tokens from the database more or less frequently.
100,000	
▪ max size of entries in bytes:	A lifetime value of 0 reduces performance requiring a database lookup for each single access_token validation. Increasing the lifetime value extends the expiration time of an access_token.
100,000 bytes	

### openIDConnectCache

<b>Properties:</b>	Used to retrieve values associated with an authorization_code in the context of OpenID Connect when the client exchanges the code for an access_token.
▪ lifetime of cache entries:	undefined
▪ max number of entries:	<b>Customization Notes</b>
10,000	OTK response_type=CODE
▪ max size of entries in bytes:	Encapsulated Assertion: OTK Session GET
10,000	Customize the lifetime by modifying the assertions "OTK Session - Store" commented with "Cache Pollkey" on line 100. The value is expressed in seconds.

## /auth/oauth/v1/authorize

### userSessionIDCacheV1

<b>Properties:</b>	Used with the session cookie "l7otk1a".
▪ lifetime of cache entries:	3,000s.
▪ max number of entries:	<b>Customization Notes</b>
1,000	Endpoint: /auth/oauth/v1/authorize

Encapsulated Assertion: OTK Session - Store

**userSessionIDCacheV1**

- max size of entries in bytes: 10,000 bytes      Customize the lifetime of the user session (cookie lifetime) by modifying the variable "ownerCacheAge". Value is expressed in seconds.

**userSessionIDCacheV1**

**Properties:** Handles values during the OAuth 1.0 message flow.

- lifetime of cache entries: 600s.      **Customization Notes**
- max number of entries: 10,000      Endpoint: /auth/oauth/v1/authorize
- max size of entries in bytes: 10,000 bytes      Encapsulated Assertion: OTK Session - Store  
Customize the lifetime of a temporary token by modifying the variable "sessionIdCacheAge". Value is expressed in seconds.

## /auth/oauth/v2/authorize

**OAuthAuthpageCache**

**Properties:** Caches the static content on the authorization server website (except for images).

- lifetime of cache entries: 60s.      **Customization Notes**
- max number of entries: 10      Endpoint: /auth/oauth/v2/authorize  
Consider modifying the cache lifetime if the web site template for the authorization server website changes often. Cache does not include form values or images. The lifetime property is modified within the cache-assertions in the block commented with "Load website template".
- max size of entries in bytes: 100,000 bytes

**userSessionIDCacheV2**

**Properties:** Used with the session cookie "l7otk2a".

- lifetime of cache entries: 3,000s.      **Customization Notes**
- max number of entries: 1,000      Endpoint: /auth/oauth/v2/authorize  
Encapsulated Assertion: OTK Session - Store

---

**userSessionIDCacheV2**

- max size of entries in bytes: Customize the lifetime of the user session (cookie lifetime) by modifying the variable "ownerCacheAge" at the beginning of the policy. The value is 10,000 bytes expressed in seconds.
- 

**userSessionIDCacheV2**

**Properties:** Used to handle values during the OAuth 2.0 message flow.

- lifetime of cache entries: **Customization Notes** 600s. Endpoint: /auth/oauth/v2/authorize
  - max number of entries: Encapsulated Assertion: OTK Session - Store 10,000 The temporary token lifetime determines the time allowed between a client request and the issuing of the user authorization code. Customize the lifetime value by modifying the variable "sessionIdCacheAge". The value is expressed in seconds.
  - max size of entries in bytes: 10,000 bytes
- 

**userSessionIDCacheV2**

**Properties:** Used to maintain the LinkedIn session during the authentication flow. LinkId is used with OAuth 1.0 which requires MAG to handle a temporary token/ secret

- lifetime of cache entries: **Customization Notes** 300s. Endpoint: /auth/oauth/v2/authorize
  - max number of entries: Encapsulated Assertion: OTK Session - Store 1,000 The temporary token lifetime determines the time allowed between a client request and the issuing of the user authorization code. Customize the lifetime value by modifying the variable "sessionIdCacheAge". The value is expressed in seconds.
  - max size of entries in bytes: 10,000 bytes
-

## /oauth/manager

I7managerCache	
Properties:	Customization Notes
▪ lifetime of cache entries: 600s	Used with the cookie "I7manager".
▪ max number of entries: undefined	Endpoint: /oauth/manager
▪ max size of entries in bytes: undefined	Encapsulated Assertion: OTK Session - Store
	Customize the lifetime value in the encapsulated assertion. The value is expressed in seconds.

## /oauth/manager/clients

I7managerCache	
Properties:	Customization Notes
▪ lifetime of cache entries: 600s	Endpoint: /oauth/manager/clients
▪ max number of entries: undefined	Encapsulated Assertion: OTK Session - Store
▪ max size of entries in bytes: undefined	Customize the lifetime value in the encapsulated assertion. The value is expressed in seconds.

## /oauth/manager/tokens

I7managerCache	
Properties:	Customization Notes
▪ lifetime of cache entries: 600s	Used with the cookie "I7manager".
▪ max number of entries: undefined	Endpoint: /oauth/manager/tokens
▪ max size of entries in bytes: undefined	Encapsulated Assertion: OTK Session - Store
	Customize the lifetime value in the encapsulated assertion. The value is expressed in seconds.

## /oauth/validation/validate/v1/signature

### consumerSecretCache

<b>Properties:</b>	Used with temporary values during the OAuth 1.0 flow to validate the signature.
▪ lifetime of cache entries:	30s
▪ max number of entries:	1,000
▪ max size of entries in bytes:	10,000
	<b>Customization Notes</b>
	Endpoint: /oauth/validation/validate/v1/signature
	Encapsulated Assertion: OTK Session - Store
	Customize the lifetime value in the encapsulated assertion. The value is expressed in seconds.

## Customize the OAuth 2.0 Authorization Server Website

The web content template for the authorization server is hosted in a policy. Customize this policy to match your corporate look and feel. Optionally host the website template on an external web server.

Navigate to OTK-version/Server/DMZ/OAuth 2.0. The service endpoint is /auth/oauth/v2/authorize.

## Customizing Website Template

To customize the website template:

1. Navigate to OTK-version/PolicyFragments/pof and open the OTK Website Template policy.
2. Modify any of the following:  
 website\_style (changes the css)  
 website\_top  
 website\_bottom



Do not modify the string: <!--oauth\_content\_placeholder--> found in websiteTemplate. This placeholder string is replaced with a HTML form-element that includes the request for username, password, grantbutton, denybutton.

## Hosting Website on an External Web Server

If you host the website template on an external web server, you must customize the OTK Variable Configuration policy in order to retrieve the website template.

In the Policy Manager, navigate to OTK-version/PolicyFragments/configuration and open the OTK Variable Configuration policy.

Variable	Notes
host_oauth2_auth_template_server	The URL of your webserver. Examples: <a href="http://myapache">http://myapache</a> <a href="https://localhost:8443">https://localhost:8443</a>
oauth2_auth_template_path	The path on your webserver. For example: /authorizationserver/index.html
location_website_template	The full path to the web app on your webserver. Default: \${host_oauth2_auth_template_server}\${oauth2_auth_template_path}
location_website_template_resources	The relative path to download resources such as images. For example: "\${host_oauth2_auth_template_server}/authorizationserver"

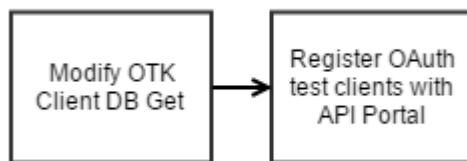
## Manage API Keys with CA API Portal



The following information applies to the on premise version of the CA API Developer Portal. No modification is required for the SaaS API Portal.

You can integrate CA API Portal with the OTK and manage OAuth clients by API key (OAuth client\_id). Integration requires modifying the OTK Client DB Get policy and registering OAuth clients with the API Portal.

After policy modification, OAuth clients no longer appear in OAuth Manager. Management is now only possible through CA API Portal. However, you can still use OAuth Manager to manage tokens.



- [Before You Begin \(see page 92\)](#)
- [Replace the OTK Client DB Get Policy \(see page 92\)](#)
- [Register OAuth Test Clients \(see page 92\)](#)

## Before You Begin

The following conditions must exist:

- /portalm/\* services are deployed on the target gateway
- /api/keys/\* services are deployed on the target gateway
- the assertion "Look Up API Key" is installed. This assertion is available when you install CA API Portal.

## Replace the OTK Client DB Get Policy

The replacement policy looks up client\_id's from the API Portal rather than the OTK database.

Replace the entire contents of the existing OTK Client DB Get Policy with the policy code provided below to create a new policy version. You can then toggle between the two versions using the **Revision History**.

The OTK Client DB Get policy is located in /OTK-version/Policy Fragments/persistence/client.

To replace the contents of the OTK Client DB Policy:

1. Open the OTK Client DB Get Policy
2. Delete the contents of the policy. Use Ctrl-A, then click Delete.
3. Click the XML icon to download the replacement code for the OTK Client DB Get Policy.  

4. Copy the policy code from the file and paste it into the empty OTK Client DB Get Policy.
5. Click **Save and Activate**.  
A new policy version is created.
6. In the services window, right-click the policy and add the revision history comment "OTK-currentversion-API-Portal-integration". Replace *currentversion* with the OTK version installed.

## Register OAuth Test Clients

This is optional. The test clients verify the integration of the OTK with the Portal.

Register the OTK test clients with the API portal by sending an HTTP PUT request to the target gateway.



The OAuth test clients are not managed applications of the API Portal. They do not appear listed as new applications.

## Create the Request

1. Create a HTTP PUT request using: `https://<yourGateway>:8443/portalman/1/api/keys`.
2. Set the content-type to "text/xml; charset=UTF-8"
3. Add an authorization header:  
`Authorization: Basic base64(username:password)`
4. Download the XML file below. Copy the contents and use it as the message body for the request.



5. Update the value of **CallbackUrl** in the XML request message as follows:
  - Replace **HOST** with the hostname of the target gateway.
  - If policies are installed with a URL prefix, update the path component accordingly.

```
<17:CallbackUrl>https://HOST:8443/mag/manager</17:CallbackUrl>
```

6. Use a tool such as SOAPUI or Fiddler to send the request.

The API keys are now registered with API Portal.

## Optimization

Optimization tasks are optional and include:

- [Remove Expired Tokens \(see page 94\)](#)
- [Configure Local Cache Lifetimes \(see page 94\)](#)
- [Configure Session Lifetime Properties \(see page 95\)](#)
- [Automate Database Maintenance \(see page 95\)](#)

## Remove Expired Tokens

By default, the policies do not search for expired tokens whenever a new token is persisted. You can search for expired tokens and remove them.

To delete expired tokens, perform one of the following tasks:

- Modify context variable settings in the OTK Storage Configuration encapsulated assertion.
- Run cron jobs to search for and remove expired tokens.

Be aware the searching for expired tokens negatively affects overall performance.

### Run Cron Jobs

Install cron jobs to search for and remove expired tokens. For cron job examples, see [Automate Database Maintenance \(see page 95\)](#).

No modification to the context variables in the OTK Storage Configuration assertion is required. Use the default `false` setting for the context variables.

### Modify Variables

If you are running cron jobs to automate the removal of the expired tokens, leave these variables at the default `false` setting.

However, if you choose not to run cron jobs, but want to remove expired tokens by default, set the value for each of the following context variables in OTK Storage Configuration to `true`:

Context Variables	Default Setting (supports cron jobs)	Modified Setting (no cron jobs)
<code>deleteExpiredSessions</code>	<code>false</code>	<code>true</code>
<code>deleteExpiredTokens</code>	<code>false</code>	<code>true</code>
<code>deleteExpiredClientKeys</code>	<code>false</code>	<code>true</code>
<code>delete_expired</code>	<code>false</code>	<code>true</code>

## Configure Local Cache Lifetimes

Caching improves the overall performance. There are two types of caches: local and cluster-wide. Local caches are implemented using the local cache assertion. These caches are visible only per gateway node. Clusterwide caches are implemented using a database and are therefore visible though a cluster.

For more information on caches used by MAG policies, see [Customize Caches \(<https://docops.ca.com/display/MAG/Customize+Caches>\)](#).

## Configure Session Lifetime Properties

To speed up the policy processing time, the OAuth Toolkit uses a combination of caching and session persistence to reduce the number of calls to the token store and client store endpoints. This functionality is provided by the oauth/session endpoint

The /oauth/validation/validate/v1/signature policy uses the session endpoint to cache the client\_key and access\_token values.

Token properties are stored in the session, but the expiry is determined by the expiration property within the token.

The cacheAge value controls how long the token and the client properties are valid within the session.

Default settings are as follows:

Context Variable	Default Value
cacheAge	30
cacheMaxEntries	1000
cacheMaxSize	10000

## Automate Database Maintenance

You can install cron jobs to perform database maintenance tasks. For example, create a cron-job to search for and remove expired tokens. If you install cron jobs, verify that the context variables in the OTK Storage Configuration are not responsible for performing the same tasks.

Cron job examples include:

- [Delete Expired Sessions and Tokens \(see page 95\)](#)
- [Delete Expired OAuth Clients \(see page 96\)](#)
- [Delete Expired Sessions \(see page 96\)](#)
- [Delete Temporary Sessions \(see page 96\)](#)
- [Delete Long Living Tokens \(see page 97\)](#)

### Delete Expired Sessions and Tokens

The following cron job:

- deletes expired sessions every 9 minutes
- deletes expired temporary tokens every 7 minutes
- deletes expired long living tokens every 15 minutes
- deletes expired client ids every 31 minutes

```
crontab -e
*/9 * * * * /home/ssgconfig/oauth_delete_sessions.sh
*/7 * * * * /home/ssgconfig/oauth_delete_temporary.sh
*/15 * * * * /home/ssgconfig/oauth_delete_longliving.sh
*/31 * * * * /home/ssgconfig/oauth_delete_client_ids.sh
```

## Delete Expired OAuth Clients

```
oauth_delete_client_ids.sh

# CA Technologies
# August, 2014
#
# Sample content for a shell script:
# Delete expired oauth clients
#
# OAuth client_id's may have a limited lifetime. Expired ones should be deleted
#
# For MySQL:
#
# Delete oauth client_id's
#
# Uncomment the last line
# Replace db_user, db_user_password, database with values valid in your environment
#
# mysql -u db_user -pdb_user_password database -e "DELETE FROM oauth_client_key WHERE
expiration > 0 AND expiration < unix_timestamp()"
```

## Delete Expired Sessions

```
oauth_delete_sessions.sh

# CA Technologies
# August, 2014
#
# Sample content for a shell script:
# Delete expired oauth sessions
#
# OAuth sessions are usually created if response_type=code is used
#
# For MySQL:
#
# Delete sessions
#
# Uncomment the last line
# Replace db_user, db_user_password, database with values valid in your environment
#
# mysql -u db_user -pdb_user_password database -e "DELETE FROM oauth_session WHERE
expiration < unix_timestamp()"
```

## Delete Temporary Sessions

```
oauth_delete_temporary.sh
```

```

# CA Technologies
# August, 2014
#
# Sample content for a shell script:
# Delete expired temporary oauth tokens
#
# For MySQL:
#
# Delete temporary tokens
#
# Uncomment the last line
# Replace db_user, db_user_password, database with values valid in your environment
#
# mysql -u db_user -pdb_user_password database -e "DELETE FROM oauth_initiate WHERE
# expiration < unix_timestamp()"

```

## Delete Long Living Tokens

`oauth_delete_longliving.sh`

```

# CA Technologies
# August, 2014
#
# Sample content for a shell script:
# Delete expired long living oauth tokens
#
# For MySQL:
#
# Two commands in order to not delete valid refresh_tokens
# 1. Search and delete expired refresh_tokens first. These rows will also include
expired access_token
# 2. Search and delete expired oauth_token/ access_tokens
#
# Uncomment the last two lines
# Replace db_user, db_user_password, database with values valid in your environment
#
# mysql -u db_user -pdb_user_password database -e "DELETE FROM oauth_token WHERE
rtoken IS NOT NULL AND rexpiration < unix_timestamp()"
# mysql -u db_user -pdb_user_password database -e "DELETE FROM oauth_token WHERE
expiration < unix_timestamp() and rtoken is null"

```

## OTK User Role Configuration

The OTK User Role Configuration encapsulated assertion found in OTK-*version*/Policy Fragments /configuration contains the following default user names associated with the administrator role:

- admin
- pmadmin
- administrator

The administrator role has a global view of clients, while the user role can see only their own clients.

To add a custom username to the administrator list:

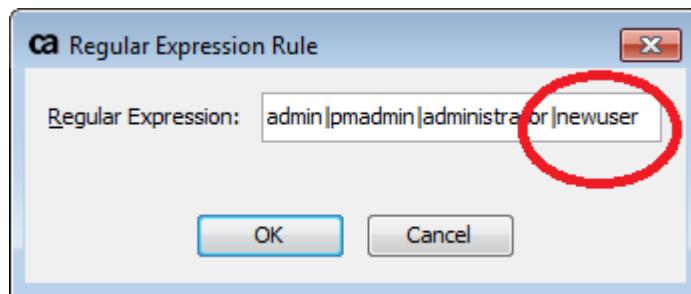
1. Open the **OTK User Role Configuration** encapsulated assertion.
2. Double-click the Compare Variable line of the Admin users section:

```

3 |     Comment: This policy determines if the current user is an admin
4 |     Comment: - The default implementation does not come with access to the actual user role.
5 |     Comment: - For the default implementation this policy will set the role 'admin' for a hardcoded list of user names.
6 |     Comment: - These usernames are 'admin', 'pmadmin', 'administrator'
7 |     ✓ default Set Context Variable current.user.role as String to: user
8 |     ROLE At least one assertion must evaluate to true Roles are configured here
9 |         Admin users All assertions must evaluate to true
10 |             Comment: Lookup the user from a list of known administrators
11 |             ✓ Compare Variable: ${current.username} matches admin|pmadmin|administrator; If Multivalued all values must pass
12 |             ✓ Set Context Variable current.user.role as String to: admin
13 |             ✓ Continue Processing

```

3. Click **Edit** and modify the Regular Expression by adding a pipe separator character | then typing the username to be assigned the administrator role.

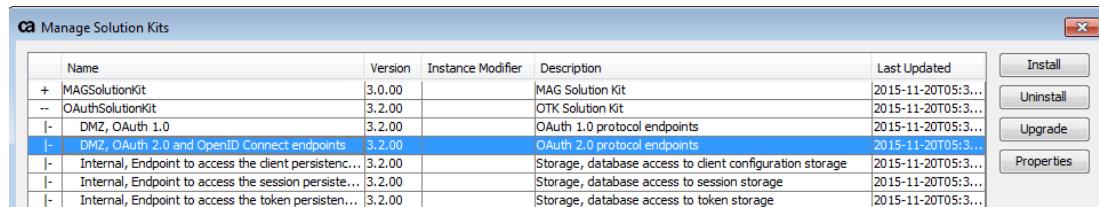


4. Click **OK** then save the policy.

The OTK User Role Configuration encapsulated assertion outputs the current.user.role and is used within OTK User Authentication policy.

# OpenID Connect Implementation

OpenID Connect is installed by default with the CA API Gateway. The OpenID Connect endpoints are installed through the OAuth Solution Kit.



The screenshot shows a software interface titled 'Manage Solution Kits'. A table lists several solution kits with columns for Name, Version, Instance Modifier, Description, and Last Updated. On the right side of the table are four buttons: 'Install', 'Uninstall', 'Upgrade', and 'Properties'. The row for 'DMZ, OAuth 2.0 and OpenID Connect endpoints' is highlighted with a blue selection bar, indicating it is currently selected. The 'Name' column shows entries like 'MAGSolutionKit', 'OAuthSolutionKit', 'DMZ, OAuth 1.0', and 'DMZ, OAuth 2.0 and OpenID Connect endpoints'. The 'Version' column shows versions such as '3.0.00', '3.2.00', and '3.2.00'. The 'Description' column provides a brief overview of each kit's purpose, and the 'Last Updated' column shows dates like '2015-11-20T05:3...'. The 'OAuth 2.0 and OpenID Connect endpoints' row has its entire row highlighted in blue.

Manage Solution Kits						
	Name	Version	Instance Modifier	Description	Last Updated	
+	MAGSolutionKit	3.0.00		MAG Solution Kit	2015-11-20T05:3...	<button>Install</button>
-	OAuthSolutionKit	3.2.00		OTK Solution Kit	2015-11-20T05:3...	<button>Uninstall</button>
-	DMZ, OAuth 1.0	3.2.00		OAuth 1.0 protocol endpoints	2015-11-20T05:3...	<button>Upgrade</button>
-	DMZ, OAuth 2.0 and OpenID Connect endpoints	3.2.00		OAuth 2.0 protocol endpoints	2015-11-20T05:3...	<button>Properties</button>
-	Internal, Endpoint to access the client persistenc...	3.2.00		Storage, database access to client configuration storage	2015-11-20T05:3...	
-	Internal, Endpoint to access the session persiste...	3.2.00		Storage, database access to session storage	2015-11-20T05:3...	
-	Internal, Endpoint to access the token persisten...	3.2.00		Storage, database access to token storage	2015-11-20T05:3...	

## Import Certificates

To complete the installation, import SSL certificates. This permits the Gateway to be used as a client.

1. Select **Tasks, Certificates, Keys and Secrets, Manage Certificates** (in browser client, from the **Manage** menu). The Manage Certificates dialog is displayed.
2. Select **Add**. The Add Certificate Wizard starts.
3. Select **Retrieve via SSL Connection** and then enter **<https://localhost:8443/>**.
4. Click **Next** and then click **Accept**. When warned about a hostname mismatch. Click **Next**.
5. Select the following check boxes:  
**Outbound SSL Connections**  
**Signing Certificates for Outbound SSL Connections**  
**Signing Client Certificates**
6. Click **Finish** to complete the wizard.
7. Restart the Gateway:  
`service ssg restart`

## Configure the Callback URL of the Test Client

To configure the callback URL through the OAuth Manager:

1. Open a browser and navigate to:  
`https://<hostname>:8443/<instanceMod>/oauth/manager`  
The *hostname* is the hostname of the gateway. For example: [gateway.com](http://gateway.com) (<http://gateway.com>)  
The optional *instanceMod* value distinguishes between multiple gateway instances on the same machine.

2. Provide a username and password. The type of access you are granted depends on your user role.
3. Click **Clients**.
4. For the **OpenID Connect Basic Client Profile** client, click **ListKeys**.  
The key details are displayed.
5. Click **Edit** and then replace the **Callback URL** field with the protocol, hostname, port and optional prefix of your gateway.  
Example: <https://ssg.example.com:8443/instanceMod> (<https://ssg.example.com:8443/prefix>)
6. Click **Save**.

## Run the Test Client

1. Open a browser and connect to one of the following URLs to open the OpenID Connect Test Client.

<code>&lt;Gateway&gt;/instanceMod:8443/oauth/v2/client/bcp</code>	Basic Client Profile
<code>&lt;Gateway&gt;/instanceMod:8443/oauth/v2/client/icp</code>	Implicit Client Profile

2. Click **send**. The browser is redirected to the authorization endpoint.
3. Provide the credentials of any user listed in the Internal Identity Provider of the Gateway and then click **Grant** or **Deny** to continue. Granting access allows the client to access not only protected resources but also personal information through the /userinfo OpenID Connect endpoint.  
The browser is redirected back to the client. The client receives the following tokens:
  - `access_token` (allows the client to access the user's personal information)
  - `refresh_token`
  - `id_token` (not used with this test client)
4. Click **Claims** to access the "/userinfo" endpoint.  
If the gateway is installed and working correctly, a JSON message containing several claims is returned.

# Implementation Details

## Supported Features

The following features are supported:

Feature	Values
Response_types	code, token id_token
OpenID Connect endpoints	userinfo
Signature methods	HS256, RS256, none
Tokens	id_token access_token refresh_token
Assertions to generate, decode, and validate id_tokens	Generate ID Token Decode ID Token

## Valid OpenID Connect Requests

A request becomes a valid OpenID Connect request if the client follows these conditions:

- The client uses response\_type=code or response\_type=token id\_token.
- The requested SCOPE includes openid.
- The client was registered with openid as a valid SCOPE value.
- The Gateway is configured for token\_type BEARER. OpenID Connect is not available with other token types.
- A client will be able to request any SCOPE but only registered SCOPES will be granted.

## Persistence

The implementation uses a database to persist all data. The client stores the id\_token with its policy using a "Store to Cache" assertion.

## Endpoints

The implementation includes these additional endpoints compared to a default OAuth Toolkit installation:

Endpoint	Notes
<code>&lt;Gateway&gt;/&lt;pr</code> <code>efix&gt;/oauth</code> <code>/validation</code> <code>/validate/v2</code> <code>/idtoken</code>	This endpoint is used when a client uses the "code" flow and when a user session must be validated.
<code>&lt;Gateway&gt;/&lt;pr</code> <code>efix&gt;/openid</code> <code>/connect/v1</code> <code>/client/*</code>	This endpoint implements the OpenID Connect test clients. The policy can be modified to change the behavior of the test clients. Client types are "Basic Client Profile" and "Implicit Client Profile".
<code>&lt;Gateway&gt;/&lt;pr</code> <code>efix&gt;/openid</code> <code>/connect/v1</code> <code>/userinfo</code>	This endpoint returns a JSON object containing claims defined by OpenID Connect. The content of the result depends on the granted SCOPE. The access_token used must be granted for the SCOPE openid.
<p><span style="color: #0070C0;">i</span> When using the Internet Explorer browser, a request to this endpoint may cause the browser to present a download menu. Internet Explorer does not support the application/json content type.</p>	
<p>By default, this endpoint returns the same result for any request except for the "sub" claim and the overall number of returned claims. The number of claims depends on granted/ requested SCOPE values. To change this default behavior, modify the policy to retrieve user specific values from an LDAP identity provider.</p>	

## Using the OpenID Connect Assertions

To add the assertions to a policy, drag and drop the assertion from the Message Validation /Transformation category in the **Assertion** tab.

- [Decode ID Token \(see page 102\)](#)
- [Generate ID Token \(see page 103\)](#)

## Decode ID Token

The OpenID Connect implementation contains additional assertions not found in the core system. In these new assertions, all text fields also support clusterwide properties and context variables. Any errors generated by these assertions will be described in detail in the Gateway Audit Events window.

Implement OpenID Connect by configuring the following assertions:

The decoding process of the **Decode ID Token** assertion extracts each value from the id token and makes it available as a context variable.

This assertion may be preceded by the **Decode JWT** assertion, which validates a JSON Web Token (JWT) and makes the JSON payload available as context variable.

## Context Variables Created by this Assertion

This assertion sets the following context variable for each check box selected under "Claims to export":

**didt.<checkboxName>**

The "didt" prefix is fixed and stands for **Decoded ID Token**.

If **Select all** is chosen, context variables are created for every claim.

## Configure Assertion Properties

When modifying the assertion, right-click **Decode ID Token** and select **Decode ID Token Properties** or double-click the assertion in the policy window. The assertion properties are displayed.

Configure the properties as shown in the following table:

Setting	Description
JSON Payload	Enter a JSON payload representing a valid id_token structure.
Validate	Select the validation to one of the following: <ul style="list-style-type: none"> <li>▪ Check for required claims When selected, the assertion fails in any of the following circumstances:               <ul style="list-style-type: none"> <li>▪ one of these claims is missing: iss, sub, aud, exp, iat</li> <li>▪ if unknown claims are included within the id_token</li> <li>▪ if claims occur more than once within the id_token</li> </ul> </li> <li>▪ Check expiration When selected, the assertion will fail if the id_token has expired. The value of "exp" is compared against the Gateway's local time</li> </ul>
Claims to export	Select the claims to be exported. A context variable will be created to hold the value of each exported claim.

## Generate ID Token

The Generate ID Token assertion is used to generate an id\_token. This assertion is divided into these sections:

- Required Claims
- Optional, but required with OAuth 2.0 implicit flow
- Optional Claims

## Context Variables Created by this Assertion

The Generate ID Token assertion sets the following context variable:

### **idtoken**

This variable contains the JSON representation of the ID\_TOKEN.

## Assertion Properties

To configured assertion properties, right-click **Generate ID Token** and select **Generate ID Token Properties** or double-click the assertion in the policy window. All fields accept context variables.



Note: The "iat" is generated automatically and is read only.

## Configuration Required

### Property Notes

iss	Enter the issuing party. This is also known as an "Issuer Identifier". This value should contain the "protocol, hostname, port" of the issuing server.
sub	Enter the user associated with this id_token.
Generate 'sub'	Select this check box to have the assertion generate a different value per requesting client. The value will be created as a PPID (Pairwise Pseudonymous Identifier).

For example, "sub" = "part1|part2|part3" generates:

*base64url(sh256(part1part2part3))*

Part3 is named as "salt" within the specification and should only be known by the issuer of the token.

For more details about the PPID, refer to the OpenID Connect specification located here: [http://openid.net/specs/openid-connect-messages-1\\_0.html#idtype.pairwise.alg](http://openid.net/specs/openid-connect-messages-1_0.html#idtype.pairwise.alg)

The assertion will fail if the content of this claim contains more than 255 characters at runtime.

aud	Enter the client_id
exp	Enter the expiration date in seconds since 01.01.1970-00:00:00.

For example, "exp" = "2016-04-21T12:00:00Z" generates:

### Property Description

automat ic	Select this check box to let the assertion determine what should be included in the token, based on the <b>response_type</b> field.
------------	---

- If response type = **code**, then **c\_hash** is included
- If response type = **tokenid\_token**, then **at\_hash** and **nonce** are included

## Configuration Required for OAuth 2.0 Implicit Flow Only

### Property Description

automat ic	Select this check box to let the assertion determine what should be included in the token, based on the <b>response_type</b> field.
------------	---

- If response type = **code**, then **c\_hash** is included
- If response type = **tokenid\_token**, then **at\_hash** and **nonce** are included

### Property Description

Be sure to complete the fields for **at\_hash**, **c\_hash**, and **nonce** if this check box is selected. Clear this check box to always include any values entered in the **at\_hash**, **c\_hash**, and **nonce** fields

For more information about at\_hash, refer to these specifications: \* [http://openid.net/specs/openid-connect-messages-1\\_0.html#id\\_token](http://openid.net/specs/openid-connect-messages-1_0.html#id_token)\*

**response\_type** Enter a response type if **automatic** is selected. Valid response types are: **code** and **token**

**at\_hash** Specify an access\_token to be used to generate an at\_hash.

**c\_hash** Specify a code to be used to generate a c\_hash value.

**nonce** Specify a nonce. This is required if the implicit flow is used (response\_type=token id\_token). The value *must* be the one provided by the client. The assertion will fail if this value is not provided with the implicit flow.

## Optional Configuration

### Property Notes

**azp** If this id\_token is usable by other clients, enter the ID of the client that will use this token.

**acr** Enter an Authentication Context Class Reference. Accepted values are between **0** and **4** (inclusive) or other values such as URLs.

**auth\_time** Enter the time in seconds since 01.01.1970-00:00:00 when the access\_token was granted.

**now** To use the current system time, select the **[now]** check box.

# APIs and Assertions

---

The OAuth Toolkit uses the APIs provided by the different components. The APIs can be used by any other third-party client.

The following APIs are provided and are required.

- [APIs \(see page 106\)](#)
- [OAuth Client Assertions \(see page 364\)](#)
- [Encapsulated Assertions \(see page 366\)](#)
- [Error Codes \(see page 372\)](#)

All APIs support HTTP GET and HTTP POST (content-type: *application/x-www-form-urlencoded*). The requirement for SSL can be configured by customizing the policy.

Values within brackets followed by a "?" are optional parameters; for example: (&parameter=value)?

## APIs

- [OAuth API Endpoints \(see page 106\)](#)
- [OAuth Validation Point \(OVP\) API \(see page 185\)](#)
- [OAuth Protected APIs \(see page 211\)](#)
- [Clientstore API \(see page 215\)](#)
- [Tokenstore API \(see page 285\)](#)
- [Sessionstore API \(see page 348\)](#)
- [Portal Storage API \(see page 357\)](#)

## OAuth API Endpoints

OAuth APIs include:

- [request\\_authorization\\_init \(see page 107\)](#)
- [request\\_authorization\\_login \(see page 118\)](#)
- [request\\_authorization\\_consent \(see page 126\)](#)
- [request\\_token\\_password\\_flow \(see page 133\)](#)
- [request\\_token\\_code\\_flow \(see page 139\)](#)
- [request\\_token\\_refresh\\_flow \(see page 145\)](#)
- [request\\_token\\_client\\_creds\\_flow \(see page 152\)](#)
- [request\\_token\\_jwt\\_flow \(see page 157\)](#)
- [request\\_token\\_saml\\_flow \(see page 164\)](#)
- [revoke\\_token \(see page 170\)](#)
- [client\\_details\\_export \(see page 174\)](#)

- [resource\\_owner\\_logout \(see page 178\)](#)
- [resource\\_owner\\_session\\_status \(see page 182\)](#)

ID	Operation	URL-Path	HTTP Method	HTTP Header	Body/Query Params (for GET attach params to URL-PATH)	Comment
th or iz at io n _i ni t	<b>Initializes an OAuth 2.0 flow using a response_type. See RFC 6749 (</b> <a href="https://tools.ietf.org/html/rfc6749">https://tools.ietf.org/html/rfc6749</a> ) <b>for more details</b>	/aut /oau /v2 /aut /s.ietf.org /html /rfc6749	GET	cookie: client_id=<client_id>& mag-identifier: scope=<scope>& content-type: response_type=<response_type>& type: nonce=<nonce>& application/x-www-form-urlencoded	state=<state>	Initiates the OAuth 2.0 flow using a response_type.  <b>cookie</b> (optional) : Used to verify the active session. If it is valid, it represents an authenticated user and the OAuth server will not request credentials. <b>mag-identifier</b> (optional) : [MAG]: used with response_type 'code' <b>mag-identifier</b> (optional) : [MAG]: mobile clients MUST use this header. It must represent a valid registered mobile device. <b>client_id</b> :

ID	Operation	URL-	HTTP Path	HTTP Method	Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
							'client_id' of the requesting client. <b>redirect_uri</b> (optional) : A 'redirect_uri' that was registered for this client. It is required if multiple redirect_uri's have been registered for this client. <b>scope</b> (optional) : Only SCOPE values that were registered for this client will be granted. If only non-matching SCOPE values are requested, the request will fail. <b>response_type</b> : Must be 'code' or 'token' or 'token id_token' <b>response_type</b> : 'token id_token' is only supported

ID	Operation	URL-	HTTP Path	HTTP Method	Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
							<p>in conjunction with scope=openid</p> <p><b>nonce</b> (optional) : This is required for response_type 'token' and 'token id_token' within the context of OpenID Connect.</p> <p><b>display</b> (optional) : This is optional and used within the context of OpenID Connect. Currently only 'page' is supported. The OAuth server returns an authorization page formatted for desktop browsers.</p> <p><b>display</b> (optional) : [MAG]: the value 'social_login' is specified. The response will be a JSON</p>

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
						message instead of an HTML page containing a list of social login providers. <b>state</b> (optional) : Value opaque to the server, used by the client to track its session. It will be returned as received.
Response					status: <html><body>...</body></html> 200 content -type: text /html; charset =UTF-8	An HTML page displaying a login screen.
Response					status: { "idp": "all", "providers": [ { "provider": { "id": "enterprise", "auth_url": "a-url" } } ] } 200 content -type: application/json; charset =UTF-8	[MAG] A JSON message if the parameter 'display' was set to 'social_login'. The key 'provider' may appear multiple times. 'idp' will contain valid providers. The value 'all' indicates

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
						that any of the list can be used to log in.
Response					location error=invalid_request&n: Pragma error_description=The given mag-identifier is either : no- invalid or points to an unknown device&cache state=<state>&Cache- x-ca-err=3000107Control : no-store status: 302	[MAG] invalid mag- identifier. If the redirect_uri is not valid, the server displays an HTML page showing the error and error_description. The HTTP status will be '400'
						<b>location :</b> Location (URL) has the 'error', 'error_description' and 'state' (if provided) parameter attached. <b>state</b> (optional) : Value opaque to the server, used by the client to track its session. It will be returned as received.
Response					location error=invalid_request&n: Pragma	[MAG] invalid mag-

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					: no-cache Cache-control: error_description=The referenced device is not activated&Control state=<state>& : no-store status: 302	identifier. If the redirect_uri is not valid, the server displays an HTML page showing the error and error_description. The HTTP status will be '400'
					<b>location :</b> Location (URL) has the 'error', 'error_description' and 'state' (if provided) parameter attached. <b>state</b> (optional) : Value opaque to the server, used by the client to track its session. It will be returned as received.	
Response					location: error=invalid_scope&n: Pragma: error_description=No registered scope value for this client has been requested&cache-control: state=<state>&Cache-control: x-ca-err=3000115 Control	invalid scope If the redirect_uri is not valid, the server displays an HTML page showing the error

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
		: no-store				and error_desciption. The HTTP status will be '400'
		status: 302				<b>location :</b> Location (URL) has the 'error', 'error_desciption' and 'state' (if provided) parameter attached. <b>state</b> (optional) : Value opaque to the server, used by the client to track its session. It will be returned as received.
Response		location			error=unsupported_response_type&n: Pragma error_description=None of the supported response_types were used&cache state=<state>&Control	unsupported response type If the redirect_uri is not valid, the server displays an HTML page showing the error and error_desciption. The HTTP status will be '400'
		: no-store			status: 302	

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					<b>location :</b> Location (URL) has the 'error', 'error_desc'ription' and 'state' (if provided) parameter attached. <b>state</b> (optional) : Value opaque to the server, used by the client to track its session. It will be returned as received.	
Response					location error=unauthorized_client&n: Pragma error_description=The client lacks authorization for : no- this request& cache state=<state>& Cache- x-ca-err=3000117 Control : no- store status: 302	unauthorized client If the redirect_uri is not valid, the server displays an HTML page showing the error and error_desc ription. The HTTP status will be '400'

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
						provided) parameter attached. <b>st</b> <b>ate</b> (optional) : Value opaque to the server, used by the client to track its session. It will be returned as received.
Error-Response		x-ca-err: 300010			<html><head><title>Authorization Server Error</title></head><body><div id="authSrvErr">300010 <p><b>error:</b>invalid_request 3 /><b>error_description:</b>Missing or duplicate Pragma parameters</p></div></body></html>	invalid parameters
Error-Response		x-ca-err: 300011			: no- cache Cache- Control : no- store status: 400 content -type: text /html; charset =UTF-8	Invalid redirect_uri .

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error-Response				<pre>text /html; charset =UTF-8</pre>	
Error-Response				<pre>x-ca- &lt;html&gt;&lt;head&gt;&lt;title&gt;Authorization Server Error&lt;br&gt; err: /title&gt;&lt;/head&gt; &lt;body&gt; &lt;div id="authSrvErr"&gt; 300013 &lt;p&gt;&lt;b&gt;error: &lt;/b&gt;invalid_request&lt;br&gt; 0     /&gt;&lt;b&gt;error_description: &lt;/b&gt;The client type is not Pragma valid&lt;/p&gt; &lt;/div&gt; &lt;/body&gt; &lt;/html&gt; : no- cache Cache- Control : no- store status: 400 content -type:  applica tion /json; charset =UTF-8</pre>	invalid client type
Error-Response				<pre>x-ca- &lt;html&gt;&lt;head&gt;&lt;title&gt;Authorization Server Error&lt;br&gt; err: /title&gt;&lt;/head&gt; &lt;body&gt; &lt;div id="authSrvErr"&gt; 300020 &lt;p&gt;&lt;b&gt;error: &lt;/b&gt;invalid_request&lt;br&gt; 1     /&gt;&lt;b&gt;error_description: &lt;/b&gt;The given client Pragma credentials were not valid&lt;/p&gt; &lt;/div&gt; &lt;/body&gt; &lt;br&gt; : no- /html&gt; cache Cache- Control : no- store status: 401 content -type: text /html; charset =UTF-8</pre>	The client could not be authenticated.
Error-Response				<pre>x-ca- &lt;html&gt;&lt;head&gt;&lt;title&gt;Authorization Server Error&lt;br&gt; err: /title&gt;&lt;/head&gt; &lt;body&gt; &lt;div id="authSrvErr"&gt; 300020 &lt;p&gt;&lt;b&gt;error: &lt;/b&gt;invalid_request&lt;br&gt; 3     /&gt;&lt;b&gt;error_description: &lt;/b&gt;SSL is required&lt;/p&gt; &lt;br&gt; Pragma /div&gt; &lt;/body&gt; &lt;/html&gt; : no-</pre>	Forbidden.

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				cache Cache- Control : no- store status: 403 content -type: text /html; charset =UTF-8	
Error- Response				Allow: <html><head><title>Authorization Server Error  GET, /title></head> <body> <div id="authSrvErr">  POST <p><b>error: </b>invalid_method  Pragma /><b>error_description: </b>{HTTP_METHOD}  : no- method is not valid</p> </div> </body> </html> }  cache Cache- Control : no- store status: 405 content -type: text /html; charset =UTF-8	The HTTP method is not valid <b>Allow :</b> This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> (https://tools.ietf.org/html/rfc2616) and contains a comma separated list of valid HTTP methods.
Error- Response		x-ca- err:		<html><head><title>Authorization Server Error  </title></head> <body> <div id="authSrvErr">  300000 <p><b>error: </b>server_error  0 /><b>error_description: </b>The request failed due Pragma to an unknown reason</p> </div> </body> </html> }  cache Cache- Control : no- store status: 500 content -type:	Unknown error.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					text /html; charset =UTF-8	
th or iz at io n _l o gi n	Authenticates a user during the OAuth 2.0 response_ e type flow.	/aut /oau /v2 /aut horiz /logi n	POST	cookie: action=<action>& content-type: sessionID=<sessionID>& application/x-www-form-urlencoded	username=<username>& password=<password>& persistent_cookie=no& provider=<provider>& state=<state>& oauth_token=<oauth_token>& oauth_verifier=<oauth_verifier>	Authenticates a resource owner in the OAuth 2.0 code-flow. The user can be authenticated via a cookie or username /password. Use these parameters for a default implementation: 'sessionID, username, password, action=log in'. [MAG]: users can authenticate using social login. [MAG]: Social login parameter combinations are: 'code, provider, state' or 'oauth_token, oauth_verifier (may be empty, depending on the step

ID	Operation	URL-	HTTP Path	HTTP Method	Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
							of the OAuth 1.0 protocol), provider, state'.

**cookie**  
(optional) : Used to verify the active session. If it is valid, it represents an authenticated user and the OAuth server will not request credentials.

**action** : Either 'login' or 'cancel' or 'reset'. 'reset' will be used if the current user is not the one whose username is displayed.

**sessionID** : The current session which was initiated at the initial authorization request. Optional if social login is used.

**username** :

ID	Operation	URL-	HTTP Path	HTTP Method	Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
							<p>The username of the user to be authenticated. Has to be provided with 'password'. Optional if a cookie is provided or social login is used.</p> <p><b>password :</b> The password of the user to be authenticated. Has to be provided with 'username'. Optional if a cookie is provided or social login is used.</p> <p><b>persistent_cookie</b> (optional) : Either 'yes' or 'no' to remember the user. This only takes effect when username /password were provided.</p> <p><b>code</b> (optional) : [MAG]: An</p>

ID	Operation	URL-	HTTP Path	HTTP Method	Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
							OAuth 2.0 authorization_code issued by a social login provider. <b>provider</b> (optional) : [MAG]: The social login provider that issued the 'code'. This value is specified as part of the redirect_uri configured with each social login provider. <b>state</b> (optional) : [MAG]: The state representing the session (state is used instead of sessionID). This was attached to the initial request to the social login provider. <b>oauth_token</b> (optional) : [MAG]: OAuth 1.0 token when the social login provider is

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
						used with OAuth 1.0. <b>oauth_verifier</b> (optional) : [MAG]: OAuth 1.0 verifier when the social login provider is used with OAuth 1.0.
Response					status: <html><body>...</body></html> 200 content -type: text /html; charset =UTF-8	An HTML page displaying the consent screen.
Response					location error=authentication_error&n: Pragma error_description=The resource_owner denied to : no- authenticate& cache state=<state>& Cache- x-ca-err=3001123 Control : no- store status: 302	Authentication denied If the redirect_uri is not valid, the server displays an HTML page showing the error and error_description. The HTTP status will be '400'  <b>location :</b> Location (URL) has the 'error', 'error_description' and 'state' (if provided) parameter attached. <b>state</b>

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
						(optional) : Value opaque to the server, used by the client to track its session. It will be returned as received.
Response					location error=invalid_request&n: Pragma error_description=Missing or duplicate parameters& If the : no- state=<state>& cache x-ca-err=3001103 Cache- Control : no- store status: 302	invalid parameters  redirect_uri is not valid, the server displays an HTML page showing the error and error_description. The HTTP status will be '400'

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error-Response	x-ca-err: 300111 0			<html><head><title>Authorization Server Error</title></head><body><div id="authSrvErr">300111 <p><b>error:</b>invalid_request 0 </p><b>error_description:</b>The session has Pragma expired or already been granted. The login process : no- has to be repeated to be successful</p></div><cache /body></html> }	returned as received.
Error-Response	x-ca-err: 300120 2			<html><head><title>Authorization Server Error</title></head><body><div id="authSrvErr">300120 <p><b>error:</b>authentication_error 2 </p><b>error_description:</b>The resource owner Pragma could not be authenticated due to missing or invalid : no- credentials</p></div></body></html> }	The session has expired or already been granted. An HTML page displaying an authentication error will be returned.
Error-Response	x-ca-err: 300120 3			<html><head><title>Authorization Server Error</title></head><body><div id="authSrvErr">300120 <p><b>error:</b>invalid_request 3 </p><b>error_description:</b>SSL is required</p><Pragma /div></body></html>	Forbidden.

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				Control : no-store status: 403 content-type: application/json; charset=UTF-8	
Error-Response				Allow: <html> <head><title>Authorization Server Error            POST /title></head> <body> <div id="authSrvErr">           Pragma <p><b>error: </b>invalid_method            : no- ><b>error_description: </b>{HTTP_METHOD}            cache method is not valid</p> </div> </body> </html> } Cache-Control : no-store status: 405 content-type: text/html; charset=UTF-8	The HTTP method is not valid. <b>Allow :</b> This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> (https://tools.ietf.org/html/rfc2616) and contains a comma separated list of valid HTTP methods.
Error-Response	x-ca-err:			<html> <head><title>Authorization Server Error            /title></head> <body> <div id="authSrvErr">           300100 <p><b>error: </b>server_error            0 ><b>error_description: </b>The request failed due to an unknown reason</p> </div> </body> </html> } : no-cache Cache-Control : no-store status: 500 content-type: text/html; charset=UTF-8	Unknown error.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
th or iz at io n _c o ns e nt nt	Handles the result of a resource owners decision at the consent page.	/aut /oau /v2 /aut /horiz /con sent	POST	content-type: application/x-www-form-urlencoded	sessionID=<sessionID>&action=<action>	<p>The resource owner has given consent (Grant) or denied (Deny) permission for a client to access resources.</p> <p>The methods 'request_authorization_init' and 'request_authorization_login' must have been processed beforehand.</p> <p>.</p> <p><b>sessionID :</b> Contains the sessionID issued when the HTML login /consent page was displayed.</p> <p><b>action :</b> Either 'Grant' or 'Deny' or 'reset' depending on the selection of the user. 'reset' will return an HTML page</p>

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
	Response				Location code=<code>& state=<state> status: 302	<p>so that the user can login again.</p> <p>Response for the 'authorization_code' flow. Redirect back to the client after a successful authentication.</p> <p><b>Location :</b> The 'redirect_uri' to the client that will receive the result of this request including parameters .</p> <p><b>code :</b> The issued authorization_code. It can be exchanged for an access_token.</p> <p><b>state (optional) :</b> Value opaque to the server, used by the client to track its session. It will be returned as received.</p>

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Response					<p>Location access_token=&lt;access_token&gt;&amp;nbsps;</p> <p>status: expires_in=&lt;expires_in&gt;&amp;nbsps;</p> <p>302 token_type=Bearer&amp;nbsps;</p> <p>scope=&lt;scope&gt;&amp;nbsps;</p> <p>id_token=&lt;id_token&gt;&amp;nbsps;</p> <p>id_token_type=&lt;id_token_type&gt;&amp;nbsps;</p> <p>state=&lt;state&gt;</p>	<p>Response for the 'implicit' flow.</p> <p>Redirect back to the client after a successful authentication.</p> <p><b>Location :</b> The 'redirect_uri' to the client that will receive the result of this request including parameters within a URL fragment.</p> <p><b>access_token :</b> The issued access_token. This parameter is part of the URL fragment.</p> <p><b>expires_in :</b> The access_token lifetime in seconds. This parameter is part of the URL fragment.</p> <p><b>token_type :</b> The token type. This parameter is part of the URL</p>

ID	Operation	URL-	HTTP Path	HTTP Method	Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
							fragment. <b>scope</b> : The granted SCOPE which may differ from the requested ones. This parameter is part of the URL fragment. <b>id_token</b> (optional) : The id_token (represented as JWT) which will be issued for response_type 'token id_token' if the requested SCOPE included 'openid'. This parameter is part of the URL fragment. <b>id_token_type</b> (optional) : The type of id_token. This is a OTK extension to allow the creation of other types, e.g.: SAML. This parameter is part of

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
						the URL fragment. <b>state</b> (optional) : Value opaque to the server, used by the client to track its session. It will be returned as received.
Response					location error=access_denied&n: Pragma error_description=The resource_owner denied : no- access to resources&cache state=<state>&Cache- x-ca-err=3002124 Control : no- store status: 302	Access was denied  <b>location</b> : Location (URL) has the 'error', 'error_desc'ription' and 'state' (if provided) parameter attached. <b>state</b> (optional) : Value opaque to the server, used by the client to track its session. It will be returned as received.
Response					location error=invalid_request&n: Pragma error_description=Missing or duplicate parameters <b>&amp;</b> If the : no- state=<state>&cache x-ca-err=3002103 Cache- Control : no-	invalid parameters  <b>redirect_uri</b> is not valid, the server displays an HTML page

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
			store			showing the error and error_desc option. The HTTP status will be '400'
			status:			
			302			
						<b>location :</b> Location (URL) has the 'error', 'error_desc' and 'state' (if provided) parameter attached. <b>state</b> (optional) : Value opaque to the server, used by the client to track its session. It will be returned as received.
Error-Response		x-ca-err:	<html><head><title>Authorization Server Error</title></head><body><div id="authSrvErr">300211 <p><b>error:</b>invalid_request 0 <b>error_description:</b>The session has Pragma expired or already been granted</p></div><: no-</body></html>}	cache		The resource owner has been authenticated but the session has expired.
		Cache-Control				
		: no-				
		store				
		status:				
		400				
		content-type:				

ID	Operation URL- HTTP Path	HTTP Method	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
			text /html; charset =UTF-8	
Error- Response	x-ca- err: 300220 3 Pragma : no- cache Cache- Control : no- store status: 403 content -type: applica tion /json; charset =UTF-8	<html><head><title>Authorization Server Error</title></head><body><div id="authSrvErr"><p><b>error:</b>invalid_request 3</p><b>error_description:</b>SSL is required</p> <div><meta http-equiv="refresh" content="0; url=https://www.google.com"></div></body></html>		Forbidden.
Error- Response	Allow: POST Pragma : no- cache Cache- Control : no- store status: 405 content -type: text /html; charset =UTF-8	<html><head><title>Authorization Server Error</title></head><body><div id="authSrvErr"><p><b>error:</b>invalid_method : no- </p><b>error_description:</b>{HTTP_METHOD} method is not valid</p></div></body></html> } This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> and contains a comma separated list of valid HTTP methods.	The HTTP method is not valid	<b>Allow :</b> <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> )
Error- Response	x-ca- err: 300200 0 Pragma : no- }	<html><head><title>Authorization Server Error</title></head><body><div id="authSrvErr"><p><b>error:</b>server_error 0</p><b>error_description:</b>The request failed due to an unknown reason</p></div></body></html>		Unknown error.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					cache Cache- Control : no- store status: 500 content -type: text /html; charset =UTF-8	
e	Request	/aut	POST	mag- access_to h	client_id=<client_id>& identifi	mag- identifier
n	ken,	/oau		er:	client_secret=<client_secret>&	(optional) :
-	refresh_t	th		authori	grant_type=password&	[MAG]
p	oken	/v2		zation:	scope=<scope>&	
as	using	/tok		content	username=<username>&	
as	OAuth	en		username	password=<password>	
s	2.0			applica		
w	grant_typ			tion/x-		
w	e=passwo			www-		
or	rd. See RF			form-		
d	C 6749 (htt			urenco		
-f	ps://tools.			ded		
lo	ietf.org					
lo	/html					
w	/rfc6749)					
	for more					
	details					

ID	Operation	URL-	HTTP Path	HTTP Method	Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
							can only be used INSTEAD of the parameters 'client_id', 'client_secret'. <b>client_id</b> : The client_id. This can only be used INSTEAD of using the authorization header. <b>client_secret</b> : The client_secret. This can only be used INSTEAD of using the authorization header. <b>grant_type</b> : MUST be set to 'password'scope (optional) : Only SCOPE values that were registered for this client will be granted. If only non-matching SCOPE values are requested, the request will fail.

---

Response	<pre>Pragma { "access_token":"an_access_token", "token_type":"The : no-   Bearer", "expires_in":3600, "refresh_token":" cache  a_refresh_token", "scope":"granted_scope" } Cache- Control : no- store status: 200 content -type:  applica tion /json; charset =UTF-8</pre>	The access_token can be used to access protected resources. The refresh_token can be exchanged for an access_token and a new refresh_token.
Response	<pre>Pragma { "access_token":"an_access_token", "token_type":"[MAG]:" : no-   Bearer", "expires_in":3600, "refresh_token":" cache  a_refresh_token", "scope":"granted_scope", Cache- "id_token":"an_id_token", "id_token_type":"" Control an_id_token_type" } : no- store status: 200 content -type:  applica tion /json; charset =UTF-8</pre>	This response includes an id_token and id_token_type if the requested SCOPE included 'mssso'.
Error-Response	<pre>x-ca- { "error":"invalid_request", "error_description":"" err: The given mag-identifier is either invalid or points 300310 to an unknown device" } 7 Pragma : no- cache Cache- Control : no- store status: 400 content -type:  applica tion</pre>	[MAG] invalid mag-identifier.

---

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error- Response					/json; charset =UTF-8	
Error- Response				x-ca- err: 300310 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	{ "error":"invalid_request", "error_description":"Missing or duplicate parameters" }	If any required parameters or headers are missing, the request will fail.
Error- Response				x-ca- err: 300311 5 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	{ "error":"invalid_scope", "error_description":"No registered scope value for this client has been requested" }	invalid scope
Error- Response				x-ca- err: 300311 9 Pragma : no-	{ "error":"unsupported_grant_type", "error_description":"The given grant_type is not supported" }	unsupported grant_type

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	
Error- Response		x-ca- err: 300320 1 WWW- Authen ticate: Basic error=" Invalid or missing credent ials" Pragma : no- cache Cache- Control : no- store status: 401 content -type: applica tion /json; charset =UTF-8		{ "error":"invalid_client", "error_description":"The given client credentials were not valid" }	The client could not be authenticated. <b>WWW-Authentica te</b> (optional) : This header is required by <a href="https://tools.ietf.org/html/rfc7235">RFC 7235</a> ( <a href="https://tools.ietf.org/html/rfc7235">https://tools.ietf.org/html/rfc7235</a> ) if an authentication scheme has been used. It contains the used authentication scheme and an error message.	
Error- Response		x-ca- err: 300320 2 Pragma : no-		{ "error":"invalid_request", "error_description":"The resource owner could not be authenticated due to missing or invalid credentials" }	The resource owner	

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				cache Cache- Control : no- store status: 401 content -type: applica tion /json; charset =UTF-8		could not be authenticat ed
Error- Response		x-ca- err:	300320 3	{ "error":"invalid_request", "error_description":"SSL Forbidden. is required" }		
Error- Response				Allow: POST Pragma : no- cache Cache- Control : no- store status: 405		The HTTP method is not valid <b>Allow :</b> This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rf c2616</a> ) and contains a comma separated

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
						list of valid HTTP methods.
	Error- Response			x-ca-err: 300300 0 Pragma : no- cache Cache- Control : no- store status: 500		Unknown error.
e	Request access_to_h_ken,_refresh_t_oken,_using_d_OAuth_e_2.0,_grant_typ_e=authori_lo_zation_co_w_de. See RFC 6749 (https://tools.ietf.org/html/rfc6749) for more details.	/aut	POST	mag- identifi er:	client_id=<client_id>& client_secret=<client_secret>& authori grant_type=authorization_code& zation: code=<code>& content redirect_uri=<redirect_uri> -type: aplica tion/x- www- form- urlenco ded	mag- identifier (optional) : [MAG]
						mag- identifier (optional) : [MAG]: mobile clients MUST use this header. It must represent a valid registered device. <b>authorizati</b> <b>on</b> : The HTTP basic authorizati on header containing the client credentials as base64 encoded string (authorizati on: Basic base64 (client_id:

ID	Operation	URL-	HTTP Path	HTTP Method	Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
							client_secret)). This can only be used INSTEAD of the parameters 'client_id', 'client_secret'. <b>client_id</b> : The client_id. This can only be used INSTEAD of using the authorization header. <b>client_secret</b> : The client_secret. This can only be used INSTEAD of using the authorization header. <b>grant_type</b> : MUST be set to 'authorization_code' <b>code</b> : The authorization_code that was requested using the 'response_type=code' flow. <b>redirect_uri</b> (optional) : A 'redirect_uri' that was registered for this

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params (for GET attach params to URL-PATH)	Comment
						client. If used during the initial code request, it is required and must match the original value.
Response					Pragma { "access_token":"an_access_token", "token_type":"Bearer", "expires_in":3600, "refresh_token": "cache_a_refresh_token", "scope":"granted_scope" } Cache-Control: no-store status: 200 content-type: application/json; charset=UTF-8	The access_token can be used to access protected resources. The refresh_token can be exchanged for an access_token and a new refresh_token.
Response					Pragma { "access_token":"an_access_token", "token_type":"Bearer", "expires_in":3600, "refresh_token": "cache_a_refresh_token", "scope":"granted_scope", "id_token":"an_id_token", "id_token_type":"id_token_type" } Cache-Control: no-store status: 200 content-type: application/json; charset=UTF-8	This response gets returned if the request included SCOPE=openid.
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The given mag-identifier is either invalid or points 300310 to an unknown device" } 7 Pragma : no-	[MAG] invalid mag-identifier.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	
Error- Response				x-ca- err:	{ "error":"invalid_grant", "error_description":"The given grant is invalid" } 300311 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	The given grant has expired or is invalid.
Error- Response				x-ca- err:	{ "error":"invalid_redirect_uri", "error_description":"Mismatching redirect_uri" } 300311 4 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica	Invalid redirect_uri . .

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error-Response	x-ca-err: 300310 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	{ "error":"invalid_request", "error_description":"Missing or duplicate parameters" }	If any required parameters or headers are missing, the request will fail.		
Error-Response	x-ca-err: 300311 7 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	{ "error":"unauthorized_client", "error_description":"The client lacks authorization for this request" }	An unauthorized client is using the code.		
Error-Response	x-ca-err: 300311 9 Pragma	{ "error":"unsupported_grant_type", "error_description":"The given grant_type is not supported" }	unsupported grant_type		

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				: no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	
Error- Response		x-ca- err: 300320 1 WWW- Authen ticate: Basic error=" Invalid or missing credent ials" Pragma : no- cache Cache- Control : no- store status: 401 content -type: applica tion /json; charset =UTF-8	{ "error":"invalid_client", "error_description":"The given client credentials were not valid" }	The client could not be authenticat ed. <b>WWW- Authentica te</b> (optional) : This header is required by <a href="https://tools.ietf.org/html/rfc7235">RFC 7235 (https: //tools.ietf.org/html /rfc7235)</a> if an authentication scheme has been used. It contains the used authentication scheme and an error message.	
Error- Response		x-ca- err: 300320 3 Pragma	{ "error":"invalid_request", "error_description":"SSL is required" }	Forbidden.	

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					: no-cache Cache-Control : no-store status: 403 content-type: application/json; charset =UTF-8	
Error-Response					Allow: POST Pragma : no-cache Cache-Control : no-store status: 405	The HTTP method is not valid. <b>Allow :</b> This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
Error-Response					x-ca-err: 300300 0 Pragma : no-cache Cache-Control : no-store status: 500	Unknown error.
e			POST		client_id=<client_id>&	

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
n	Request access_to h	/aut	mag-	client_secret=<client_secret>&		mag- identifier
_r	ken using OAuth 2.0	/oau	identifi	grant_type=refresh_token&		(optional) :
ef	th	/v2	er:	scope=<scope>&		[MAG]
re	grant_typ /tok		authori	refresh_token=<refresh_token>		
sh	e=refresh		zation:			
_f	_token.		content			
See	RFC 6749 (http://tools.ietf.org/html/rfc6749)		-type:			
lo			applica			
			tion/x-			
			www-			
			form-			
			urlenco			
			ded			
	for more details					

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
						using the authorization header. <b>client_secret</b> : The client_secret. This can only be used INSTEAD of using the authorization header. <b>grant_type</b> : MUST be set to 'refresh_token' <b>scope</b> (optional) : Only the SCOPE values or a subset of those values originally requested will be granted.
Response					Pragma { "access_token":"an_access_token", "token_type":"Bearer", "expires_in":3600, "refresh_token": "a_refresh_token", "scope":"granted_scope" } Cache-Control : no-store status: 200 content-type: application/json; charset =UTF-8	The access_token can be used to access protected resources. The refresh_token can be exchanged for an access_token and a new refresh_token.
Error-Response		x-ca-err:			{ "error":"invalid_request", "error_description":"The given mag-identifier is either invalid or points 300310 to an unknown device" }	[MAG] invalid mag-identifier.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					<pre> Pragma : no- cache Cache- Control : no- store status: 400 content -type:  applica tion /json; charset =UTF-8 </pre>	
Error- Response				x-ca- err:	<pre> { "error":"invalid_grant", "error_description":"The given grant is invalid" } 300311 3 </pre>	The given grant has expired or is invalid.
Error- Response				x-ca- err:	<pre> { "error":"invalid_request", "error_description":"" Missing or duplicate parameters" } 300310 3 </pre>	If any required parameters or headers are missing, the request will fail.

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				-type: application /json; charset =UTF-8	
Error-Response		x-ca-err:	{ "error":"invalid_scope", "error_description":"No registered scope value for this client has been 300311 requested" }	5	invalid scope
		Pragma	: no-	cache	
		Cache- Control	: no-	store	
		status:	400	content	
		-type: application /json; charset =UTF-8			
Error-Response		x-ca-err:	{ "error":"unauthorized_client", "error_description":"The client lacks authorization 300311 for this request" }	7	An unauthorized client is using the code.
		Pragma	: no-	cache	
		Cache- Control	: no-	store	
		status:	400	content	
		-type: application /json; charset =UTF-8			
Error-Response		x-ca-err:	{ "error":"unsupported_grant_type", "error_description":"The given grant_type is not 300311 supported" }		unsupported grant_type

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				<pre> 9 Pragma : no- cache Cache- Control : no- store status: 400 content -type:  applica tion /json; charset =UTF-8 </pre>	
Error- Response		x-ca- err:	{ "error":"invalid_client", "error_description":"The given client credentials were not valid" }	300320 1 WWW- Authen ticate: Basic error=" Invalid or missing credent ials" Pragma : no- cache Cache- Control : no- store status: 401 content -type:  applica tion /json; charset =UTF-8	The client could not be authenticated. <b>WWW-Authentica</b> te(optional) : This header is required by <a href="https://tools.ietf.org/html/rfc7235">RFC 7235</a> if an authentication scheme has been used. It contains the used authentication scheme and an error message.
Error- Response		x-ca- err:	{ "error":"invalid_request", "error_description":"Validation error" }	300399	

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				3 Pragma : no- cache Cache- Control : no- store status: 401 content -type: applica tion /json; charset =UTF-8	The given token is not valid, it is disabled.
Error- Response		x-ca- err:	{ "error":"invalid_request", "error_description":"SSL Forbidden. is required" }	300320 3 Pragma : no- cache Cache- Control : no- store status: 403 content -type: applica tion /json; charset =UTF-8	
Error- Response		Allow: POST Pragma : no- cache Cache- Control : no- store status: 405			The HTTP method is not valid <b>Allow :</b> This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https: //tools.ietf.org/html /rfc2616</a> ) and

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
						contains a comma separated list of valid HTTP methods.
	Error- Response			x-ca-err: 300300 0 Pragma: no-cache Cache-Control: no-store status: 500		Unknown error.
e	Request access_to_h ken using /oau _c the OAuth 2.0 /v2 /tok e grant_typ en nt e=client_c redentials _c . SeeRFC re 6749 (http ds s://tools. ietf.org _f /html lo /rfc6749) w for more details	/aut	POST	mag- identifi er: authori zation: content -type: applica tion/x-www- form- urlenco ded	client_id=<client_id>& client_secret=<client_secret>& grant_type=client_credentials& scope=<scope>	mag- identifier (optional) : [MAG] mag- identifier (optional) : [MAG]: mobile clients MUST use this header. It must represent a valid registered device. <b>authorizati on</b> : The HTTP basic authorization header containing the client credentials as base64 encoded string (authorizati

ID	Operation	URL-	HTTP Path	HTTP Method	Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
							on: Basic base64 (client_id: client_secr et)). This can only be used INSTEAD of the parameters 'client_id', 'client_secr et'. <b>client_id</b> : The client_id. This can only be used INSTEAD of using the authorization header. <b>client_secr et</b> : The client_secr et. This can only be used INSTEAD of using the authorization header. <b>grant_type</b> : MUST be set to 'client_credentials' <b>scope</b> (optional) : Only SCOPE values that were registered for this client will be granted. If only non-matching

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					SCOPE values are requested, the request will fail.
Response				<pre>Pragma { "access_token":"an_access_token", "token_type":"The : no-   Bearer", "expires_in":3600, "scope":"" cache  granted_scope" } Cache- Control : no- store status: 200 content -type:  applica tion /json; charset =UTF-8</pre>	The access_token can be used to access protected resources. The refresh_token can be exchanged for an access_token and a new refresh_token.
Error-Response				<pre>x-ca- { "error":"invalid_request", "error_description":"" err: The given mag-identifier is either invalid or points 300310 to an unknown device" } 7 Pragma : no- cache Cache- Control : no- store status: 400 content -type:  applica tion /json; charset =UTF-8</pre>	[MAG] invalid mag- identifier.
Error-Response				<pre>x-ca- { "error":"invalid_request", "error_description":"" err: Missing or duplicate parameters" } 300310 3 Pragma : no-</pre>	If any required parameters or headers are missing,

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8		the request will fail.
Error- Response			x-ca- err:	{ "error":"invalid_scope", "error_description":"No registered scope value for this client has been 300311 requested" }	5 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	invalid scope
Error- Response			x-ca- err:	{ "error":"unsupported_grant_type", "error_description":"The given grant_type is not 300311 supported" }	9 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica	unsupporte d grant_type

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error- Response					tion /json; charset =UTF-8	
Error- Response					x-ca-err: { "error":"invalid_client", "error_description":"The given client credentials were not valid" } 300320 1 WWW-Authen ticate: Basic error=" Invalid or missing credent ials" Pragma : no- cache Cache- Control : no- store status: 401 content -type: applica tion /json; charset =UTF-8	The client could not be authenticated. <b>WWW-Authenticat</b> e(optional) : This header is required by <a href="https://tools.ietf.org/html/rfc7235">RFC 7235</a> if an authentication scheme has been used. It contains the used authentication scheme and an error message.
Error- Response					x-ca-err: { "error":"invalid_request", "error_description":"SSL is required" } 300320 3 Pragma : no- cache Cache- Control : no- store status: 403 content -type: applica	

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				tion /json; charset =UTF-8	
Error-Response				Allow: POST Pragma : no- cache Cache- Control : no- store status: 405	The HTTP method is not valid. <b>Allow :</b> This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> (https://tools.ietf.org/html/rfc2616) and contains a comma separated list of valid HTTP methods.
Error-Response				x-ca- err: 300300 0 Pragma : no- cache Cache- Control : no- store status: 500	Unknown error.
e n _j w t fl o w	Request /aut POST access_to h ken using /oau the th OAuth /v2 2.0 /tok extension en grant_typ e=urn: ietf: params:			client_id=<client_id>& identifi er: client_secret=<client_secret>& authori grant_type=jwt-bearer& zation: scope=<scope>& content assertion=<assertion> -type: aplica tion/x- www- form- urlenco ded	mag- identifier (optional) : [MAG]  mag- identifier (optional) : [MAG]: mobile clients MUST use this

ID	Operation	URL-	HTTP Path	HTTP Method	Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
<b>oauth: grant- type:jwt- bearer</b>							<p>header. It must represent a valid registered device.</p> <p><b>authorizati</b> <b>on</b> : The HTTP basic authorization header containing the client credentials as base64 encoded string (authorizati on: Basic base64 (client_id: client_secr et)). This can only be used INSTEAD of the parameters 'client_id', 'client_secr et'.</p> <p><b>client_id</b> : The client_id. This can only be used INSTEAD of using the authorization header.</p> <p><b>client_secr</b> <b>et</b> : The client_secr et. This can only be used INSTEAD of using the authorization header.</p>

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
						<p><b>grant_type</b> : MUST be set to 'urn:ietf:params:oauth:grant-type:jwt-bearer'.  <b>scope</b> (optional) : Only SCOPES values that were registered for this client will be granted. If only non-matching SCOPES values are requested, the request will fail.  <b>id_token</b> : This parameter MUST contain an id_token (represented as JWT) that represents an authenticated resource owner.</p>
Response					Pragma { "access_token":"an_access_token", "token_type":"The access_token", "expires_in":3600, "refresh_token":"a_refresh_token", "scope":"granted_scope" } Cache-Control : no-store status:	can be used to access protected resources. The

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				200 content -type: applica tion /json; charset =UTF-8	refresh_tok en can be exchanged for an access_tok en and a new refresh_tok en.
Error- Response		x-ca- err: 7		{ "error":"invalid_request", "error_description":" The given mag-identifier is either invalid or points 300310 to an unknown device" }	[MAG] invalid mag- identifier.
Error- Response		x-ca- err: 300310 3		{ "error":"invalid_request", "error_description":" Missing or duplicate parameters" }	If any required parameters or headers are missing, the request will fail.

ID	Operation URL- Path	HTTP Method	HTTP Header	Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error-Response		x-ca-err:	{ "error":"invalid_scope", "error_description":"No registered scope value for this client has been requested" }	5	invalid scope
Error-Response		x-ca-err:	{ "error":"unauthorized_client", "error_description":"The client lacks authorization for this request" }	7	An unauthorized client is using the code.
Error-Response		x-ca-err:	{ "error":"unsupported_grant_type", "error_description":"The given grant_type is not supported" }	9	unsupported grant_type

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					store status: 400 content -type: applica tion /json; charset =UTF-8	
Error- Response			x-ca- err: 300320 1 WWW- Authen ticate: Basic error=" Invalid or missing credent ials" Pragma : no- cache Cache- Control : no- store status: 401 content -type: applica tion /json; charset =UTF-8		{ "error":"invalid_client", "error_description":"The given client credentials were not valid" }	The client could not be authenticated. <b>WWW-Authentica</b> te(optional) : This header is required by <a href="https://tools.ietf.org/html/rfc7235">RFC 7235</a> if an authentication scheme has been used. It contains the used authentication scheme and an error message.
Error- Response			x-ca- err: 300320 2 Pragma : no- cache Cache- Control : no-		{ "error":"invalid_request", "error_description":"The resource owner could not be authenticated due to missing or invalid credentials" }	The resource owner could not be authenticated

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				store status: 401 content -type: application /json; charset =UTF-8	
Error-Response		x-ca-err:	{ "error":"invalid_request", "error_description":"SSL Forbidden. is required" }	300320 3 Pragma : no- cache Cache- Control : no- store status: 403 content -type: application /json; charset =UTF-8	
Error-Response		Allow:	POST	Pragma : no- cache Cache- Control : no- store status: 405	The HTTP method is not valid <b>Allow :</b> This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment	
Error-Response				x-ca-err: 300300 0 Pragma : no- cache Cache- Control : no- store status: 500		Unknown error.	
e n _s a m L f o w	Request n ken using _s the a OAuth 2.0 m extension grant_typ e=urn: ietf: params: oauth: grant- type: saml2- bearer	/aut /oau /th /v2 /tok en applica tion/x- www- form- urlenco ded	POST	mag- identifi er: authori zation: content -type: aplica tion/x- www- form- urlenco ded	client_id=<client_id>& client_secret=<client_secret>& grant_type=saml-bearer& scope=<scope>& assertion=<assertion>	mag- identifier (optional) : [MAG]	

ID	Operation	URL-	HTTP Path	HTTP Method	Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
							on: Basic base64 (client_id: client_secr et)). This can only be used INSTEAD of the parameters 'client_id', 'client_secr et'. <b>client_id</b> : The client_id. This can only be used INSTEAD of using the authorization header. <b>client_secr et</b> : The client_secr et. This can only be used INSTEAD of using the authorization header. <b>grant_type</b> : MUST be set to 'urn:ietf:params:oauth:grant-type:saml2-bearer'. <b>scope</b> (optional) : Only SCOPE values that were registered for this client will

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
						be granted. If only non-matching SCOPE values are requested, the request will fail. <b>assertion :</b> This parameter must contain a base64 encoded SAML token that represents an authenticated resource owner.
Response					Pragma { "access_token":"an_access_token", "token_type":"The : no- Bearer", "expires_in":3600, "scope":" cache granted_scope" } Cache- Control : no- store status: 200 content -type: applica tion /json; charset =UTF-8	access_token can be used to access protected resources. The refresh_token can be exchanged for an access_token and a new refresh_token.
Error- Response				x-ca- err:	{ "error":"invalid_request", "error_description":" The given mag-identifier is either invalid or points 300310 to an unknown device" }	[MAG] invalid mag- identifier.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	
Error- Response		x-ca- err: 300310 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8		{ "error":"invalid_request", "error_description":"Missing or duplicate parameters" }		If any required parameters or headers are missing, the request will fail.
Error- Response		x-ca- err: 300311 5 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica		{ "error":"invalid_scope", "error_description":"No registered scope value for this client has been requested" }		invalid scope

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error- Response					tion /json; charset =UTF-8	
Error- Response					x-ca- { "error":"unsupported_grant_type", err: "error_description":"The given grant_type is not 300311 supported" } 9 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	unsupporte d grant_type
Error- Response					x-ca- { "error":"invalid_client", "error_description":"The err: given client credentials were not valid" } 300320 1 WWW- Authen ticate: Basic error=" Invalid or missing credent ials" Pragma : no- cache Cache- Control : no- store status: 401 content -type: applica	The client could not be authenticat ed. <b>WWW-Authentica te</b> (optional) : This header is required by <a href="#">RFC 7235</a> ( <a href="https://tools.ietf.org/html/rfc7235">https://tools.ietf.org/html/rfc7235</a> ) if an authentication scheme has been used. It contains the used authentication

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					tion /json; charset =UTF-8	scheme and an error message.
Error- Response			x-ca- err:	{ "error":"invalid_request", "error_description":"The resource owner could not be authenticated due to missing or invalid credentials" }	2 Pragma : no- cache Cache- Control : no- store status: 401 content -type: applica tion /json; charset =UTF-8	The resource owner could not be authenticat ed
Error- Response			x-ca- err:	{ "error":"invalid_request", "error_description":"SSL is required" }	300320 3 Pragma : no- cache Cache- Control : no- store status: 403 content -type: applica tion /json; charset =UTF-8	SSL Forbidden. is required"
Error- Response			Allow: POST Pragma : no- cache			The HTTP method is not valid <b>Allow :</b> This

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				Cache- Control : no- store status: 405		header is required by <a href="#">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
	Error- Response			x-ca- err: 300300 0 Pragma : no- cache Cache- Control : no- store status: 500		Unknown error.
<a href="#">§</a>	<b>Revoke</b>  an h access_to /oau ken or th refresh_t /v2 oken. See /tok <a href="#">RFC 7009</a> ( <a href="https://tools.ietf.org/html/rfc7009">https://tool/rev s.ietf.org/oke</a> <a href="#">/html</a> <a href="#">/rfc7009</a> ) <b>for more</b> <b>details.</b> <b>Section</b> <b>2.3 of</b> <b>that RFC</b> <b>is not</b> <b>supported</b> •	/aut	POST	authori zation: token_type_hint=<token_type_hint> content -type: applica tion/x- www- form- urlenco ded		The token can only be revoked by the client that initially requested and received it! This API returns a success response even if the given token does not exist or has already been revoked.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
						<b>authorizati</b> <b>on</b> : The HTTP basic authorizati on header containing the client credentials as base64 encoded string (authorizati on: Basic base64 (client_id: client_secr et)). For public clients, the client_secr et is not required. <b>token</b> : The token as it was issued. <b>token_type</b> <b>_hint</b> : Valid values are 'access_tok en' and 'refresh_to ken'.  <b>Response</b>
					status: {"result":"revoked"} 200 content -type: applica tion /json; charset =UTF-8	The token was successfully revoked or it was invalid.
	Error- Response			x-ca- err:	{ "error":"invalid_request", "error_description": "Missing or duplicate parameters" } 300410 3 Pragma : no- cache Cache-	If any required parameters or headers are

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	missing, the request will fail.
Error- Response		x-ca- err:	{ "error":"unauthorized_client", "error_description":"The client lacks authorization 300411 for this request" }	7 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	unauthoriz ed client
Error- Response		x-ca- err:	{ "error":"invalid_client", "error_description":"The given client credentials were not valid" }	300420 1 WWW- Authen ticate: Basic error=" Invalid or missing credent ials" Pragma : no- cache Cache-	The client could not be authenticat ed. <b>WWW- Authentica te</b> (optional) : This header is required by <a href="https://tools.ietf.org/html/rfc7235">RFC 7235 (https: //tools.ietf.org/html /rfc7235)</a> if an authenticat

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				Control : no- store status: 401 content -type: applica tion /json; charset =UTF-8	ion scheme has been used. It contains the used authenticat ion scheme and an error message.
Error- Response	x-ca- err:	300420 3	{ "error":"invalid_request", "error_description":"SSL Forbidden. is required" }	Pragma : no- cache Cache- Control : no- store status: 403 content -type: applica tion /json; charset =UTF-8	
Error- Response			Allow: POST, DELETE	Allow: POST, DELETE	The HTTP method is not valid <b>Allow :</b> This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rf c2616</a> ) and contains a comma

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error-Response	x-ca-err: 8	{ "error":"unsupported_token_type", "error_description":"the given type of token is not supported" }	Pragma: no-cache	Cache-Control: no-store	separated list of valid HTTP methods.
			status: 503	content-type: application/json;	unsupported token type
			charset =UTF-8		
ils	The endpoint returns details about a registered OAuth client.	/aut /oau /v2 /clie nt /exp ort	GET	authori zation: content-type: applica tion/x-www-form-urlencoded	This endpoint is disabled by default. If requests fail due to an unknown endpoint, contact the system administrator.
					<b>authorizati on :</b> The HTTP basic authorization header containing resource_owner credentials as base64

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					encoded string (authorizati on: Basic base64 (username: password)). Depending on the server configurati on this header may not be required. <b>cli ent_id</b> : The client_id for which details are requested.	
Response					<pre> status: { "server": { "hostname": "example.ca.com (<a href="http://example.ca.com">http://example.ca.com</a>)", "port": 9443, "prefix": "/urlprefix", "content" "server_certs": [ [ "-----BEGIN CERTIFICATE-----", "-type: "MIIC...ybEYFkq", "BLQ=", "-----END CERTIFICATE-----" ] ] }, "oauth": { "client": { "organization": "CA Technologies", "description": "/json; "Example application for Mobile SSO Demo", "charset": "client_name": "AppA", "client_type": "=UTF-8 "confidential", "registered_by": "John Doe", "client_ids": [ { "client_id": "84695 ... b39770c3d", "client_secret": "abcde ... IdjjakkC", "scope": "openid email profile", "redirect_uri": "<a href="https://example.com/client/consent?state=23n23n...13k1j4">https://example.com/client/consent?state=23n23n...13k1j4</a>", "environment": "ALL", "status": "ENABLED", "registered_by": "John Doe" } ] }, "system_endpoints": { "authorization_endpoint_path": "/auth/oauth/v2 /authorize", "token_endpoint_path": "/auth/oauth /v2/token", "token_revocation_endpoint_path": "/auth/oauth/v2/token/revoke", "usersession_logout_endpoint_path": "/connect /session/logout" }, "oauth_protected_endpoints": { "userinfo_endpoint_path": "/openid/connect/v1 /userinfo", "usersession_status_endpoint_path": "/connect/session/status" } }, "custom": { "mag_demo_products_endpoint_path": "/protected /resource/products", "oauth_demo_protected_api_endpoint_path": "/oauth/v2/protectedapi/foo" } } </pre>	A JSON message including details about a client in order to configure it. The client_secret may or may not be included depending on the OAuth server configuration. The JSON message will also include the public certificate of the OAuth server as PEM.

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error-Response		x-ca-err:	{ "error":"invalid_request", "error_description":"Missing or duplicate parameters" }	300510 3 Pragma : no-cache Cache-Control : no-store status: 400 content-type: application/json; charset= UTF-8	If any required parameters or headers are missing, the request will fail.
Error-Response		x-ca-err:	{ "error":"invalid_request", "error_description":"The server configuration is invalid. Contact the administrator" }	300513 2 Pragma : no-cache Cache-Control : no-store status: 400 content-type: application/json; charset= UTF-8	If the server cert to be exported has not been configured correctly.
Error-Response		x-ca-err:	{ "error":"invalid_client", "error_description":"The client is unknown or disabled" }	300520 1 Pragma : no-cache Cache-Control : no-	The client is invalid.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					store status: 401 content -type: applica tion /json; charset =UTF-8	
Error- Response		x-ca- err:	{ "error":"invalid_request", "error_description": " The resource owner could not be authenticated due to missing or invalid credentials" } 2 WWW- Authen ticate: Basic error=" Invalid or missing credent ials" Pragma : no- cache Cache- Control : no- store status: 401 content -type: applica tion /json; charset =UTF-8			The resource owner could not be authenticat ed <b>WWW- Authentica te</b> : This header is required by <a href="https://tools.ietf.org/html/rfc7235">RFC 7235</a> ( <a href="https://tools.ietf.org/html/rfc7235">https: //tools.ietf. org/html /rfc7235</a> ) and will contain the used authentication scheme and an error message.
Error- Response		x-ca- err:	{ "error":"invalid_request", "error_description":"SSL is required" } 300520 3 Pragma : no- cache Cache- Control : no-			Forbidden. 300520 3 SSL is required

ID	Operation URL-	HTTP Path	HTTP Method	Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					store status: 403 content -type: applica tion /json; charset =UTF-8	
Error- Response					Allow: GET, POST Pragma : no- cache Cache- Control : no- store status: 405	The HTTP method is not valid <b>Allow :</b> This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https:// //tools.ietf. org/html /rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
Error- Response					x-ca- err: 300500 0 Pragma : no- cache Cache- Control : no- store status: 500	Unknown error.
<i>n</i> <b>owner</b> <b>er</b> <i>_l</i>	<b>The resource owner deletes his active session</b>	/con nect /sess ion /log out	POST	mag- nect /sess ion /log out	logout_apps=true& identifi er: id_token=<id_token>& authori zation: jwt-bearer content	<b>mag- identifier</b> (optional) : [MAG]: mobile clients

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
<b>o</b> <b>on the OAuth server.</b>				-type: application/x-www-form-urlencoded	MUST use this header. It must represent a valid registered device.

**logout\_apps** (optional) : [MAG]: mobile clients MAY use this parameter to invalidate all related oauth token.

**authorization** : The HTTP basic authorization header containing the client credentials as base64 encoded string (authorization: Basic base64 (client\_id: client\_secret)).

**id\_token** : The id\_token that represents the authenticated user.

**id\_token\_type** (optional) : The id\_token\_t

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					ype to be invalidated. Required if not default type.
Response				status: {"session_status":"logged out"} 200 content -type: applica tion /json; charset =UTF-8	Resource owner successfully logged out
Error-Response				x-ca- { "error":"unauthorized_client", err: "error_description":"The client lacks authorization 300611 for this request" } 7 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	unauthorized client
Error-Response				x-ca- { "error":"invalid_client", "error_description":"The err: given client credentials were not valid" } 300620 1 WWW- Authen ticate: Basic error=" Invalid or missing credent ials" Pragma	The client could not be authenticated. <b>WWW-Authentica</b> te(optional): This header is required by <a href="https://tools.ietf.org/html/RFC7235">RFC 7235</a> ( <a href="https://tools.ietf.org/html/RFC7235">https://tools.ietf.org/html/RFC7235</a> )

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				: no-cache Cache-Control : no-store status: 401 content-type: application/json; charset =UTF-8	/rfc7235) if an authentication scheme has been used. It contains the used authentication scheme and an error message.
Error-Response		x-ca-err:	{ "error":"invalid_request", "error_description":"SSL Forbidden." } 300620 3 Pragma : no-cache Cache-Control : no-store status: 403 content-type: application/json; charset =UTF-8		is required"
Error-Response			Allow: POST Pragma : no-cache Cache-Control : no-store status: 405		The HTTP method is not valid <b>Allow :</b> This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma

ID	Operation URL- Path	HTTP Method	HTTP Header	Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error-Response	x-ca-err: 300600 0 Pragma: no-cache Cache-Control: no-store status: 500				separated list of valid HTTP methods.
<i>N</i> The client requests the session status by passing in the id_token of the authentic ated user.  <i>S</i> ta tu s	authori zation: id_token_type=urn:ietf:params:oauth:grant-type: content jwt-bearer -type: applica tion/x-www-form-urlencoded				Unknown error.
					This informs the client if the resource owner is logged in and has a valid session. The session may be active or not existing.
					<b>authorizati on</b> : A valid access_token (e.g.: authorizati on: Bearer access-token-value) <b>id_token</b> : The id_token of the resource

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
						ownerid_to ken_type(optional) : The id_token_t ype to be invalidated. Required if not default type.
Response					status: {"session":"active"} 200 content -type: applica tion /json; charset =UTF-8	Session is active
Response					status: {"session_status":"none"} 200 content -type: applica tion /json; charset =UTF-8	Session does not exist
Error- Response			x-ca- err:	300710 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	{ "error":"invalid_request", "error_description":"Missing or duplicate parameters" } If any required parameters or headers are missing, the request will fail.	

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error-Response		x-ca-err:	{ "error":"invalid_request", "error_description":"Validation error" }	300799 0 Pragma : no-cache Cache-Control : no-store status: 401 content-type: application/json; charset =UTF-8	The given token is not valid. The error code may be 1003990, 1003991, 1003992, 1003993
Error-Response		x-ca-err:	{ "error":"invalid_request", "error_description":"SSL is required" }	300720 3 Pragma : no-cache Cache-Control : no-store status: 403 content-type: application/json; charset =UTF-8	
Error-Response		Allow:	GET, POST Pragma : no-cache Cache-Control		The HTTP method is not valid <b>Allow :</b> This header is required by <a href="https://tools.ietf.org/html/RFC 2616">RFC 2616 (https://tools.ietf.org/html/RFC 2616)</a>

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
		: no-store				/rfc2616) and contains a comma separated list of valid HTTP methods.
	Error-Response	x-ca-err: 300700 0 Pragma : no-cache Cache-Control : no-store status: 500				Unknown error.

## OAuth Validation Point (OVP) API

There are several endpoints that are used to validate requests using OAuth.

All endpoints described start here:

<Gateway\_host\_and\_port>/oauth/validation/\*

For example:

\_http://my.securespan.gateway.com:8080/oauth/validation/v1/authorize\_  
If an endpoint contains "v1", then it is used by OAuth 1.0. "v2" is used by OAuth 2.0

### Endpoints

- [validate\\_client \(see page 185\)](#)
- [validate\\_token \(see page 191\)](#)
- [validate\\_refresh\\_token \(see page 194\)](#)
- [token\\_revocation \(see page 197\)](#)
- [validate\\_id\\_token \(see page 200\)](#)
- [create\\_id\\_token \(see page 204\)](#)

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
e			POST		client_key=<client_key>&	

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params	Comment
nt	Validates the given values of the given client_key. It also returns the registered values of this client (except for the secret). It is not to be exposed to any client, only shared between different OTK servers.	/oau/vali/v2/clients	POST		content-type: application/x-www-form-urlencoded client_secret=<client_secret>&client_name=<client_name>&client_type=<client_type>&environment=<environment>&urlenco >&ded expiration=true&status=ENABLED&scope=oob	<p>Authentication is done via ssl mutual authentication.</p> <p><b>client_key</b> : the client_id of the client to be validated. It is invalid if it matches the 'client_secret'.</p> <p><b>client_secret</b> : the expected client_secret of the client_id.</p> <p><b>client_name</b> (optional) : the expected client_name.</p> <p><b>client_type</b> (optional) : the expected client_type, either 'confidential' or 'public'.</p> <p><b>environment</b> (optional) : the expected environment.</p> <p><b>expiration</b> (optional) : either 'true' or 'false'.</p> <p><b>status</b> (optional) : the expected device status. Either 'ENABLED' or 'DISABLED'.<b>scope</b> (optional) : the expected SCOPE. The client must have been registered for the given values.</p>
Response					status: 200 content-type: application/xml; charset=UTF-8	<validation xmlns="http://ns.l7tech.com/2012/11/otk"> <result>valid</result> <client_type>value</client_type> <client_name>value</client_name> <environment>value</environment> <created_by>value</created_by> </validation>
Error-Response					x-ca-err: 500010 Pragma	{ "error":"invalid_request", "error_description":"Missing or unknown parameters" } If any required parameters or headers are missing, the request will fail. : no-cache

ID Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params	Comment
				(for GET attach params to URL-PATH)	
				<pre>Cache- Control : no- store status: 400 content -type:  applica tion /json; charset =UTF-8</pre>	
Error-Response		x-ca- err:	{ "error":"invalid_scope", "error_description":"No registered scope value for this client has been requested" }		invalid scope
				<pre>: no- cache Cache- Control : no- store status: 400 content -type:  applica tion /json; charset =UTF-8</pre>	
Error-Response		x-ca- err:	{ "error":"invalid_request", "error_description":"The given status does not match the actual client status" }		invalid status
				<pre>Pragma : no- cache Cache- Control : no- store status: 400 content -type:  applica</pre>	

ID Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				tion /json; charset =UTF-8	
Error-Response			x-ca- err: 500012 7	{ "error":"invalid_request", "error_description":"The given client_id has expired" }	
			Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8		
Error-Response			x-ca- err: 500012 8	{ "error":"invalid_request", "error_description":"The client name is not valid" } 8	
			Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8		
Error-Response			x-ca- err: 500012 9	{ "error":"invalid_request", "error_description":"The environment is not valid" }	

ID Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				<pre> Pragma : no- cache Cache- Control : no- store status: 400 content -type:  applica tion /json; charset =UTF-8 </pre>	
Error-Response				<pre> x-ca- { "error":"invalid_request", invalid client type err:   "error_description":"The 500013 client type is not valid" } 0 Pragma : no- cache Cache- Control : no- store status: 400 content -type:  applica tion /json; charset =UTF-8 </pre>	x-ca- { "error":"invalid_request", invalid client type err:   "error_description":"The 500013 client type is not valid" } 0
Error-Response				<pre> x-ca- { "error":"invalid_client", The client could not be err:   "error_description":"The 500020 given client credentials 1      were not valid" } 1 Pragma : no- cache Cache- Control : no- store status: 401 </pre>	The client could not be authenticated.

ID Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				content -type: application /json; charset =UTF-8	
Error-Response				x-ca-err: 500020 4 Pragma : no- cache Cache- Control : no- store status: 403 content -type: application /json; charset =UTF-8	{ "error":"invalid_request", "error_description":"SSL with client authentication is required" }
Error-Response				x-ca-err: 500020 5 Pragma : no- cache Cache- Control : no- store status: 401 content -type: application /json; charset =UTF-8	{ "error":"invalid_request", "error_description":"The client could not be authenticated." }
Error-Response					

ID Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				Allow: POST, GET Pragma : no- cache Cache- Control : no- store status: 405	The HTTP method is not valid <b>Allow</b> : This header is required by <a href="http://tools.ietf.org/html/rfc2616">RFC 2616 (http://tools.ietf.org/html/rfc2616)</a> and contains a comma separated list of valid HTTP methods.
<b>ke</b> <a href="#">Validates the given access_token.</a> <b>n</b>	/oau POST			authori zation: n>& content -type: isonetime=false& on applica tion/x- /vali date /v2 /tok en	Authentication is done via ssl mutual authentication. Validates 'BEARER' tokens (passed in as 'access_token') and 'MAC' tokens (passed in as 'authorization' header). However, only token types supported by the server can be successfully validated. At least one type of token has to be used, either through 'access_token' or 'authorization'.
				scope_required=oob& tion/xe- scope_fail=false& www- http_method=<http_metho form- d>& urlenco url=<url> ded	<b>authorization</b> (optional) : 'MAC' token only: the MAC access_token to be validated. Required for MAC tokens. <b>access_token</b> (optional) : the BEARER access_token to be validated. Required for BEARER tokens. <b>isonetime</b> : Use 'false' to allow the token to be used again. Use 'true' if the token should be invalidated. <b>scope_required</b> (optional) : the list of required SCOPES will be verified against the SCOPES associated with the given token.

ID Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					<b>scope_fail</b> (optional) : specify if the validation should fail if the 'scope_required' is associated with the given token. <b>http_method</b> (optional) : 'MAC' token only: the HTTP method originally used by the requesting client. <b>url</b> (optional) : 'MAC' token only: the URL originally used by the requesting client.
Response				<pre>status: &lt;validation xmlns="http://ns 200 .l7tech.com/2012/11/otk- content ovp"&gt; &lt;result&gt;valid&lt; -type: /result&gt; &lt;client_key&gt;value&lt; text /client_key&gt; /xml; &lt;resource_owner&gt;value&lt; charset /resource_owner&gt; =UTF-8 &lt;scope&gt;value&lt;/scope&gt; &lt;expiration&gt;value&lt; /expiration&gt; &lt;/validation&gt;</pre>	
Error-Response				<pre>x-ca- { "error":"invalid_request", err: "error_description":" 500110 Missing or unknown 3 parameters" }</pre>	If any required parameters or headers are missing, the request will fail.
Error-Response				<pre>Pragma : no- cache Cache- Control : no- store status: 400 content -type:  applica tion /json; charset =UTF-8</pre>	
Error-Response				<pre>x-ca- { "error":"invalid_request", err: "error_description":"SSL 500120 with client authentication is 4 required" }</pre>	Authentication failed.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					<pre>Pragma : no- cache Cache- Control : no- store status: 403 content -type:  applica tion /json; charset =UTF-8</pre>	
	Error-Response				<pre>x-ca- { "error":"invalid_request", err: "error_description":"The 500120 client certificate is not 5 valid" }</pre> <pre>Pragma : no- cache Cache- Control : no- store status: 401 content -type:  applica tion /json; charset =UTF-8</pre>	The client could not be authenticated.
	Error-Response				<pre>x-ca- { "error":"invalid_request", err: "error_description":" 500199 Validation error" } 0</pre> <pre>Pragma : no- cache Cache- Control : no- store status: 401</pre>	The given token is not valid. The error code may be 1003990, 1003991, 1003992, 1003993

ID Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				content -type: application /json; charset =UTF-8	
Error-Response				Allow: POST, GET Pragma : no- cache Cache- Control : no- store status: 405	The HTTP method is not valid <b>Allow</b> : This header is required by <a href="http://tools.ietf.org/html/rfc2616">RFC 2616 (http://tools.ietf.org/html/rfc2616)</a> and contains a comma separated list of valid HTTP methods.
validate_refreshtoken	Validates the given refresh_token.	/oau POST		content rtoken=<rtoken>&th=validatetoken/v2/refresheshtoken	Authentication is done via ssl mutual authentication.  <b>rtoken</b> : the refresh_token to be validated. <b>client_key</b> : the client_id to which the refresh_token must have been issued. <b>scope</b> (optional) : the response includes the SCOPE values that are valid for this request. If the original SCOPE is 'oob', it is returned and the passed in values are not respected. The request fails if the SCOPE values do not match any original SCOPE values.
Response				status: <validation xmlns="http://ns.l7tech.com/2012/11/otk-content ovp"> <result>valid</result> <client_key>value</client_key> <resource_owner>value</resource_owner>	

ID Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				text <scope>value</scope> /xml; <expiration>value< charset /expiration> </validation> =UTF-8	
Error-Response				x-ca-err: { "error":"invalid_request", "error_description":"500210 Missing or unknown parameters" } Pragma : no-cache Cache-Control : no-store status: 400 content-type: application/json; charset =UTF-8	If any required parameters or headers are missing, the request will fail.
Error-Response				x-ca-err: { "error":"invalid_request", "error_description":"SSL 500220 with client authentication is required" } Pragma : no-cache Cache-Control : no-store status: 403 content-type: application/json; charset =UTF-8	Authentication failed.
Error-Response				x-ca-err: { "error":"invalid_request", "error_description":"The 500220 client certificate is not valid" } 	The client could not be authenticated.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					<pre>Pragma : no- cache Cache- Control : no- store status: 401 content -type:  applica tion /json; charset =UTF-8</pre>	
	Error-Response				<pre>x-ca- { "error":"invalid_request", err: "error_description":" 500299 Validation error" } 0</pre>	The given token is not valid. The error code may be 1003990, 1003991, 1003992, 1003993
	Error-Response				<pre>Pragma : no- cache Cache- Control : no- store status: 401 content -type:  applica tion /json; charset =UTF-8</pre>	
	Error-Response				<pre>x-ca- { "error":"invalid_scope", err: "error_description":"No 500211 registered scope value for this client has been Pragma requested" } : no- cache Cache- Control : no- store status: 400</pre>	invalid scope. None of the given values is valid for the refresh_token

ID Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params (for GET attach params to URL-PATH)	Comment
				content -type: application /json; charset =UTF-8	
Error-Response				x-ca-err: { "error":"unauthorized_client", 500211 "error_description":"The 7 client is not authorized to Pragma use the token" } : no- cache Cache- Control : no- store status: 400 content -type: application /json; charset =UTF-8	The client is not authorized to use the token.
Error-Response				Allow: GET, POST Pragma : no- cache Cache- Control : no- store status: 405	The HTTP method is not valid <b>Allow</b> : This header is required by <a href="http://tools.ietf.org/html/rfc2616">RFC 2616 (http://tools.ietf.org/html/rfc2616)</a> and contains a comma separated list of valid HTTP methods.
ca ti o n	Revokes an oauth_token or refresh_token or access_token.	/oau	POST /vali dati on	content token=<token>& -th -type: application/x-www-form-urlencoded token_type_hint=<token_ty pe_hint>& client_id=<client_id>	Authentication is done via ssl mutual authentication. <b>token</b> : the token to be deleted. <b>token_type_hint</b> : either 'access_token' or 'refresh_token'. Use

ID Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
	/v2 /rev ocati on				'access_token' if the token is a 'oauth_token' (OAuth 1.0).client_id : the client_id of the requesting client.
Response				status: {"result":"revoked"} 200 content -type: application /json; charset =UTF-8	
Error-Response				x-ca-err: { "error":"invalid_request", "error_description":"500310 Missing or unknown parameters" } 500310 Missing or unknown parameters Pragma : no- cache Cache- Control : no- store status: 400 content -type: application /json; charset =UTF-8	If any required parameters or headers are missing, the request will fail.
Error-Response				x-ca-err: { "error":"invalid_request", "error_description":"SSL 500320 with client authentication is required" } 500320 with client authentication is required Pragma : no- cache Cache- Control : no- store status: 403 content	Authentication failed.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					-type: application/json; charset=UTF-8	
	Error-Response			x-ca-err:	{ "error":"invalid_request", "error_description":"The client could not be authenticated." }  500320  Pragma: no-cache Cache-Control: no-store status: 401 content-type: application/json; charset=UTF-8	The client could not be authenticated.
	Error-Response			x-ca-err:	{ "error":"unauthorized_client", "error_description":"The client is not authorized to use the token." }  500311  Pragma: no-cache Cache-Control: no-store status: 400 content-type: application/json; charset=UTF-8	The client is not authorized to use the token.
	Error-Response					

ID Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				Allow: POST Pragma : no- cache Cache- Control : no- store status: 405	The HTTP method is not valid <b>Allow</b> : This header is required by <a href="http://tools.ietf.org/html/rfc2616">RFC 2616 (http://tools.ietf.org/html/rfc2616)</a> and contains a comma separated list of valid HTTP methods.
Error-Response		x-ca- err: 500311 8 Pragma : no- cache Cache- Control : no- store status: 503 content -type: applica tion /json; charset =UTF-8	{ "error": " unsupported_token_type", "error_description": "the given type of token is not supported" } Pragma : no- cache Cache- Control : no- store status: 503 content -type: applica tion /json; charset =UTF-8		unsupported token type
validate_id_t ok e n	Validates an id_token that is represented as JWT.	/oau th /vali dati on /vali date /v2 /idto ken	POST -type: applica tion/x- www- form- urlenco ded	content action=<action>& id_token=<id_token>& check_expiration=true	Authentication is done via ssl mutual authentication. <b>action</b> : MUST be 'validate'. <b>id_token</b> : the id_token to be validated represented as <b>JWT.check_expiration</b> (optional) : a flag that is used to determine if the expiration of the id_token should be checked. Default is 'true'
Response					the id_token does not exist.

ID Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				<pre> status: &lt;validation xmlns="http://ns 200   .i7tech.com/2012/11/otk- content ovp"&gt; &lt;result&gt;none&lt; -type: /result&gt; &lt;/validation&gt; text /xml </pre>	
Response				<pre> status: &lt;validation xmlns="http://ns the id_token is valid. 200   .i7tech.com/2012/11/otk- content ovp"&gt; &lt;result&gt;valid&lt; -type: /result&gt; text   &lt;resource_owner&gt;value&lt; /xml    /resource_owner&gt; &lt;sub&gt;value&lt;/sub&gt; &lt;aud&gt;value&lt;/aud&gt; &lt;expiration&gt;value&lt; /expiration&gt; &lt;azp&gt;value&lt; /azp&gt; &lt;/validation&gt; </pre>	
Response				<pre> status: &lt;validation xmlns="http://ns [MAG] only: the response 200   .i7tech.com/2012/11/otk- content ovp"&gt; &lt;result&gt;valid&lt; -type: /result&gt; text   &lt;resource_owner&gt;value&lt; /xml    /resource_owner&gt; &lt;sub&gt;value&lt;/sub&gt; &lt;aud&gt;value&lt;/aud&gt; &lt;expiration&gt;value&lt; /expiration&gt; &lt;azp&gt;value&lt; /azp&gt; &lt;third_party_sso_token&gt;val ue&lt; /third_party_sso_token&gt; &lt;third_party_sso_token_typ e&gt;value&lt; /third_party_sso_token_typ e&gt; &lt;/validation&gt; </pre>	[MAG] only: the response appears as follows if it has been used with MAG (CA Mobile API Gateway).
Error-Response				<pre> x-ca- { "error":"invalid_request", err: "error_description":"" 500410 Missing or unknown 3     parameters" } Pragma : no- cache Cache- Control : no- store status: 400 </pre>	If any required parameters or headers are missing, the request will fail.

ID Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				content -type: application /json; charset =UTF-8	
Error-Response				x-ca-err: { "error":"invalid_request", "error_description":"SSL 500420 with client authentication is 4 required" } Pragma : no- cache Cache- Control : no- store status: 403 content -type: application /json; charset =UTF-8	Authentication failed.
Error-Response				x-ca-err: { "error":"invalid_request", "error_description":"The client could not be authenticated. 500420 client certificate is not 5 valid" } Pragma : no- cache Cache- Control : no- store status: 401 content -type: application /json; charset =UTF-8	The client could not be authenticated.
Error-Response					The JWT is not valid.

ID Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				x-ca-err: 500412 1 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	{ "error":"invalid_request", "error_description":"The given JWT is invalid" }
Error-Response				x-ca-err: 500411 7 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	{ "error":"unauthorized_client", "error_description":"The client is not authorized to execute a status check" }
Error-Response				x-ca-err: 500412 0 Pragma : no- cache Cache- Control	{ "error":"invalid_request", "error_description":"The id_token has missing claims or has expired" }

ID Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params (for GET attach params to URL-PATH)	Comment
				: no-store status: 400 content-type: application/json; charset =UTF-8	
Error-Response				Allow: POST Pragma : no-cache Cache-Control : no-store status: 405	The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616 (https://tools.ietf.org/html/rfc2616)</a> and contains a comma separated list of valid HTTP methods.
<b>ok</b> Creates an id_token represented as JWT.	/oau /vali /dati /on /vali /date /v2 /idto ken	POST		content action=<action>&-type: applica tion/x-www-form-urlencoded owner>&azp=<azp>&nonce=<nonce>&salt=<salt>&third_party_sso_token=<third_party_sso_token>&third_party_sso_token_type=<third_party_sso_token_type>	Authentication is done via ssl mutual authentication. <b>action</b> : must be 'create'. <b>client_key</b> : the client_id of the requesting client. <b>access_token</b> (optional) : the access_token issued to the client when this id_token was requested. <b>code</b> (optional) : the authorization_code issued to the client when this id_token was requested. <b>resource_owner</b> (optional) : the resource_owner represented by this id_token. <b>azp</b> (optional) : a space separated list of valid users, e.g.: multiple client_ids. <b>nonce</b> (optional) : a nonce which is optional, but required for implicit

ID Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					OAuth 2.0 flows. Therefore, if 'access_token' is provided, it is required. <b>salt</b> (optional) : a value to generate the 'sub' value. For a given 'client_key - resource_owner' combination, that value has to be deterministic. <b>third_party_sso_token</b> (optional) : [MAG]: used with CA Mobile API Gateway only. In MAG this is a SiteMinder SSO token by default. <b>third_party_sso_token_type</b> (optional) : [MAG]: used with CA Mobile API Gateway only. In MAG this is 'urn:ca: params:oauth:grant-type: jwt-bearer-smssotoken' by default.
Response				status: <validation xmlns=" <a href="http://ns.l7tech.com/2012/11/otk">http://ns.l7tech.com/2012/11/otk</a> "> 200 <result>valid</result> content ovp"> <result>valid</result> -type: /result> <jwt>value</jwt> text <id_token_type>value</id_token_type> /xml </id_token_type> </validation>	the id_token does not exist.
Error-Response				x-ca-err: { "error":"invalid_request", "error_description":"500410 Missing or unknown parameters" } Pragma: no-cache Cache-Control: no-store status: 400 content-type: application/json	If any required parameters or headers are missing, the request will fail.

ID Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error-Response	x-ca-err: 500420	{ "error":"invalid_request", "error_description":"SSL with client authentication is required" }	Authentication failed.	Pragma: no-cache	
Error-Response	x-ca-err: 500420	{ "error":"invalid_request", "error_description":"The client certificate is not valid" }	The client could not be authenticated.	Pragma: no-cache	
Error-Response	x-ca-err: 500412	{ "error":"invalid_request", "error_description":"The JWT could not be created" }	The JWT could not be created.	2	

ID Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				<pre>Pragma : no- cache Cache- Control : no- store status: 400 content -type:  applica tion /json; charset =UTF-8</pre>	
Error-Response				<pre>Allow: POST Pragma : no- cache Cache- Control : no- store status: 405</pre>	<p>The HTTP method is not valid</p> <p><b>Allow</b> : This header is required by <a href="http://tools.ietf.org/html/rfc2616">RFC 2616 (http://tools.ietf.org/html/rfc2616)</a> and contains a comma separated list of valid HTTP methods.</p>
Error-Response				<pre>x-ca- err: 500400 0 Pragma : no- cache Cache- Control : no- store status: 500</pre>	Unknown error.

## Associated Tasks

- [Authorize a request\\_token \(see page 208\)](#)
- [Validate Grant Types \(see page 208\)](#)
- [Validate a Refresh Token \(see page 209\)](#)
- [Validate a refresh\\_token \(see page 209\)](#)
- [Validate OAuth Parameters and Signature \(see page 210\)](#)

## Authorize a request\_token

Authorizes a request\_token, making it available in exchange for an access\_token.

```
/oauth/validation/v1/authorize?token=<value>&expiration=<value>&verifier=<value>
```

**token:** the temporary token to be authorized

**expiration:** the new expiration date that will be used if the token is valid

**verifier:** the verifier used if the token is valid

Validation requirements:

- the expiration date has not expired
- a *resource\_owner* must be assigned to the token
- a callback must be assigned to the token

Response:

- status: 200, content-type: text/xml, body
- status: 401, content-type: text/xml, body

## Validate Grant Types

Only auth\_code is validated here.

```
/oauth/validation/validate/v2/granttype?callback=<value>&client_key= <value>&token=<value>
```

**callback:** the *redirect\_uri* passed in by the client

**client\_key:** the associated *client\_key*

**token:** the temporary token (*authorization\_code*)

Validation requirements:

- *client\_key* must match the one that was used when the token was generated
- the expiration date must not be expired
- a callback must either be empty or equal to the one used when the token was generated

Response:

- status: 200, content-type: text/xml, body
- status: 401, content-type: text/xml, body

## Validate a Refresh Token

---

`/oauth/validation/validate/v2/granttype?callback=<value>&client_key= <value>&token=<value>`

**callback:** the *redirect\_uri* passed in by the client

**client\_key:** the associated *client\_key*

**token:** the temporary token (*authorization\_code*)

---

The validation includes the validation steps:

- *client\_key* must match the one that was used when the token was generated
- the expiration date must not be expired
- a callback must either be empty or equal to the one used when the token was generated

Response:

- status: 200, content-type: text/xml, body
- status: 401, content-type: text/xml, body

## Validate a refresh\_token

---

`/oauth/validation/validate/v2/refreshToken?client_key=<value>&rtoken= <value>&scope=<value>`

**client\_key:** the *client\_key* issued for this token

**rtoken:** the *refresh\_token* to be validated

**scope:** the scope to be used

---

Validation requirements:

- *client\_key* must match the one that was used when the token was generated
- the expiration date must not be expired
- the status must be ENABLED

Response:

- status: 200, content-type: text/xml, body
- status: 401, content-type: text/xml, body

---

`/oauth/validation/validate/v2/tokenrequest?client_key=<value>&secret= <value>`

Used to validate a the client credentials when requesting an access\_token

---

**client\_key:** the *client\_key*

---

`/oauth/validation/validate/v2/tokenrequest?client_key=<value>&secret= <value>`

**secret:** secret for the `client_key`

---

Validation requirements:

- the secret
- the expiration date must not be expired
- the status must be ENABLED

Response:

- status: 200, content-type: text/xml, body
- status: 401, content-type: text/xml, body

## Validate OAuth Parameters and Signature

The OAuth Validation Point (OVP) is used when clients access resources. Validated tokens and signatures are cached to improve performance. Ensure the default values for caching conform to the security policy at your organization.

APIs	Notes
<code>/oauth/validation/validate/v2/token</code>	Used to validate "oauth" parameters.  Validation depends on the token type. If token type is "MAC" or "BEARER", the expiration date and the status will be verified.
	The default <code>cacheAge</code> context variable is 60 seconds. A revoked token continues to be authorized for up to 60 seconds beyond revocation.
<code>/oauth/validation/validate/v1/signature</code>	Used to validate "oauth_signature"  Validation requirements: <ul style="list-style-type: none"> <li>▪ The signature is verified</li> <li>▪ The client_key is verified</li> <li>▪ expiration date has not expired</li> <li>▪ status is ENABLED</li> <li>▪ The token is verified</li> <li>▪ Expiration date has not expired</li> <li>▪ status is ENABLED (for access_tokens only)</li> <li>▪ For authorized request token, the verifier must exist in the tokenstore</li> </ul> The default <code>cacheAge</code> context variable is 30 seconds. An accepted signature is cached for 30 seconds.

---

## OAuth Protected APIs

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/Query Params	Notes
us_01	This API reflects an implementation according to OpenID Connect. For more information refer to this website: <a href="http://openid.net/specs/openid-connect-core-1_0.html#UserInfo">http://openid.net/specs/openid-connect-core-1_0.html#UserInfo</a>	/openid/nid/connect/v1/use/info	GET	authorizat-ion: /content-type: application/x-www-form-urlencoded	ation: params to URL-PATH	<p>This endpoint returns a message with user details. The content depends on the SCOPE associated with the given access_token.</p> <p><b>IMPORTANT:</b> The API will fail if not active id_token is available for the associated user.</p> <p><b>authorization :</b> A valid access_token (e.g.: authorization: Bearer access-token-value)<b>authorization :</b> The access_token has to be SCOPE'd for at least 'openid'. Other valid SCOPE values are: 'email', 'address', 'phone', 'profile'. In addition to those SCOPE values 'user_role' is also supported. That SCOPE is an extension for OTK/MAG and returns the role of the user</p>
Response					status: { "sub": "248289761001", "name": "Jane Doe", "given_name": "Jane", "family_name": "Doe", "preferred_username": "j.doe", "email": "janedoe@example.com", "picture": "http://example.com/janedoe/me.jpg" }	A JSON message containing details about the user. The content depends on the SCOPE associated with the access_token.
Error-Response		x-ca-err: 990	Pragma :no-store	cache	{ "error": "invalid_request", "error_description": "Validation error" }	The given token is not valid. The error code may be 990, 991, 992, 993

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/Query Params	Notes
				status: 401 content -type: application /json; charset =UTF-8		
	Error-Response			x-ca-err: 203 : no- cache Cache- Control : no- store status: 403 content -type: application /json; charset =UTF-8	{ "error":"invalid_request", "error_description" "Pragma :\"SSL is required\""} Forbidden.	
	Error-Response			Allow: GET, POST Pragma : no- cache Cache- Control : no- store status: 405	The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616 (https://tools.ietf.org/html/rfc2616)</a> and will contain a comma separated list of valid http methods.	
<b>res</b> <b>ou</b> <b>rce</b> <b>_o</b> <b>wn</b> <b>er_</b> <b>ses</b> <b>sio</b>	The client requests the session status by passing in the id_token of the authenticated user. /con nect /sess ion /stat us	/con nect /sess ion /stat us	GET	authori zation: id_token=<id_toke n>& content id_token_type=urn: -type: ietf:params:oauth: application/x- bearer	This informs the client if the resource owner is logged in and has a valid session. The session may be active or not existing. grant-type:jwt- bearer	<b>authorization</b> : A valid access_token (e.g.: authorization: Bearer access-token-value) <b>id_token</b> : The id_token of the

ID	Operation	URL- Path	HTTP Method	HTTP Body/Query Params	Notes
n_status			www-form-urlencoded		resource ownerid_token_type(optional) : The id_token_type to be invalidated. Required if not default type.
	Response			status: {"session":"active"} 200 content-type: application/json; charset=UTF-8	Session is active
	Response			status: {"session_status":"none"} 200 content-type: application/json; charset=UTF-8	Session does not exist
	Error-Response			x-ca-err: 300710 err: "invalid_request", 3 "error_description" :"Missing or Pragma duplicate : no- cache Cache- Control : no- store status: 400 content-type: application/json; charset=UTF-8	If any required parameters or headers are missing, the request will fail.
	Error-Response			x-ca-err: 300799 err: "invalid_request", 3 "error_description" 0 "Validation error" Pragma } : no- cache	The given token is not valid. The error code may be 1003990, 1003991, 1003992, 1003993

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/Query Params	Notes
				Cache- Control : no- store status: 401 content -type: applica tion /json; charset =UTF-8		
	Error-Response		x-ca- err: 300720 3	{ "error": "invalid_request", "error_description" :"SSL is required" } Pragma : no- cache Cache- Control : no- store status: 403 content -type: applica tion /json; charset =UTF-8		Forbidden.
	Error-Response		Allow: GET, POST Pragma : no- cache Cache- Control : no- store status: 405		The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616 (https://tools.ietf.org/html/rfc2616)</a> and contains a comma separated list of valid HTTP methods.	
	Error-Response		x-ca- err: 300700 0		Unknown error.	

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/Query Params	Notes
				Pragma : no- cache Cache- Control : no- store status: 500		

## Clientstore API

This API allows you to manage client applications and client keys (client\_id, oauth\_consumer\_key).

All endpoints described start here:

*<Gateway\_host\_and\_port>/oauth/clientstore/\**

- [get\\_all\\_client \(see page 215\)](#)
- [get\\_client\\_by\\_name\\_org \(see page 218\)](#)
- [get\\_client\\_org \(see page 222\)](#)
- [get\\_client\\_registered\\_by \(see page 224\)](#)
- [get\\_client\\_by\\_clientkey \(see page 227\)](#)
- [get\\_client\\_by\\_ident \(see page 230\)](#)
- [get\\_all\\_client\\_id \(see page 233\)](#)
- [get\\_client\\_id \(see page 236\)](#)
- [get\\_client\\_id\\_name \(see page 239\)](#)
- [get\\_client\\_id\\_org \(see page 242\)](#)
- [get\\_client\\_id\\_ident \(see page 245\)](#)
- [get\\_client\\_id\\_filter \(see page 248\)](#)
- [get\\_client\\_client\\_id\\_values \(see page 252\)](#)
- [persist\\_client \(see page 255\)](#)
- [persist\\_client\\_id \(see page 259\)](#)
- [persist\\_client\\_and\\_client\\_id \(see page 264\)](#)
- [delete\\_client \(see page 269\)](#)
- [revoke\\_client\\_id \(see page 271\)](#)
- [update\\_client \(see page 274\)](#)
- [update\\_client\\_id \(see page 277\)](#)
- [update\\_client\\_id\\_registered\\_by \(see page 281\)](#)

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params	Comment
					(for GET attach params to URL-PATH)	

GET

ID	Operation	URL- Path	HTTP Method	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
int	<b>Get the values of all clients</b>	/oau/th/clie/ntst/or/ore/get		content offset=0&-type: format=xmlapplication/xml; www-form-urlencoded	Authentication is done via ssl mutual authentication.  <b>offset</b> (optional) : The API returns up to 100 values. This parameter is the query offset which can be used for pagination purposes. <b>format</b> (optional) : Either 'xml' or 'json'. The success response is returned in the requested format. Errors are always returned in JSON.
	Response			status: <values xmlns="http://ns.l7tech.com/2012/11/otk-clientstore"> <value index="0" content integer"> <client_ident>value<-type: /client_ident> <name>value</name> <text>value</text> <description>value</description> <organization>value</organization> <registered_by>value<=UTF-8 /registered_by> <created>value</created> <client_custom>URL-Encoded-JSON-structure</client_custom> </value> </values>	An XML response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found). 'index' is a temporary list number. No value is tied to the index value. It is valid for this response only.
	Response			status: { "values": { "value": [ { "organization": "value", "index": "value", "created": "value", "description": "value", "name": "value", "type": "value", "client_ident": "value", "registered_by": "value", "client_custom": "{URL-Encoded-JSON-Structure}" } ] } }	A JSON response that includes the key 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
	Error-Response			x-ca-err: 400010 err: "error_description":"Missing or unknown parameters" 3 Pragma: no-cache	If any required parameters or headers are missing, the request will fail.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					Cache-Control : no-store status: 400 content-type: application/json; charset=UTF-8	
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"SSL with client authentication is required" }	Authentication failed.
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The client certificate is not valid" }	The client could not be authenticated.

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params (for GET attach params to URL-PATH)	Comment
					/json; charset =UTF-8	
Error-Response				x-ca- err: 400030 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: text /xml; charset =UTF-8	{ "error":"invalid_request", "error_description":"The path component 400030 of the URL is invalid" }  The path component is invalid.	
Error-Response				Allow: GET Pragma : no- cache Cache- Control : no- store status: 405	The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.	
Error-Response				x-ca- err: 400000 0 Pragma : no- cache Cache- Control : no- store status: 500	Unknown error.	
<b>Get the values of the client by /clie</b>		/oau	GET		content name=<name>& -type: applica	Authentication is done via ssl mutual authentication.

ID	Operation	URL- Path	HTTP Method	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
<b>the name and optionally the organization</b>	ntst	tion/x-www-form-urlencoded	/get	org=<org>&offset=0&urlenco ded	<b>name</b> : the client name <b>org</b> (optional) : the org(organization) <b>offset</b> (optional) : The API returns up to 100 values. This parameter is the query offset which can be used for pagination purposes. <b>format</b> (optional) : Either 'xml' or 'json'. The response is returned in the requested format.
Response				<pre>status: &lt;values xmlns="http://ns.l7tech.com/2012/11/otk-clientstore"&gt; &lt;value index="0" type="client_ident"&gt; &lt;name&gt;value&lt;/name&gt; &lt;text&gt;value&lt;/text&gt; &lt;description&gt;value&lt;/description&gt; &lt;organization&gt;value&lt;/organization&gt; &lt;registered_by&gt;value&lt;content type="UTF-8"/&gt; &lt;created&gt;value&lt;/created&gt; &lt;client_custom&gt;URL-Encoded-JSON-structure&lt;/client_custom&gt; &lt;/value&gt; &lt;/values&gt;</pre>	An XML response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found). 'index' is a temporary list number. No value is tied to the index value. It is valid for this response only.
Response				<pre>status: { "values": { "value": [ { "organization": "value", "index": "value", "created": "value", "description": "value", "name": "value", "type": "value", "client_ident": "value", "registered_by": "value", "client_custom": "{URL-Encoded-JSON-Structure}" } ] } }</pre>	A JSON response that includes the key 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
Error-Response		x-ca-err-3		<pre>{ "error": "invalid_request", "error_description": "Missing or unknown parameters" }</pre>	If any required parameters or headers are missing, the request will fail.
		Pragma			
		: no-cache			
		Cache-Control			

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					: no-store status: 400 content-type: application/json; charset=UTF-8	
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"SSL with client authentication is required" } 4 Pragma : no-cache Cache-Control : no-store status: 403 content-type: application/json; charset=UTF-8	Authentication failed.
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The client certificate is not valid" } 5 Pragma : no-cache Cache-Control : no-store status: 401 content-type: application/json;	The client could not be authenticated.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					tion /json; charset =UTF-8	
Error-Response		x-ca- err: 400030	3	{ "error":"invalid_request", "error_description":"The path component 400030 of the URL is invalid" }		The path component is invalid.
Error-Response				Pragma : no- cache Cache- Control : no- store status: 400 content -type: text /xml; charset =UTF-8		
Error-Response				Allow: GET		The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616 (https://tools. ietf.org/html/rfc2616)</a> and contains a comma separated list of valid HTTP methods.
Error-Response		x-ca- err: 400000	0	Pragma : no- cache Cache- Control : no- store status: 500		Unknown error.
	GET			org=<org>&		

ID	Operation	URL- Path	HTTP Method	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
	<b>Get the values of a client by organization on</b>	/oau/th/clients/ <b>org</b> /get		content-type: application/xml offset=0&format=xml	Authentication is done via ssl mutual authentication.  <b>org</b> : the org (anization) <b>offset</b> (optional) : The API returns up to 100 values. This parameter is the query offset which can be used for pagination purposes. <b>format</b> (optional) : Either 'xml' or 'json'. The response is returned in the requested format.
	Response			status: <values xmlns="http://ns.l7tech.com/2012/11/otk-clientstore"> <value index="0" content="integer"> <client_ident>value</client_ident> <name>value</name> <text>value</text> <type>value</type> <description>value</description> <organization>value</organization> <registered_by>value</registered_by> <created>value</created> <client_custom>URL-Encoded-JSON-structure</client_custom> </value> </values>	An XML response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found). 'index' is a temporary list number. No value is tied to the index value. It is valid for this response only.
	Response			status: { "values": { "value": [ { "organization": "value", "index": "value", "created": "value", "description": "value", "name": "value", "type": "value", "client_ident": "value", "registered_by": "value", "client_custom": "{URL-Encoded-JSON-Structure}" } ] } }	A JSON response that includes the key 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
Error- Response		x-ca-err-3-Pragma		{ "error": "invalid_request", "error_description": "Missing or unknown parameters" }	If any required parameters or headers are missing, the request will fail.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					Cache-Control : no-store status: 400 content-type: application/json; charset=UTF-8	
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"SSL with client authentication is required" }	Authentication failed.
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The client certificate is not valid" }	The client could not be authenticated.

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params (for GET attach params to URL-PATH)	Comment
					/json; charset =UTF-8	
Error-Response			x-ca- err: 400030	{ "error":"invalid_request", "error_description":"The path component 400030 of the URL is invalid" }	3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: text /xml; charset =UTF-8	The path component is invalid.
Error-Response			Allow: GET Pragma : no- cache Cache- Control : no- store status: 405			The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
Error-Response			x-ca- err: 400000	0	Unknown error.	
				Pragma : no- cache Cache- Control : no- store status: 500		
<b>Get the values of a client by /client</b>		/oau	GET	content	registered_by=<registered_by>&-type: application	Authentication is done via ssl mutual authentication.

ID	Operation	URL- Path	HTTP Method	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
	<b>whom it was registered and optionally the organization</b>	ntst ore /get	tion/x- www- form- urlenco ded	org=<org>& offset=0& format=xml	<b>registered_by</b> : whoever has registered the client <b>org</b> (optional) : the org(ization) <b>offset</b> (optional) : The API returns up to 100 values. This parameter is the query offset which can be used for pagination purposes. <b>format</b> (optional) : Either 'xml' or 'json'. The response is returned in the requested format.
	Response			status: <values xmlns="http://ns.l7tech.com/2012/11/otk-clientstore"> <value index="0" content="integer"> <client_ident>value<-type> /<client_ident> <name>value</name> <text>value</text> <description>value</description> <organization>value</organization> <registered_by>value<=UTF-8 /<registered_by> <created>value</created> <client_custom>URL-Encoded-JSON-structure</client_custom> </value> </values>	An XML response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found). 'index' is a temporary list number. No value is tied to the index value. It is valid for this response only.
	Response			status: { "values": { "value": [ { "organization": "value", "index": "value", "created": "value", "description": "value", "name": "value", "type": "value", "client_ident": "value", "registered_by": "value", "client_custom": "{URL-Encoded-JSON-Structure}" } ] } }	A JSON response that includes the key 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
	Error- Response			x-ca- err: 400010 3 Pragma : no- cache	If any required parameters or headers are missing, the request will fail.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					Cache-Control : no-store status: 400 content-type: application/json; charset=UTF-8	
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"SSL with client authentication is required" }	Authentication failed.
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The client certificate is not valid" }	The client could not be authenticated.

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params (for GET attach params to URL-PATH)	Comment
					/json; charset =UTF-8	
Error- Response			x-ca- err: 400030 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: text /xml; charset =UTF-8	{ "error":"invalid_request", "error_description":"The path component 400030 of the URL is invalid" }  3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: text /xml; charset =UTF-8		The path component is invalid.
Error- Response			Allow: GET Pragma : no- cache Cache- Control : no- store status: 405			The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools. ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
Error- Response			x-ca- err: 400000 0 Pragma : no- cache Cache- Control : no- store status: 500			Unknown error.
by	/oau th /clie	GET		content client_key=<client_key>& -type: format=xml applica		Authentication is done via ssl mutual authentication.

ID	Operation	URL- Path	HTTP Method	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
<b>Get the values of a client by a client_key</b>	ntst ore /get a client_key			tion/x-www-form-urlencoded	<b>client_key</b> : The client_id of the client is returned. In the context of OAuth 1.0 it is the oauth_consumer_key. <b>format</b> (optional) : Either 'xml' or 'json'. The response is returned in the requested format.
Response				status: <values xmlns="http://ns.l7tech.com/2012/11/otk-clientstore"> <value index="200" type="client_ident"> <name>value</name> <text>value</text> <description>value</description> <organization>value</organization> <registered_by>value<=UTF-8</registered_by> <created>value</created> <client_custom>URL-Encoded-JSON-structure</client_custom> </value></values>	An XML response that includes the element 'values'. 'value' will appear once or not at all. 'index' is a temporary list number. No value is tied to the index value. It is valid for this response only.
Response				status: { "values": { "value": [ { "organization": "200", "value", "index": "value", "created": "content", "description": "value", "name": "value", "type": "value", "client_ident": "application", "registered_by": "value", "client_custom": "{URL-Encoded-JSON-structure}" } ] } } charset =UTF-8	A JSON response that includes the key 'values'. 'value' will appear once or not at all.
Error- Response		x-ca- err: 400010 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion		{ "error": "invalid_request", "error_description": "Missing or unknown parameters" }	If any required parameters or headers are missing, the request will fail.

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error- Response					/json; charset =UTF-8	
Error- Response					x-ca- { "error":"invalid_request", err: "error_description":"SSL with client 400020 authentication is required" } 4 Pragma : no- cache Cache- Control : no- store status: 403 content -type: applica tion /json; charset =UTF-8	Authentication failed.
Error- Response					x-ca- { "error":"invalid_request", err: "error_description":"The client certificate 400020 is not valid" } 5 Pragma : no- cache Cache- Control : no- store status: 401 content -type: applica tion /json; charset =UTF-8	The client could not be authenticated.
Error- Response					x-ca- { "error":"invalid_request", err: "error_description":"The path component 400030 of the URL is invalid" } 3 Pragma : no-	The path component is invalid.

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params	Comment
					(for GET attach params to URL-PATH)	
					cache Cache- Control : no- store status: 400 content -type: text /xml; charset =UTF-8	
	Error- Response			Allow: GET Pragma : no- cache Cache- Control : no- store status: 405		The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
	Error- Response			x-ca- err: 400000 0 Pragma : no- cache Cache- Control : no- store status: 500		Unknown error.
	<b>Get the values of a client by its client_id by</b>	/oau	GET		content client_ident=<client_ident>& -type: format=xml aplica tion/x- www- form- urlenco ded	Authentication is done via ssl mutual authentication.  <b>client_ident</b> : Client values selected by <b>client_id</b> <b>format</b> (optional) : Either

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
						'xml' or 'json'. The response is returned in the requested format.
Response					<pre> status: &lt;values xmlns="http://ns.l7tech.com/2012/11/otk-clientstore"&gt; &lt;value index="content integer"&gt; &lt;client_ident&gt;value&lt;-type: /client_ident&gt; &lt;name&gt;value&lt;/name&gt; &lt;text&gt;value&lt;/text&gt; &lt;description&gt;value&lt;/description&gt; &lt;organization&gt;value&lt;/organization&gt; &lt;registered_by&gt;value&lt;=UTF-8 /registered_by&gt; &lt;created&gt;value&lt;/created&gt; &lt;client_custom&gt;URL-Encoded-JSON-structure&lt;/client_custom&gt; &lt;/value&gt; &lt;/values&gt; </pre>	An XML response that includes the element 'values'. 'value' will appear once or not at all. 'index' is a temporary list number. No value is tied to the index value. It is valid for this response only.
Response					<pre> status: { "values": { "value": [ { "organization": "value", "index": "value", "created": "content", "value", "description": "value", "name": "-type: value", "type": "value", "client_ident": "aplica", "value", "registered_by": "value", "client_custom": "{URL-Encoded-JSON-/json; Structure}" ] } } </pre>	A JSON response that includes the key 'values'. 'value' will appear once or not at all.
Error-Response					<pre> x-ca- { "error": "invalid_request", err: "error_description": "Missing or unknown 400010 parameters" } </pre> <p>3</p> <pre> Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8 </pre>	If any required parameters or headers are missing, the request will fail.
Error-Response					<pre> x-ca- { "error": "invalid_request", err: "error_description": "SSL with client 400020 authentication is required" } </pre> <p>4</p> <pre> Pragma </pre>	Authentication failed.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					: no-cache Cache-Control : no-store status: 403 content-type: application/json; charset =UTF-8	
Error-Response				x-ca-err:	{ "error": "invalid_request", "error_description": "The client certificate 400020 is not valid" }	The client could not be authenticated.
Error-Response				x-ca-err:	{ "error": "invalid_request", "error_description": "The path component 400030 of the URL is invalid" }	The path component is invalid.

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params (for GET attach params to URL-PATH)	Comment
					text /xml; charset =UTF-8	
	Error- Response			Allow: GET Pragma : no- cache Cache- Control : no- store status: 405		The HTTP method is not valid. <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
	Error- Response			x-ca- err: 400000 0 Pragma : no- cache Cache- Control : no- store status: 500		Unknown error.
get_all_client_id	<b>Get the values of all client_ids</b>	/oau th all client_ids	GET /clie ntst ore /get Key		content offset=0& -type: format=xml aplica tion/x- www- form- urleco ded	Authentication is done via ssl mutual authentication.  <b>offset</b> (optional) : The API returns up to 100 values. This parameter is the query offset which can be used for pagination purposes. <b>format</b> (optional) : Either 'xml' or 'json'. The response is returned in the requested format.
	Response				status: <values xmlns="http://ns.l7tech.com/2012/11/otk-clientstore"> <value index="content integer"> <client_id>value<	An XML response that includes the element 'values'.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					<pre>-type: /client_ident&gt; &lt;client_name&gt;value&lt; text /client_name&gt; &lt;client_key&gt;value&lt; /xml; /client_key&gt; &lt;secret&gt;value&lt;/secret&gt; charset &lt;scope&gt;value&lt;/scope&gt; &lt;callback&gt;value&lt; =UTF-8 /callback&gt; &lt;environment&gt;value&lt; /environment&gt; &lt;expiration&gt;value&lt; /expiration&gt; &lt;status&gt;value&lt;/status&gt; &lt;created&gt;value&lt;/created&gt; &lt;created_by&gt;value&lt;/created_by&gt; &lt;client_key_custom&gt;URL-Encoded-JSON- structure&lt;/client_key_custom&gt; &lt;serviceIds&gt;value&lt;/serviceIds&gt; &lt;accountPlanMappingIds&gt;value&lt; /accountPlanMappingIds&gt; &lt;/value&gt; &lt; /values&gt;</pre>	'value' appears once, multiple times, or not at all (indicating no values found). 'index' is a temporary list number. No value is tied to the index value. It is valid for this response only.
	Response				<pre>status: { "values": { "value": [ { "client_ident": 200 "value", "client_name": "value", content "client_key": "value", "secret": "value", -type: "scope": "value", "callback": "value", aplica "expiration": "value", "status": "value", tion "created": "value", "created_by": "value", /json; "client_key_custom": "{URL-Encoded-JSON- charset Structure}", "serviceIds": "value", =UTF-8 "accountPlanMappingIds": "value" ] } }</pre>	A JSON response that includes the key 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
Error- Response					<pre>x-ca- { "error": "invalid_request", err: "error_description": "Missing or unknown 400010 parameters" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: aplica tion /json; charset =UTF-8</pre>	If any required parameters or headers are missing, the request will fail.
Error- Response					<pre>x-ca- { "error": "invalid_request", err: "error_description": "SSL with client 400020 authentication is required" } 4 Pragma</pre>	Authentication failed.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					: no-cache Cache-Control : no-store status: 403 content-type: application/json; charset =UTF-8	
Error-Response				x-ca-err:	{ "error": "invalid_request", "error_description": "The client certificate 400020 is not valid" }	The client could not be authenticated.
Error-Response				x-ca-err:	{ "error": "invalid_request", "error_description": "The path component 400030 of the URL is invalid" }	The path component is invalid.

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params (for GET attach params to URL-PATH)	Comment
					text /xml; charset =UTF-8	
	Error- Response			Allow: GET Pragma : no- cache Cache- Control : no- store status: 405		The HTTP method is not valid. <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
	Error- Response			x-ca- err: 400000 0 Pragma : no- cache Cache- Control : no- store status: 500		Unknown error.
get_client_id	Get the values of the client_id by client_id	/oau /client_id	GET		content client_key=<client_key>&-type: format=xml application/x-www-form-urlencoded	Authentication is done via ssl mutual authentication.
						<b>client_key</b> : The OAuth 2.0 client_id. In the context of OAuth 1.0, this is the oauth_consumer_key. <b>format</b> (optional) : Either 'xml' or 'json'. The response is returned in the requested format.
	Response				status: <values xmlns="http://ns.l7tech.com/2012/11/otk-clientstore"> <value index="content integer"> <client_ident>value<-type: /client_ident> <client_name>value<text /client_name> <client_key>value<	An XML response that includes the element 'values'. 'value' will appear once or not at all.

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params (for GET attach params to URL-PATH)	Comment
					/xml; /client_key> <secret>value</secret> charset <scope>value</scope> <callback>value< =UTF-8 /callback> <environment>value< /environment> <expiration>value< /expiration> <status>value</status> <created>value</created> <created_by>value</created_by> <client_key_custom>URL-Encoded-JSON- structure</client_key_custom> <serviceIds>value</serviceIds> <accountPlanMappingIds>value< /accountPlanMappingIds> </value> < /values>	'index' is a temporary list number. No value is tied to the index value. It is valid for this response only.
	Response				status: { "values": { "value": [ { "client_ident": 200 "value", "client_name": "value", content "client_key": "value", "secret": "value", -type: "scope": "value", "callback": "value", aplica "expiration": "value", "status": "value", tion "created": "value", "created_by": "value", /json; "client_key_custom": "URL-Encoded-JSON- charset Structure", "serviceIds": "value", =UTF-8 "accountPlanMappingIds": "value" ] } }	A JSON response that includes the key 'values'. 'value' will appear once or not at all.
	Error- Response				x-ca- { "error": "invalid_request", err: "error_description": "Missing or unknown 400010 parameters" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: aplica tion /json; charset =UTF-8	If any required parameters or headers are missing, the request will fail.
	Error- Response				x-ca- { "error": "invalid_request", err: "error_description": "SSL with client 400020 authentication is required" } 4 Pragma : no- cache	Authentication failed.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					Cache-Control : no-store status: 403 content-type: application/json; charset=UTF-8	
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The client certificate 400020 is not valid" }	The client could not be authenticated.
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The path component 400030 of the URL is invalid" }	The path component is not valid.

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params (for GET attach params to URL-PATH)	Comment
					text /xml; charset =UTF-8	
Error- Response				Allow: GET Pragma : no- cache Cache- Control : no- store status: 405		The HTTP method is not valid. <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
Error- Response				x-ca- err: 400000 0 Pragma : no- cache Cache- Control : no- store status: 500		Unknown error.
<b>Get the values of a client_id by client name</b>		/oau /client/{client_id}	GET		content name=<name>&-type: application/x-www-form-urlencoded	Authentication is done via ssl mutual authentication.
					format=xml	<b>name</b> : the client name <b>org</b> (optional) : optional org (anization) <b>offset</b> (optional) : The API returns up to 100 values. This parameter is the query offset which can be used for pagination purposes.. <b>format</b> (optional) : Either

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
						'xml' or 'json'. The response is returned in the requested format.
Response					<pre> status: &lt;values xmlns="http://ns.l7tech.com/2012/11/otk-clientstore"&gt; &lt;value index="content integer"&gt; &lt;client_ident&gt;value&lt;-type: /client_ident&gt; &lt;client_name&gt;value&lt;text /client_name&gt; &lt;client_key&gt;value&lt;/xml; /client_key&gt; &lt;secret&gt;value&lt;/secret&gt; charset &lt;scope&gt;value&lt;/scope&gt; &lt;callback&gt;value&lt;=UTF-8 /callback&gt; &lt;environment&gt;value&lt;/environment&gt; &lt;expiration&gt;value&lt;/expiration&gt; &lt;status&gt;value&lt;/status&gt; &lt;created&gt;value&lt;/created&gt; &lt;created_by&gt;value&lt;/created_by&gt; &lt;client_key_custom&gt;URL-Encoded-JSON-structure&lt;/client_key_custom&gt; &lt;serviceIds&gt;value&lt;/serviceIds&gt; &lt;accountPlanMappingIds&gt;value&lt;/accountPlanMappingIds&gt; &lt;/value&gt; &lt;/values&gt; </pre>	An XML response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found). 'index' is a temporary list number. No value is tied to the index value. It is valid for this response only.
Response					<pre> status: { "values": { "value": [ { "client_ident": "200" "value", "client_name": "value", content "client_key": "value", "secret": "value", -type: "scope": "value", "callback": "value", applica "expiration": "value", "status": "value", tion "created": "value", "created_by": "value", /json; "client_key_custom": "{URL-Encoded-JSON-charset Structure}", "serviceIds": "value", =UTF-8 "accountPlanMappingIds": "value" ] } } </pre>	A JSON response that includes the key 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
Error-Response					<pre> x-ca- { "error": "invalid_request", err: "error_description": "Missing or unknown 400010 parameters" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica </pre>	If any required parameters or headers are missing, the request will fail.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error- Response					tion /json; charset =UTF-8	
Error- Response					x-ca- { "error":"invalid_request", err: "error_description":"SSL with client 400020 authentication is required" } 4 Pragma : no- cache Cache- Control : no- store status: 403 content -type: aplica tion /json; charset =UTF-8	Authentication failed.
Error- Response					x-ca- { "error":"invalid_request", err: "error_description":"The client certificate 400020 is not valid" } 5 Pragma : no- cache Cache- Control : no- store status: 401 content -type: aplica tion /json; charset =UTF-8	The client could not be authenticated.
Error- Response					x-ca- { "error":"invalid_request", err: "error_description":"The path component 400030 of the URL is invalid" } 3 Pragma	The path component is invalid.

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params	Comment
					(for GET attach params to URL-PATH)	
					: no-cache Cache-Control : no-store status: 400 content-type: text/xml; charset =UTF-8	
	Error-Response				Allow: GET Pragma : no-cache Cache-Control : no-store status: 405	The HTTP method is not valid. <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
	Error-Response				x-ca-err: 400000 0 Pragma : no-cache Cache-Control : no-store status: 500	Unknown error.
<b>Get the values of a client_id by organization</b>		/oau /client_id /get	GET		content org=<org>&-type: applica tion/x- www-form-urlencoded	Authentication is done via ssl mutual authentication.  <b>org</b> : org(ization) <b>offset</b> (optional) : The API returns up to 100 values. This parameter is the query offset which can be used for

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
						pagination purposes. <b>format</b> (optional) : Either 'xml' or 'json'. The response is returned in the requested format.
Response					<pre> status: &lt;values xmlns="http://ns.l7tech.com/2012/11/otk-clientstore"&gt; &lt;value index="content integer"&gt; &lt;client_ident&gt;value&lt;-type: /client_ident&gt; &lt;client_name&gt;value&lt;text /client_name&gt; &lt;client_key&gt;value&lt;/xml; /client_key&gt; &lt;secret&gt;value&lt;/secret&gt; charset &lt;scope&gt;value&lt;/scope&gt; &lt;callback&gt;value&lt;=UTF-8 /callback&gt; &lt;environment&gt;value&lt;/environment&gt; &lt;expiration&gt;value&lt;/expiration&gt; &lt;status&gt;value&lt;/status&gt; &lt;created&gt;value&lt;/created&gt; &lt;created_by&gt;value&lt;/created_by&gt; &lt;client_key_custom&gt;URL-Encoded-JSON-structure&lt;/client_key_custom&gt; &lt;servicelds&gt;value&lt;/servicelds&gt; &lt;accountPlanMappingIds&gt;value&lt;/accountPlanMappingIds&gt; &lt;/value&gt; &lt;/values&gt; </pre>	An XML response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found). 'index' is a temporary list number. No value is tied to the index value. It is valid for this response only.
Response					<pre> status: { "values": { "value": [ { "client_ident": "200" "value", "client_name": "value", content "client_key": "value", "secret": "value", "-type: "scope": "value", "callback": "value", applica "expiration": "value", "status": "value", tion "created": "value", "created_by": "value", /json; "client_key_custom": "URL-Encoded-JSON-charset Structure]", "servicelds": "value", =UTF-8 "accountPlanMappingIds": "value" ] } } </pre>	A JSON response that includes the key 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
Error-Response					<pre> x-ca- { "error": "invalid_request", err: "error_description": "Missing or unknown 400010 parameters" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: </pre>	If any required parameters or headers are missing, the request will fail.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					application/json; charset=UTF-8	
Error-Response			x-ca-err:	{ "error": "invalid_request", "error_description": "SSL with client authentication is required" }	4	Authentication failed.
Error-Response			x-ca-err:	{ "error": "invalid_request", "error_description": "The client certificate is not valid" }	5	The client could not be authenticated.
Error-Response			x-ca-err:	{ "error": "invalid_request", "error_description": "The path component of the URL is invalid" }	3	The path component is invalid.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					Pragma : no- cache Cache- Control : no- store status: 400 content -type: text /xml; charset =UTF-8	
	Error- Response				Allow: GET Pragma : no- cache Cache- Control : no- store status: 405	The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
	Error- Response				x-ca- err: 400000 0 Pragma : no- cache Cache- Control : no- store status: 500	Unknown error.
<b>id</b>	<b>Get the values of a client_id by client_id</b>	/oau /clie /clie /get	GET		content-type: application/x-www-form-urlencoded	Authentication is done via ssl mutual authentication.
					client_ident=<client_ident>&offset=0&format=xml	<b>client_ident</b> : the client_ident which may result in many associated client_ids <b>offset</b> (optional) :

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
						The API returns up to 100 values. This parameter is the query offset which can be used for pagination purposes. <b>format</b> (optional) : Either 'xml' or 'json'. The response is returned in the requested format.
Response					<pre>status: &lt;values xmlns= "http://ns.l7tech.com/2012/11/otk-clientstore"&gt; &lt;value index="0" content integer"&gt; &lt;client_ident&gt;value&lt;-type: /client_ident&gt; &lt;client_name&gt;value&lt;text: /client_name&gt; &lt;client_key&gt;value&lt;/xml; /client_key&gt; &lt;secret&gt;value&lt;/secret&gt; charset &lt;scope&gt;value&lt;/scope&gt; &lt;callback&gt;value&lt;=UTF-8 /callback&gt; &lt;environment&gt;value&lt;/environment&gt; &lt;expiration&gt;value&lt;/expiration&gt; &lt;status&gt;value&lt;/status&gt; &lt;created&gt;value&lt;/created&gt; &lt;created_by&gt;value&lt;/created_by&gt; &lt;client_key_custom&gt;URL-Encoded-JSON-structure&lt;/client_key_custom&gt; &lt;serviceIds&gt;value&lt;/serviceIds&gt; &lt;accountPlanMappingIds&gt;value&lt;/accountPlanMappingIds&gt; &lt;/value&gt; &lt;/values&gt;</pre>	An XML response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found). 'index' is a temporary list number. No value is tied to the index value. It is valid for this response only.
Response					<pre>status: { "values": { "value": [ { "client_ident": "0", "value", "client_name": "value", "content", "client_key": "value", "secret": "value", "-type: "scope": "value", "callback": "value", "application": "expiration": "value", "status": "value", "creation": "created": "value", "created_by": "value", "JSON": "client_key_custom": "{URL-Encoded-JSON-charset Structure}", "serviceIds": "value", "UTF-8": "accountPlanMappingIds": "value" } ] } }</pre>	A JSON response that includes the key 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
Error-Response					<pre>x-ca-err: { "error": "invalid_request", "error_description": "Missing or unknown parameters" } 3 Pragma: no- Cache-Control: no-</pre>	If any required parameters or headers are missing, the request will fail.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					store status: 400 content -type: application /json; charset =UTF-8	
Error-Response				x-ca- err:	{ "error":"invalid_request", "error_description":"SSL with client 4000020 authentication is required" }	Authentication failed.
Error-Response				x-ca- err:	{ "error":"invalid_request", "error_description":"The client certificate 4000020 is not valid" }	The client could not be authenticated.

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The path component 400030 of the URL is invalid" }	The path component is invalid.
Error-Response				3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: text /xml; charset =UTF-8		
Error-Response				Allow: GET Pragma : no- cache Cache- Control : no- store status: 405	The HTTP method is not valid. <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.	
Error-Response				x-ca-err: 400000 0 Pragma : no- cache Cache- Control : no- store status: 500	Unknown error.	
<b>Get the values of a client_id by using additional filter</b>		/oau	GET		content client_ident=<client_ident>&-type: applica tion/x-	Authentication is done via ssl mutual authentication. In combination with 'client_ident' and 'filterStatus'

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
ntst ore /get Key		www-form-urlencoded	filterStatus=<filterStatus>& registered_by=<registered_by>& environment=<environment>& client_key=<client_key>& offset=0& format=xml			optionally add one of these combinations: (register_by environment) OR (registered_by) OR (environment) OR (client_key)
Response					status: <values xmlns= " <a href="http://ns.l7tech.com/2012/11/otk-clientstore">http://ns.l7tech.com/2012/11/otk-clientstore</a> "> <value index="200" type="integer"> <client_ident>value<-text /> <client_name>value</text> <client_key>value</client_key> <secret>value</secret> <charset>value</scope> <scope>value</scope> <callback>value<=UTF-8 /> <callback>value</callback> <environment>value</environment>	<p><b>client_ident</b> : the client_ident</p> <p><b>filterStatus</b> : the status</p> <p><b>registered_by</b> (optional) : whoever has registered the client</p> <p><b>environment</b> (optional) : the environment</p> <p><b>client_key</b> (optional) : the client_key. In the context of OAuth 1.0 it is the oauth_consumer_key, in OAuth 2.0 the client_id</p> <p><b>offset</b> (optional) : The API returns up to 100 values. This parameter is the query offset which can be used for pagination purposes.</p> <p><b>format</b> (optional) : Either 'xml' or 'json'. The response is returned in the requested format.</p>

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					/environment> <expiration>value</expiration> <status>value</status><created>value</created><created_by>value</created_by><client_key_custom>URL-Encoded-JSON-structure</client_key_custom><servicelds>value</servicelds><accountPlanMappingIds>value</accountPlanMappingIds> </value> </values>	'index' is a temporary list number. No value is tied to the index value. It is valid for this response only.
Response					status: { "values": { "value": [ { "client_ident": 200 "value", "client_name": "value", content "client_key": "value", "secret": "value", -type: "scope": "value", "callback": "value", applica "expiration": "value", "status": "value", tion "created": "value", "created_by": "value", /json; "client_key_custom": "{URL-Encoded-JSON- charset Structure}", "servicelds": "value", =UTF-8 "accountPlanMappingIds": "value" ] } } }	A JSON response that includes the key 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
Error-Response					x-ca-err: { "error": "invalid_request", "error_description": "Missing or unknown parameters" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	If any required parameters or headers are missing, the request will fail.
Error-Response					x-ca-err: { "error": "invalid_request", "error_description": "SSL with client authentication is required" } 4 Pragma : no- cache Cache- Control : no-	Authentication failed.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					store status: 403 content -type: application /json; charset =UTF-8	
Error-Response					x-ca-err: { "error":"invalid_request", "error_description":"The client certificate 400020 is not valid" } 5 Pragma : no- cache Cache- Control : no- store status: 401 content -type: application /json; charset =UTF-8	The client could not be authenticated.
Error-Response					x-ca-err: { "error":"invalid_request", "error_description":"The path component 400030 of the URL is invalid" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: text /xml; charset =UTF-8	The path component is invalid.

ID	Operation	URL- Path	HTTP Method	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
	Error- Response			Allow: GET Pragma : no- cache Cache- Control : no- store status: 405	The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
	Error- Response			x-ca- err: 400000 0 Pragma : no- cache Cache- Control : no- store status: 500	Unknown error.
get_client_client_id_values	Get the values of a clients and client_ids in one request. By default this method is used when processing oauth token requests.	/oau /clients /client_ids	GET /get	content-type: application/x-www-form-urlencoded client_key=<client_key>&format=xml	Authentication is done via ssl mutual authentication. The response will have two more child elements which are surrounding the result of 'get' and 'getKey' operations.  <b>client_key</b> : The OAuth 2.0 client_id. In the context of OAuth 1.0, this is the oauth_consumer_key. <b>format</b> (optional) : Either 'xml' or 'json'. The response is returned in the requested format.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
	Response				<pre>status: &lt;values xmlns="http://ns.l7tech.com/2012 200   /11/otk-clientstore"&gt; &lt;clients&gt;&lt;values&gt; ... content see result from 'get' ... &lt;/values&gt;&lt;/clients&gt; -type: &lt;keys&gt;&lt;values&gt; ... see result from 'getKey' text   ... &lt;/values&gt;&lt;/keys&gt; &lt;/values&gt; /xml; charset =UTF-8</pre>	
	Response				<pre>status: { "values": { "keys": { "values": { "value": [ 200   [...] } }, "clients": { "values": { "value": [...] } } } } content -type:  applica tion /json; charset =UTF-8</pre>	
	Error- Response				<pre>x-ca- err: { "error": "invalid_request",         "error_description": "Missing or unknown 400010 parameters" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type:  applica tion /json; charset =UTF-8</pre>	If any required parameters or headers are missing, the request will fail.
	Error- Response				<pre>x-ca- err: { "error": "invalid_request",         "error_description": "SSL with client 400020 authentication is required" } 4 Pragma : no- cache Cache- Control : no- store status:</pre>	Authentication failed.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					403 content -type: application /json; charset =UTF-8	
Error- Response				x-ca- err:	{ "error":"invalid_request", "error_description":"The client certificate 400020 is not valid" }	The client could not be authenticated.
				5 Pragma : no- cache Cache- Control : no- store status: 401 content -type: application /json; charset =UTF-8		
Error- Response				x-ca- err:	{ "error":"invalid_request", "error_description":"The path component 400030 of the URL is invalid" }	The path component is not valid.
				3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: text /xml; charset =UTF-8		
Error- Response						

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				Allow: GET Pragma : no- cache Cache- Control : no- store status: 405		The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
	Error- Response			x-ca- err: 400000 0 Pragma : no- cache Cache- Control : no- store status: 500		Unknown error.
	<b>Registers</b> nt a new client on the OAuth server.	/oau /clie ntst ore /stor e	POST	content-type: application/x-www-form-urlencoded	client_id= <client_ident>& name=<name>& org=<org>& registered_by=<registered_by>& type=<type>& description=<description>& client_custom=<client_custom>& persist_type=client	Authentication is done via ssl mutual authentication.  <b>client_ident</b> : The unique identifier of this client within the oauth client storage. <b>name</b> : The name given for this client. <b>org</b> : The organization that is registering this client. <b>registered_by</b> : The user who registers. <b>type</b> : For OAuth 2.0 clients, either 'confidential' or 'public'. <b>description</b> (optional) : An

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
						optional description. <b>client_custom</b> (optional) : A JSON structure containing custom values. <b>persist_type</b> : MUST be 'client'.
	Response				status: persisted 200 content -type: text /plain; charset =UTF-8	
	Error-Response				x-ca- { "error":"invalid_request", err: "error_description":"SSL with client 400020 authentication is required" } 4 Pragma : no- cache Cache- Control : no- store status: 403 content -type: applica tion /json; charset =UTF-8	Authentication failed.
	Error-Response				x-ca- { "error":"invalid_request", err: "error_description":"The client certificate 400020 is not valid" } 5 Pragma : no- cache Cache- Control : no- store status: 401	The client could not be authenticated.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					content -type: application/json; charset=UTF-8	
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The path component 400130 of the URL is invalid" }	The path component is invalid.
				3 Pragma : no-cache Cache-Control : no-store status: 400 content -type: text/xml; charset=UTF-8		
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"Client(_id) value could 400130 not be persisted" }	The client could not be persisted due to duplicate values.
				0 Pragma : no-cache Cache-Control : no-store status: 400 content -type: application/json; charset=UTF-8		
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The value for 400130 'persist_type' is not supported" }	The used 'persist_type' is invalid.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					1 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	
Error- Response				x-ca- err:	{ "error":"invalid_request", "error_description":"Missing or unknown 400110 parameters" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	If any required parameters or headers are missing, the request will fail.
Error- Response				Allow: POST Pragma : no- cache Cache- Control : no- store status: 405	The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools. ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.	
						Unknown error.

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error-Response				x-ca-err: 400100 0 Pragma : no- cache Cache- Control : no- store status: 500		
<b>Registers a new client_id for a known client on the OAuth server.</b>		/oau	POST	content-type: application/x-www-form-urlencoded client_key=<client_key>& name=<name>& registered_by=<registered_by>& status=<status>& urlenco_expiration=<expiration>& secret=<secret>& scope=oob& callback=oob& environment=ALL& client_key_custom=<client_key_custom>& service_ids=<service_ids>& account_plan_mapping_ids=<account_plan_mapping_ids>& persist_type=client_key		<p>Authentication is done via ssl mutual authentication.</p> <p><b>client_id</b> : It has to match an existing 'client_ident' on the OAuth server. This client_id will be added as an addition for the referenced client.</p> <p><b>client_key</b> : The client_id to be registered. In the context of OAuth 1.0 this is the 'oauth_consumer_key'.</p> <p><b>name</b> : The name given for this client.</p> <p><b>registered_by</b> : The user who registers.</p> <p><b>status</b> : Either 'ENABLED' or 'DISABLED'.</p> <p><b>expiration</b> : This is the expiration date in seconds (unix timestamp). The value '0' indicates no expiration.</p> <p><b>secret</b> : The client_secret to be registered. In the</p>

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
						<p>context of OAuth 1.0 this is the 'oauth_consumer_key_secret'. <b>secret</b> : For OAuth 2.0 clients of 'type=public' this value is optional</p> <p><b>scope</b> (optional) : The SCOPE to be available for this client_id. Used with OAuth 2.0 only.</p> <p><b>callback</b> (optional) : In OAuth 2.0 this is the 'redirect_uri'. Although it is optional it should contain a value.</p> <p><b>environment</b> (optional) : This can be used to categorize the client_id, e.g.: 'environment=iOS' if this client_id will be used for iOS clients.</p> <p><b>client_key_custom</b> (optional) : A JSON structure containing custom values.</p> <p><b>service_ids</b> (optional) : A comma separated list of serviceIds for which this client_id is valid. It is used in the context of CA API Portal.</p> <p><b>account_plan_map</b></p> <p><b>ping_ids</b> (optional) : A comma separated list of account plan mapping ID's. It is</p>

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
						used in the context of CA API Portal. <b>persist_type</b> : MUST be 'client_key'.
Response					status: persisted 200 content -type: text /plain; charset =UTF-8	
Error-Response					x-ca- { "error":"invalid_request", err: "error_description":"Client(_id) value could 400130 not be persisted" } 0 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	The client_id could not be persisted due to duplicate values.
Error-Response					x-ca- { "error":"invalid_request", err: "error_description":"SSL with client 400020 authentication is required" } 4 Pragma : no- cache Cache- Control : no- store status: 403 content -type: applica tion	Authentication failed.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error- Response					/json; charset =UTF-8	
Error- Response					x-ca- { "error":"invalid_request", err: "error_description":"The client certificate 400020 is not valid" } 5 Pragma : no- cache Cache- Control : no- store status: 401 content -type: applica tion /json; charset =UTF-8	The client could not be authenticated.
Error- Response					x-ca- { "error":"invalid_request", err: "error_description":"The path component 400130 of the URL is invalid" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: text /xml; charset =UTF-8	The path component is invalid.
Error- Response					x-ca- { "error":"invalid_request", err: "error_description":"The value for 400130 'persist_type' is not supported" } 1 Pragma : no- cache	The used 'persist_type' is invalid.

ID	Operation	URL-Path	HTTP Method	HTTP Header	Body/Query Params	Comment
					(for GET attach params to URL-PATH)	
					Cache-Control : no-store status: 400 content-type: application/json; charset=UTF-8	
Error-Response		x-ca-err:	{ "error":"invalid_request", "error_description":"Missing or unknown parameters" }	3	Pragma : no-cache Cache-Control : no-store status: 400 content-type: application/json; charset=UTF-8	If any required parameters or headers are missing, the request will fail.
Error-Response		Allow:	POST		Pragma : no-cache Cache-Control : no-store status: 405	The HTTP method is not valid. <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
Error-Response		x-ca-err:	4001000	0	Pragma	Unknown error.

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params (for GET attach params to URL-PATH)	Comment
					: no-cache Cache-Control : no-store status: 500	
<a href="#">ntt_j</a>	<b>Registers a new client including a new client_id on the OAuth server.</b>	/oau /client /clients	POST		content-type: application/x-www-form-urlencoded; ded	<p>Authentication is done via ssl mutual authentication.</p> <p><b>client_ident</b> : The unique identifier of this client within the oauth client storage.</p> <p><b>client_key</b> : The client_id to be registered. In the context of OAuth 1.0 this is the 'oauth_consumer_key'.</p> <p><b>name</b> : The name given for this client.</p> <p><b>org</b> : The organization that is registering this client.</p> <p><b>registered_by</b> : The user who registers.</p> <p><b>type</b> : For OAuth 2.0 clients, either 'confidential' or 'public'.</p> <p><b>description</b> (optional) : An optional description.</p> <p><b>status</b> : Either 'ENABLED' or 'DISABLED'.</p> <p><b>expiration</b> : This is the expiration date in seconds (unix timestamp). The value '0' indicates no expiration.</p>

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
						<p><b>secret</b> : The client_secret to be registered. In the context of OAuth 1.0 this is the 'oauth_consumer_key_secret'.</p> <p><b>scope</b> (optional) : The SCOPE to be available for this client_id. Used with OAuth 2.0 only.</p> <p><b>callback</b> (optional) : In OAuth 2.0 this is the 'redirect_uri'. Although it is optional it should contain a value.</p> <p><b>environment</b> (optional) : This can be used to categorize the client_id, e.g.: 'environment=iOS' if this client_id will be used for iOS clients.</p> <p><b>client_custom</b> (optional) : A JSON structure containing custom values for a client.</p> <p><b>client_key_custom</b> (optional) : A JSON structure containing custom values for a client_key (client_id, oauth_consumer_key, apiKey).</p> <p><b>service_ids</b> (optional) : A comma separated list of serviceIds for which this client_id is valid. It is used in the context of CA API Portal.</p>

ID	Operation	URL-Path	HTTP Method	HTTP Headers	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
						<b>account_plan_map</b> <b>ping_ids</b> (optional) : A comma separated list of account plan mapping ID's. It is used in the context of CA API Portal. <b>persist_type</b> : MUST be 'client_and_key'.
Response					status: persisted 200 content -type: text /plain; charset =UTF-8	
Error-Response					x-ca- { "error":"invalid_request", err: "error_description":"Client(_id) value could 400130 not be persisted" } 0 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	The client_id could not be persisted due to duplicate values.
Error-Response					x-ca- { "error":"invalid_request", err: "error_description":"SSL with client 400020 authentication is required" } 4 Pragma : no- cache Cache- Control : no-	Authentication failed.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					store status: 403 content -type: application /json; charset =UTF-8	
Error-Response					x-ca-err: { "error":"invalid_request", "error_description":"The client certificate 400020 is not valid" } 5 Pragma : no- cache Cache- Control : no- store status: 401 content -type: application /json; charset =UTF-8	The client could not be authenticated.
Error-Response					x-ca-err: { "error":"invalid_request", "error_description":"The path component 400130 of the URL is invalid" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: text /xml; charset =UTF-8	The path component is invalid.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error-Response				x-ca-err: 1 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	{ "error":"invalid_request", "error_description":"The value for 'persist_type' is not supported" }	The used 'persist_type' is invalid.
Error-Response				x-ca-err: 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	{ "error":"invalid_request", "error_description":"Missing or unknown parameters" }	If any required parameters or headers are missing, the request will fail.
Error-Response				Allow: POST Pragma : no- cache Cache- Control		The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools. ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					: no-store status: 405	
	Error-Response			x-ca-err:	400100 0 Pragma : no-cache Cache-Control : no-store status: 500	Unknown error.
delete_client		Deletes a client and all its OAuth client_id's of the OAuth server.	/oau DELETE	content	client_ident=<client_ident> -type: application/x-www-form-urlencoded	Authentication is done via ssl mutual authentication. <b>client_ident</b> : The unique identifier of this client that should be deleted. All client_id's will also be removed.
	Response			status:	{n} client(s) deleted 200 content -type: text /plain; charset =UTF-8	
	Error-Response			x-ca-err:	{ "error":"invalid_request", "error_description":"Missing or unknown parameters" } 3 Pragma : no-cache Cache-Control : no-store status: 400 content	If any required parameters or headers are missing, the request will fail.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					-type: application /json; charset =UTF-8	
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"SSL with client 400020 authentication is required" } 4 Pragma : no- cache Cache- Control : no- store status: 403 content -type: application /json; charset =UTF-8	Authentication failed.
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The client certificate 400020 is not valid" } 5 Pragma : no- cache Cache- Control : no- store status: 401 content -type: application /json; charset =UTF-8	The client could not be authenticated.
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The path 400230 component of the URL is invalid" }	The path component is invalid.

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params (for GET attach params to URL-PATH)	Comment
					3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: text /xml; charset =UTF-8	
	Error- Response				Allow: DELETE Pragma : no- cache Cache- Control : no- store status: 405	The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616 (https://tools. ietf.org/html/rfc2616)</a> and contains a comma separated list of valid HTTP methods.
	Error- Response				x-ca- err: 400200 0 Pragma : no- cache Cache- Control : no- store status: 500	Unknown error.
<b>nt</b>	<b>Revokes one or many client_id (s).</b>	/oau	DELETE	content	client_ident=<client_ident>& -type: client_key=<client_key>	Authentication is done via ssl mutual authentication. In comparison to ' /delete' this API only deletes client_ids, not the client itself.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
						Only one of the parameters can be and has to be used.
						<b>client_ident</b> (optional) : The unique identifier of this client of which the client_ids should be revoked. Use this instead of 'client_key'. <b>client_id</b> (optional) : The client_id to be revoked. In the context of OAuth 1.0 this is the 'oauth_consumer_key'. Use this instead of 'client_ident'.
Response					status: {n} client_id(s) deleted 200 content -type: text /plain; charset =UTF-8	
Error-Response		x-ca-err:	{ "error":"invalid_request", "error_description":"Missing or unknown 400210 parameters" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: application /json; charset =UTF-8			If any required parameters or headers are missing, the request will fail.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error-Response				x-ca-err: 4	{ "error":"invalid_request", "error_description":"SSL with client authentication is required" }  Pragma : no- cache Cache- Control : no- store status: 403 content -type: application /json; charset =UTF-8	Authentication failed.
Error-Response				x-ca-err: 5	{ "error":"invalid_request", "error_description":"The client certificate is not valid" }  Pragma : no- cache Cache- Control : no- store status: 401 content -type: application /json; charset =UTF-8	The client could not be authenticated.
Error-Response				x-ca-err: 3	{ "error":"invalid_request", "error_description":"The path component of the URL is invalid" }  Pragma : no- cache Cache- Control : no-	The path component is invalid.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				store status: 400 content -type: text /xml; charset =UTF-8		
Error- Response				Allow: DELETE Pragma : no- cache Cache- Control : no- store status: 405		The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
Error- Response				x-ca- err: 400200 0 Pragma : no- cache Cache- Control : no- store status: 500		Unknown error.
<b>int</b>	<b>Updates the values of the client of the given client_ide nt</b>	/oau th /clie ntst ore /upd ate	PUT	content client_ident=<client_ident>& -type: applica tion/x- type=<type>& www- description=<description>& form- org=<org>& urlenco client_custom=<client_custom> ded		Authentication is done via ssl mutual authentication. Empty values or missing parameters will overwrite existing values!
						<b>client_ident</b> : the client identified by the <b>client_ident</b> will be updated <b>name</b> : the new name.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
						<b>type</b> (optional) : the new type (may be empty), either 'confidential' or 'public'. <b>description</b> (optional) : A description. <b>org</b> (optional) : the new organization (may be empty) <b>client_custom</b> (optional) : A JSON structure containing custom values for a client.
Response					status: {no-of-clients} client(s) updated 200 content -type: text /plain; charset =UTF-8	
Error-Response		x-ca-err:	{ "error":"invalid_request", "error_description":"Missing or unknown parameters" }	3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: application /json; charset =UTF-8	If any required parameters or headers are missing, the request will fail.	
Error-Response		x-ca-err:	{ "error":"invalid_request", "error_description":"SSL with client authentication is required" }	4 Pragma	Authentication failed.	

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					: no-cache Cache-Control : no-store status: 403 content-type: application/json; charset =UTF-8	
Error-Response				x-ca-err:	{ "error": "invalid_request", "error_description": "The client certificate 400020 is not valid" }	The client could not be authenticated.
Error-Response				x-ca-err:	{ "error": "invalid_request", "error_description": "The path component 400330 of the URL is invalid" }	The path component is invalid.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					text /xml; charset =UTF-8	
Error- Response				Allow: PUT, POST Pragma : no- cache Cache- Control : no- store status: 405		The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
Error- Response				x-ca- err: 400300 0 Pragma : no- cache Cache- Control : no- store status: 500		Unknown error.
<b>Updates values for the given client_id.</b>	/oau int th the /clie ntst ore /upd ate	PUT		content-type: application/x-www-form-urlencoded	client_key=<client_key>& secret=<secret>& status=<status>& scope=oob& callback=oob& environment=ALL& client_key_custom=<client_key_custom>& service_ids=<service_ids>& account_plan_mapping_ids=<account_plan_mapping_ids>	Authentication is done via ssl mutual authentication. Empty values will overwrite existing values!  <b>client_key</b> : update the values of the <b>secret</b> : The client_secret to be updated. In the context of OAuth 1.0 this is the 'oauth_consumer_key'.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
						'oauth_consumer_key_secret'. <b>status</b> : Either 'ENABLED' or 'DISABLED'. <b>scope</b> (optional) : the new scope. The SCOPE to be updated for this client_id. Used with OAuth 2.0 only. <b>callback</b> (optional) : new callback (redirect_uri). In OAuth 2.0 this is the 'redirect_uri'. This value is only verified in OAuth 2.0. <b>environment</b> (optional) : set the new environment. This can be used to categorize the client_id, e.g.: 'environment=iOS' if this client_id will be used for iOS clients. <b>client_key_custom</b> (optional) : A JSON structure containing custom values for a client_key (client_id, oauth_consumer_key, apiKey). <b>service_ids</b> (optional) : A comma separated list of serviceIds for which this client_id is valid. It is used in the context of CA API Portal. <b>account_plan_mapping_ids</b> (optional) : A comma separated list of account plan

ID	Operation	URL-Path	HTTP Method	HTTP Header	Body/Query Params (for GET attach params to URL-PATH)	Comment
						mapping ID's. It is used in the context of CA API Portal.
Response					status: {no-of-clients} client_key(s) updated 200 content -type: text /plain; charset =UTF-8	
Error-Response					x-ca- { "error":"invalid_request", err: "error_description":"Missing or unknown 400310 parameters" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	If any required parameters or headers are missing the request will fail.
Error-Response					x-ca- { "error":"invalid_request", err: "error_description":"SSL with client 400020 authentication is required" } 4 Pragma : no- cache Cache- Control : no- store status: 403 content -type: applica	Authentication failed.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error- Response					tion /json; charset =UTF-8	
Error- Response				x-ca- err: 400020	{ "error":"invalid_request", "error_description":"The client certificate is not valid" } 5 Pragma : no- cache Cache- Control : no- store status: 401 content -type: applica tion /json; charset =UTF-8	The client could not be authenticated.
Error- Response				x-ca- err: 400330	{ "error":"invalid_request", "error_description":"The path component of the URL is invalid" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: text /xml; charset =UTF-8	The path component is invalid.
Error- Response				Allow: PUT, POST Pragma : no- cache	Allow : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools. ietf.org/html/rfc2616</a> )	The HTTP method is not valid

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params (for GET attach params to URL-PATH)	Comment
				Cache- Control : no- store status: 405		and contains a comma separated list of valid HTTP methods.
	Error- Response			x-ca- err: 400300 0 Pragma : no- cache Cache- Control : no- store status: 500		Unknown error.
update_client_id_registers_by	<b>Updates a client_id.</b>	/oau /client_id_registers_by	POST	content-type: application/x-www-form-urlencoded	client_key=<client_key>&created_by=ALL	Authentication is done via ssl mutual authentication.
	Response			status: 200 content-type: text/plain; charset=UTF-8	n client_key(s) updated	<b>client_key</b> : The client_id to be updated. In the context of OAuth 1.0 this is the 'oauth_consumer_key'. <b>created_by</b> : Update the value of whoever registered the client_id.
	Error- Response			x-ca-err: 3 Pragma : no- cache	{ "error":"invalid_request", "error_description":"Missing or unknown parameters" }	If any required parameters or headers are missing, the request will fail.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					Cache-Control : no-store status: 400 content-type: application/json; charset=UTF-8	
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"SSL with client authentication is required" }	Authentication failed.
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The client certificate is not valid" }	The client could not be authenticated.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error- Response					/json; charset =UTF-8	
Error- Response				x-ca- err: 400330 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: text /xml; charset =UTF-8	{ "error":"invalid_request", "error_description":"The path component 400330 of the URL is invalid" }  The path component is invalid.	
Error- Response				Allow: PUT, POST Pragma : no- cache Cache- Control : no- store status: 405	The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> and contains a comma separated list of valid HTTP methods.	
Error- Response				x-ca- err: 400300 0 Pragma : no- cache Cache- Control : no- store status: 500	Unknown error.	

Request values of a given client

These APIs will use the given parameters as the search parameter in selecting only certain clients. All of these APIs accept the additional parameter "format". "Format" can either take the value "xml" (which is the default) or "json". If it is "json", the values will be returned as a JSON message.

---

APIs

---

`/get`

---

`/get?name=<value>&`

---

`/get?name=<value>&org=<value>`

---

`/get?registered_by=<value>`

---

`/get?registered_by=<value>&org=<value>`

---

`/get?client_key=<value>`

---

`/get?client_ident=<value>&`

---

---

Parameters

---

**name:** the name of the client as a filter**org:** the organization of the client**registered\_by:** all clients registered by the given parameter**client\_key:** client values that "own" the given client\_key**client\_ident:** the unique identifier identifying a certain client

---

## Response:

- status: 200, content-type: text/xml, body
- status: 200, content-type: application/json, body
- status: 400, content-type: text/plain, body: Non-valid parameter list for this operation

## Request values of a given client\_key

These APIs will use the given parameters as a search parameter in selecting only certain client\_keys. All of these APIs accept the additional parameter "format". "Format" can either take the value "xml" (which is the default) or "json". If it is "json", the values will be returned as a json message.

`/getKey returns all values of all client_keys`

---

APIs

---

`/getKey`

---

`/getKey?client_key=<value>`

---

`/getKey?name=<value>`

---

`/getKey?name=<value>&org=<value>`

---

`/getKey?org=<value>`

---

---

*/getKey?client\_ident=<value>*

---

Response

- status: 200, content-type: text/xml, body
  - status: 200, content-type: application/json, body
- 

Parameters

**name:** the name of the client used for this key

**org:** the name of the organization used for this key

**client\_ident:** the identifier of the client issued for this

---

Response:

- status: 200, content-type: text/xml, body:
- status: 200, content-type: application/json, body:
- status: 400, content-type: text/plain, body: Non-valid parameter list for this operation

## Request values of a given client, client\_key at once

These APIs will use the given parameters as search parameter to select certain clients and client\_keys. All of these APIs accept the additional parameter "format". "Format" can either take the value "xml" (which is the default) or "json". If it is "json", the values will be returned as a json message.

*/getAll*

"*/getAll*" can always be used with the same parameters as for "*/get*" and "*getKey*". It will respond with the content of the selected client and client\_key.

Response:

- status: 200, content-type: text/xml, body
- status: 200, content-type: application/json, body
- status: 400, content-type: text/plain, body: Non-valid parameter list for this operation

## Tokenstore API

This API updated for OAuth Toolkit 3.3.01 allows you to manage tokens.

All tokenstore API endpoints start here:

*<Gateway\_host\_and\_port>/oauth/tokenstore/\**

For example:

<http://myGateway.com:8080/oauth/tokenstore/store>

- [persist临时 \(see page 286\)](#)
- [persist\\_token\\_oauth1 \(see page 289\)](#)
- [persist\\_token\\_oauth2 \(see page 292\)](#)
- [update\\_token\\_status \(see page 296\)](#)
- [update\\_oauth1\\_token\\_owner \(see page 298\)](#)
- [update\\_oauth1\\_token\\_verifier \(see page 301\)](#)
- [revoke\\_oauth\\_token \(see page 304\)](#)
- [delete\\_oauth\\_token \(see page 307\)](#)
- [disable\\_oauth\\_token \(see page 310\)](#)
- [get\\_oauth\\_token \(see page 312\)](#)
- [get\\_oauth\\_token\\_param \(see page 315\)](#)
- [get\\_oauth\\_token\\_cid\\_ro \(see page 319\)](#)
- [get\\_oauth\\_token\\_status\\_ro \(see page 322\)](#)
- [get\\_temporary\\_token\\_t \(see page 325\)](#)
- [get\\_temporary\\_token\\_t\\_v \(see page 328\)](#)
- [get\\_temporary\\_token\\_cid\\_ro \(see page 331\)](#)
- [register\\_jwt \(see page 334\)](#)
- [lookup\\_jwt \(see page 338\)](#)
- [remove\\_jwt \(see page 341\)](#)

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
1p or ar y	Registers a new temporary token on the OAuth server.	/oau/th/tok/est/stor/e	POST		content token=<token>&-type: application/x-www-form-urlencoded resource_owner=<resource_owner>&secret=<secret>&custom=<custom>	<p>Authentication is done via ssl mutual authentication.</p> <p><b>token</b> : the token, either a request_token (OAuth 1.0) or an authorization_code (OAuth 2.0) to be stored.</p> <p><b>expiration</b> : the date this token expires (UNIX timestamp in seconds).</p> <p><b>client_key</b> : the oauth_consumer_key for which this token was issued.</p> <p><b>client_name</b> : the client name for which this token was issued.</p>

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
						<p><b>callback</b> : the callback uri.  <b>'redirect_uri'</b> in OAuth 2.0. The parameter may be left empty.</p> <p><b>scope</b> : the scope used with OAuth 2.0. The parameter may be left empty.</p> <p><b>resource_owner</b>(optional) : the resource_owner who granted this token. REQUIRED for OAuth 2.0 (authorization code), OPTIONAL for OAuth 1.0. The request_token is issued without a resource_owner involved.</p> <p><b>secret</b> (optional) : the secret for this token. Optional because OAuth 2.0 does not have a secret if configured to issue BEARER tokens.</p> <p><b>custom</b> (optional) : A JSON structure containing custom values. This is only supported for OAuth 2.0.</p>
Response					status: persisted 200 content -type: text /plain; charset =UTF-8	
Error-Response			x-ca-err:	{ "error":"invalid_request", "error_description":"Missing or 410010 unknown parameters" }		If any required parameters or headers are missing, the request will fail.
			3	Pragma		
			: no-	cache		
			Cache-			

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					Control : no-store status: 400 content-type: application/json; charset=UTF-8	
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The path 410030 component of the URL is invalid" } 3 Pragma : no-cache Cache-Control : no-store status: 400 content-type: application/json; charset=UTF-8	The path component is invalid.
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"SSL with client 410020 authentication is required" } 4 Pragma : no-cache Cache-Control : no-store status: 403 content-type: application/json;	Authentication failed.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params	Comment
					tion /json; charset =UTF-8	
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The client certificate is not valid" }	The client could not be authenticated.
Error-Response				5	Pragma : no-cache Cache-Control : no-store status: 401 content-type: application/json; charset =UTF-8	
Error-Response				Allow: POST	Pragma : no-cache Cache-Control : no-store status: 405	The HTTP method is not valid. <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
Error-Response				x-ca-err: 410000	0 Pragma : no-cache Cache-Control : no-store status: 500	Unknown error.
en	POST			token=<token>&		

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
_0_1	<b>Registers a new OAuth 1.0 token on the OAuth server.</b>	/oauth/1.0/token	POST		content-type: application/x-www-form-urlencoded secret= <secret>&client_key= <client_key>&status= <status>&www-temp_token= <temp_token>	Authentication is done via ssl mutual authentication. This API adds the client_name and resource_owner associated with the temp_token.
						<b>token</b> : This is the oauth_token. <b>secret</b> : the shared secret for this token. <b>expiration</b> : the date this token expires (UNIX timestamp in seconds). <b>client_key</b> : the oauth_consumer_key for which this token was issued. It must match the oauth_consumer_key that has received the temporary token <b>status</b> : either ENABLED or DISABLED. DISABLED tokens will cause a request to fail. <b>temp_token</b> : the temporary token exchanged for this token.
	Response				status: persisted 200 content-type: text/plain; charset=UTF-8	
	Error-Response				x-ca-err: { "error": "invalid_request", "error_description": "Missing or 410110 unknown parameters" } 3 Pragma: no-cache Cache-Control	If any required parameters or headers are missing, the request will fail.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params	Comment
					: no-store status: 400 content-type: application/json; charset =UTF-8	
Error-Response				x-ca-err: 410130	{ "error": "invalid_request", "error_description": "The path component of the URL is invalid" }	The path component is invalid.
Error-Response				x-ca-err: 410020	{ "error": "invalid_request", "error_description": "SSL with client authentication is required" }	Authentication failed.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params	Comment
					tion /json; charset =UTF-8	
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The client certificate is not valid" }	The client could not be authenticated.
Error-Response				5		
Error-Response				Pragma : no-cache		
Error-Response				Cache-Control : no-store		
Error-Response				status: 401		
Error-Response				content-type: application/json;		
Error-Response				charset =UTF-8		
Error-Response				Allow: POST		The HTTP method is not valid
Error-Response				Pragma : no-cache		<b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
Error-Response				Cache-Control : no-store		
Error-Response				status: 405		
Error-Response				x-ca-err: 410100		Unknown error.
				0		
				Pragma : no-cache		
				Cache-Control : no-store		
				status: 500		
en	POST			token=<token>&		

ID	Operation	URL-Path	HTTP Method	HTTP Headers	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
-0- a ut h 2	<b>Registers a new OAuth 2.0 access_token on the OAuth server.</b> <b>Optionally also a refresh_token.</b>	/oauth/token	POST		content-type: application/x-www-form-urlencoded client_id: <client_id>&client_secret: <client_secret>&grant_type: <grant_type>&refresh_token: <refresh_token>&scope: <scope>&state: <state>	<p>Authentication is done via ssl mutual authentication.</p> <p>This API updates an existing token of the combination 'resource_owner-client_key'. The original token becomes invalid!</p> <p><b>token</b> : This is the access_token.</p> <p><b>expiration</b> : the date this token expires (timestamp in seconds).</p> <p><b>client_key</b> : the client_id for which this token was issued.</p> <p><b>resource_owner</b> : the resource_owner who granted this token. For tokens issued via 'client_credentials' flow this should be the client name.</p> <p><b>status</b> : either ENABLED or DISABLED. DISABLED tokens will cause a request to fail.</p> <p><b>scope</b> : the scope that was granted.</p> <p><b>client_name</b> : the client name for which this token was issued.</p> <p><b>secret</b> (optional) : the shared secret for this token. Not used with token profile 'BEARER'.</p> <p><b>custom</b> (optional) : A JSON structure containing custom values.</p> <p><b>rtoken</b> (optional) : the refresh token.</p> <p><b>rexpiration</b> (optional) : the date this refresh</p>

ID	Operation	URL- Path	HTTP Method	HTTP Header	Query Params (for GET attach params to URL-PATH)	Comment
						token expires (timestamp in seconds). This is required if 'rtoken' is provided.
	Response				status: persisted 200 content -type: text /plain; charset =UTF-8	
	Error- Response				x-ca- { "error":"invalid_request", err: "error_description":"Missing or 410210 unknown parameters" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	If any required parameters or headers are missing, the request will fail.
	Error- Response				x-ca- { "error":"invalid_request", err: "error_description":"The path 410230 component of the URL is invalid" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica	The path component is invalid.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error- Response					tion /json; charset =UTF-8	
Error- Response					x-ca- { "error":"invalid_request", err: "error_description":"SSL with client 410020 authentication is required" } 4 Pragma : no- cache Cache- Control : no- store status: 403 content -type: applica tion /json; charset =UTF-8	Authentication failed.
Error- Response					x-ca- { "error":"invalid_request", err: "error_description":"The client 410020 certificate is not valid" } 5 Pragma : no- cache Cache- Control : no- store status: 401 content -type: applica tion /json; charset =UTF-8	The client could not be authenticated.
Error- Response					Allow: POST Pragma : no- cache	The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org">RFC 2616</a> <a href="https://tools.ietf.org">https://tools.ietf.org</a>

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				Cache- Control : no- store status: 405		/html/rfc2616) and contains a comma separated list of valid HTTP methods.
	Error- Response			x-ca- err: 410200 0 Pragma : no- cache Cache- Control : no- store status: 500		Unknown error.
update_token	Updates token status on the OAuth server. tu s	/oau /tok enst ore /upd ate	PUT	content token=<token>& -type: status=<status>		Authentication is done via ssl mutual authentication.
				application/x- www-form- urlenco ded		<b>token</b> : the oauth_token or access_token to be updated. <b>status</b> : either ENABLED or DISABLED. DISABLED tokens will cause a request to fail.
	Response				status: {no-of-tokens} token(s) updated 200 content -type: text /plain; charset =UTF-8	
	Error- Response			x-ca- err: 410310 3 Pragma : no- cache Cache- Control	{ "error":"invalid_request", "error_description":"Missing or unknown parameters" }	If any required parameters or headers are missing, the request will fail.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params	Comment
					: no-store status: 400 content-type: application/json; charset =UTF-8	
Error-Response		x-ca-err: 410330		{ "error": "invalid_request", "error_description": "The path component of the URL is invalid" }		The path component is invalid.
Error-Response		x-ca-err: 410020		{ "error": "invalid_request", "error_description": "SSL with client authentication is required" }		Authentication failed.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The client certificate is not valid" }	The client could not be authenticated.
Error-Response				5		
Error-Response				Allow:		The HTTP method is not valid
				PUT		<b>Allow :</b> This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
Error-Response				Pragma : no-cache		
				Cache-Control : no-store		
				status: 405		
Error-Response				x-ca-err:		Unknown error.
				410300		
				0		
				Pragma : no-cache		
				Cache-Control : no-store		
				status: 500		

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
<b>h_1_t_</b> <b>ok_e_n_</b> <b>-_o_w_</b> <b>n_</b>	<b>Sets the resources_ owner associated with the request_to token in the context of OAuth 1.0 on the OAuth server.</b>	/oau/th/tok/est/ore/upd/ate		content-type: application/x-www-form-urlencoded	token= <token>&-type: resource_owner=<resource_owner>	Authentication is done via ssl mutual authentication.  <b>token</b> : the OAuth 1.0 request_token after it has been authorized by a resource_owner. <b>resource_owner</b> : the resource_owner who authorized this token.
	Response				status: {no-of-tokens} token(s) updated 200 content-type: text/plain; charset=UTF-8	
	Error-Response			x-ca-err: 3	{ "error": "invalid_request", "error_description": "Missing or 410310 unknown parameters" }	If any required parameters or headers are missing, the request will fail.
	Error-Response			x-ca-err: 3	{ "error": "invalid_request", "error_description": "The path 410330 component of the URL is invalid" }	The path component is invalid.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params	Comment
					cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	
Error- Response				x-ca- err: 4	{ "error":"invalid_request", "error_description":"SSL with client 410020 authentication is required" }  Pragma : no- cache Cache- Control : no- store status: 403 content -type: applica tion /json; charset =UTF-8	Authentication failed.
Error- Response				x-ca- err: 5	{ "error":"invalid_request", "error_description":"The client 410020 certificate is not valid" }  Pragma : no- cache Cache- Control : no- store status: 401 content -type: applica	The client could not be authenticated.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					tion /json; charset =UTF-8	
	Error-Response				Allow: PUT, POST Pragma : no- cache Cache- Control : no- store status: 405	The HTTP method is not valid. <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
	Error-Response				x-ca- err: 410400 0 Pragma : no- cache Cache- Control : no- store status: 500	Unknown error.
update_oaut h 1 _t ok e n _ ve rif ie r	Sets the verifier associated with the request_to ok context of OAuth 1.0 on the OAuth server.	/oau /tok /upd /at /token	PUT		content token=<token>&-type: application/x-www-form-urlencoded	Authentication is done via ssl mutual authentication.  <b>token</b> : the OAuth 1.0 request_token to receive a verifier. <b>expiration</b> : the date this token expires (timestamp in seconds). <b>verifier</b> : the verifier to be assigned to the token.
	Response				status: {no-of-tokens} token(s) updated 200 content	

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					-type: text /plain; charset =UTF-8	
Error-Response				x-ca-err: 410312 5 Pragma : no- cache Cache- Control : no- store status: 400 content -type: text /plain; charset =UTF-8	The verifier is invalid	A verifier for this token has been set already.
Error-Response				x-ca-err: 410310 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: application /json; charset =UTF-8	{ "error":"invalid_request", "error_description":"Missing or unknown parameters" }	If any required parameters or headers are missing, the request will fail.
Error-Response				x-ca-err: 410330 3 Pragma	{ "error":"invalid_request", "error_description":"The path component of the URL is invalid" }	The path component is invalid.

ID	Operation	URL-Path	HTTP Method	HTTP Header	Query Params	Comment
				: no-cache Cache-Control : no-store status: 400 content-type: application/json; charset=UTF-8		
Error-Response				x-ca-err: 4	{ "error": "invalid_request", "error_description": "SSL with client 410020 authentication is required" }	Authentication failed.
Error-Response				x-ca-err: 5	{ "error": "invalid_request", "error_description": "The client 410020 certificate is not valid" }	The client could not be authenticated.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					application/json; charset=UTF-8	
Error-Response				Allow: PUT, POST Pragma : no- cache Cache- Control : no- store status: 405	The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.	
Error-Response				x-ca- err: 410300 0 Pragma : no- cache Cache- Control : no- store status: 500	Unknown error.	
th	Revokes the given token.	/oau/ _token/ ok/en/ e/access_ n/tok	DELETE	content-type: application/x-www-form-urlencoded	token=<token>& rtoken=<rtoken>& resource_owner=<resource_owner>& client_key=<client_key>	Authentication is done via ssl mutual authentication. Valid parameter combinations: either 'token' or 'rtoken' or 'resource_owner' or 'client_key' or 'resource_owner and client_key'
						<b>token</b> (optional) : the oauth_token or access_token to be revoked. MUST be used on its own <b>rtoken</b> (optional) : the refresh_token to

ID	Operation	URL-Path	HTTP Method	HTTP Header	Query Params (for GET attach params to URL-PATH)	Comment
						be revoked. MUST be used on its own <b>resource_owner</b> (optional) : the resource_owner whose long-living tokens should all be revoked. Can be used with 'client_key' <b>client_key</b> (optional) : the client_key of which all temporary and long-living tokens should be revoked. MAY be used with 'resource_owner'
Response					status: {no-of-tokens} token(s) revoked 200 content -type: text /plain; charset =UTF-8	
Error-Response					x-ca- { "error":"invalid_request", err: "error_description":"Missing or 410610 unknown parameters" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	If any required parameters or headers are missing, the request will fail.
Error-Response					x-ca- { "error":"invalid_request", err: "error_description":"The path 410630 component of the URL is invalid" } 3 Pragma	The path component is invalid.

ID	Operation	URL-Path	HTTP Method	HTTP Header	Query Params	Comment
				: no-cache Cache-Control : no-store status: 400 content-type: application/json; charset=UTF-8		
Error-Response				x-ca-err: 4 Pragma : no-cache Cache-Control : no-store status: 403 content-type: application/json; charset=UTF-8	{ "error":"invalid_request", "error_description":"SSL with client 410020 authentication is required" }	Authentication failed.
Error-Response				x-ca-err: 5 Pragma : no-cache Cache-Control : no-store status: 401 content-type:	{ "error":"invalid_request", "error_description":"The client 410020 certificate is not valid" }	The client could not be authenticated.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				application/json; charset=UTF-8		
	Error- Response			Allow: DELETE Pragma : no- cache Cache- Control : no- store status: 405	The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.	
	Error- Response			x-ca- err: 410600 0 Pragma : no- cache Cache- Control : no- store status: 500	Unknown error.	
th_token_enstoken. _token.	Deletes the given oauth_token. /access_to token.	/oau th /tok enst ore /del ete	DELETE	content token=<token>&-type: application/x-www-form-urlencoded	rtoken=<rtoken>&temp_token=<temp_token>&client_key=<client_key>	Authentication is done via ssl mutual authentication. Only ONE of the parameters can and MUST be used.  <b>token</b> (optional) : the oauth_token or access_token to be deleted. <b>rtoken</b> (optional) : the refresh_token to be deleted. <b>temp_token</b> (optional) : the temporary token to be deleted. <b>client_key</b> (optional) : the client_key of

ID	Operation	URL-Path	HTTP Method	HTTP Headers	HTTP Body/Query Params	Comment
					(for GET attach params to URL-PATH)	
						which all temporary and long-living tokens should be deleted.
	Response				status: {no-of-tokens} token(s) deleted 200 content -type: text /plain; charset =UTF-8	
	Error-Response				x-ca- { "error":"invalid_request", err: "error_description":"Missing or 410710 unknown parameters" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	If any required parameters or headers are missing, the request will fail.
	Error-Response				x-ca- { "error":"invalid_request", err: "error_description":"The path 410730 component of the URL is invalid" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion	The path component is invalid.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
Error-Response					/json; charset =UTF-8	
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"SSL with client authentication is required" } 4 Pragma : no- cache Cache- Control : no- store status: 403 content -type: application /json; charset =UTF-8	Authentication failed.
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The client certificate is not valid" } 5 Pragma : no- cache Cache- Control : no- store status: 401 content -type: application /json; charset =UTF-8	The client could not be authenticated.
Error-Response				Allow: DELETE Pragma : no- cache Cache-		The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org /html/rfc2616</a> ) and

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
				Control : no-store status: 405		contains a comma separated list of valid HTTP methods.
	Error-Response			x-ca-err: 410700 0 Pragma : no-cache Cache-Control : no-store status: 500		Unknown error.
disable_oaut	disables all tokens that were issued to the given client_id. Response	/oau PUT		content client_key=<client_key> -type: applica tion/x-www-form-urlencoded		Authentication is done via ssl mutual authentication. <b>client_key</b> : the client_key of which all long-living tokens should be disabled (status=DISABLED).
	Error-Response			status: {no-of-tokens} token(s) disabled 200 content -type: text /plain; charset =UTF-8		If any required parameters or headers are missing, the request will fail.
				x-ca-err: 410810 3 Pragma : no-cache Cache-Control : no-store status: 400	{ "error":"invalid_request", "error_description":"Missing or unknown parameters" }	

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params	Comment
					content -type: application/json; charset=UTF-8	
Error-Response				x-ca-err: 3	{ "error": "invalid_request", "error_description": "The path component of the URL is invalid" }	The path component is invalid.
Error-Response				x-ca-err: 4	{ "error": "invalid_request", "error_description": "SSL with client authentication is required" }	Authentication failed.
Error-Response						The client could not be authenticated.

ID	Operation	URL-Path	HTTP Method	HTTP Header	Query Params (for GET attach params to URL-PATH)	Comment
				x-ca-err:	{ "error":"invalid_request", "error_description":"The client certificate is not valid" }	
				5		
				Pragma	: no-	
				cache	Cache-	
				Control	Control	
				: no-	: no-	
				store	store	
				status:	status:	
				401	401	
				content	content	
				-type:	-type:	
				application	application	
				/json;	/json;	
				charset	charset	
				=UTF-8	=UTF-8	
Error-Response				Allow:		The HTTP method is not valid
Error-Response				PUT,		<b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
Error-Response				POST		
Error-Response				Pragma		
Error-Response				: no-		
Error-Response				cache		
Error-Response				Cache-		
Error-Response				Control		
Error-Response				: no-		
Error-Response				store		
Error-Response				status:		
Error-Response				405		
		x-ca-err:		410800		Unknown error.
		0				
		Pragma				
		: no-				
		cache				
		Cache-				
		Control				
		: no-				
		store				
		status:				
		500				
to ke	Get the values of tokens.	/oau /th /tok	GET	content-type: applica	offset=0&format=xml	Authentication is done via ssl mutual authentication.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
n		enst ore /get		tion/x- www- form- urlenco ded		<b>offset</b> (optional) : The API returns up to 100 values. This parameter is the query offset which can be used for pagination purposes. <b>format</b> (optional) : Either 'xml' or 'json'. The response is returned in the requested format.
	Response				status: <values xmlns="http://ns.l7tech.com 200 /2012/11/otk-tokenstore"> <value content index="integer"> <token>value</token> -type: <secret>value</secret> text <expiration>value</expiration> /xml; <rtoken>value</rtoken> charset <expiration>value</expiration> =UTF-8 <scope>value</scope> <resource_owner>. value< /resource_owner> <client_key>value< /client_key> <client_name>value< /client_name> <status>value</status> <created>value</created> <custom>URL-Encoded-JSON-structure< /custom> </value> </values>	An XML response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
	Response				status: {"values": { "value": [ { "index": 1, 200 "client_key": "value", "scope": "value", content "rtoken": "value", "created": -type: 1234567890, "status": "value", applica "expiration": 1234567890, "token": tion "value", "resource_owner": "value", /json; "secret": "value", "rexpiration": charset 1234567890, "client_name": "value", =UTF-8 "custom": "URL-Encoded-JSON- Structure" } ]} }	A JSON response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
	Error- Response			x-ca- err: 410910 3 Pragma : no- cache Cache- Control : no- store status:	{ "error": "invalid_request", "error_description": "Missing or 410910 unknown parameters" }	If any required parameters or headers are missing, the request will fail.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params	Comment
					400 content -type: application /json; charset =UTF-8	
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The path 410930 component of the URL is invalid" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: application /json; charset =UTF-8	The path component is invalid.
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"SSL with client 410020 authentication is required" } 4 Pragma : no- cache Cache- Control : no- store status: 403 content -type: application /json; charset =UTF-8	Authentication failed.

ID	Operation	URL-Path	HTTP Method	HTTP Headers	Query Params (for GET attach params to URL-PATH)	Comment
Error-Response				x-ca-err: 410020 5 Pragma : no- cache Cache- Control : no- store status: 401 content -type: applica tion /json; charset =UTF-8	{ "error":"invalid_request", "error_description":"The client certificate is not valid" }	The client could not be authenticated.
Error-Response				Allow: GET Pragma : no- cache Cache- Control : no- store status: 405	Allow : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.	The HTTP method is not valid
Error-Response				x-ca-err: 410900 0 Pragma : no- cache Cache- Control : no- store status: 500		Unknown error.
<b>to Get the values of ke tokens n selected by value. —</b>	GET			content token=<token>& -type: applica rtoken=<rtoken>& tion/x- resource_owner=<resource_owner>& www- client_key=<client_key>&		Authentication is done via ssl mutual authentication. Only one of the parameters can and

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
p ar a m		/oau th /tok enst ore /get		form- urlenco ded	status=<status>& offset=0& format=xml	<p>must be used in addition to 'offset' and 'format'.</p> <p><b>token</b> (optional) : the oauth_token or access_token to be retrieved. 'offset' is ignored.</p> <p><b>rtoken</b> (optional) : the refresh_token to be retrieved. 'offset' is ignored.</p> <p><b>resource_owner</b>(optional) : all tokens that have been granted by this resource_owner.</p> <p><b>client_key</b>(optional) : the client_key of which all long-living tokens should be retrieved.</p> <p><b>status</b> (optional) : get all tokens with the given status.</p> <p><b>offset</b> (optional) : The API returns up to 100 values. This parameter is the query offset which can be used for pagination purposes.</p> <p><b>format</b> (optional) : Either 'xml' or 'json'. The response is returned in the requested format.</p>
Response					<pre>status: &lt;values xmlns="http://ns.l7tech.com 200   /2012/11/otk-tokenstore"&gt; &lt;value content index="integer"&gt; &lt;token&gt;value&lt;/token&gt; -type: &lt;secret&gt;value&lt;/secret&gt; text   &lt;expiration&gt;value&lt;/expiration&gt; /xml;  &lt;rtoken&gt;value&lt;/rtoken&gt; charset &lt;expiration&gt;value&lt;/expiration&gt; =UTF-8 &lt;scope&gt;value&lt;/scope&gt; &lt;resource_owner&gt;. value&lt; /resource_owner&gt; &lt;client_key&gt;value&lt; /client_key&gt; &lt;client_name&gt;value&lt;</pre>	An XML response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					/client_name> <status>value</status><created>value</created><custom>URL-Encoded-JSON-structure</custom> </value> </values>	
Response					status: {"values": { "value": [ { "index": 1, 200 "client_key": "value", "scope": "value", content "rtoken": "value", "created": "-type: 1234567890, "status": "value", applica "expiration": 1234567890, "token": tion "value", "resource_owner": "value", /json; "secret": "value", "rexpiration": charset 1234567890, "client_name": "value", =UTF-8 "custom": {} } ]}}	A JSON response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
Error-Response		x-ca-err:	3	Pragma	{ "error": "invalid_request", "error_description": "Missing or 410910 unknown parameters" }	If any required parameters or headers are missing, the request will fail.
Error-Response		x-ca-err:	3	Pragma	{ "error": "invalid_request", "error_description": "The path 410930 component of the URL is invalid" }	The path component is invalid.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error- Response					tion /json; charset =UTF-8	
Error- Response				x-ca-err:	{ "error":"invalid_request", "error_description":"SSL with client 410020 authentication is required" } 4 Pragma : no- cache Cache- Control : no- store status: 403 content -type: applica tion /json; charset =UTF-8	Authentication failed.
Error- Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The client 410020 certificate is not valid" } 5 Pragma : no- cache Cache- Control : no- store status: 401 content -type: applica tion /json; charset =UTF-8	The client could not be authenticated.
Error- Response				Allow: GET Pragma : no- cache		The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org">RFC 2616</a> ( <a href="https://tools.ietf.org">https://tools.ietf.org</a> )

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
				Cache-Control : no-store status: 405		/html/rfc2616) and contains a comma separated list of valid HTTP methods.
	Error-Response			x-ca-err: 410900 0 Pragma : no-cache Cache-Control : no-store status: 500		Unknown error.
get_oauth_tokens_by_client_id_and_resource_owner.	Get the values of tokens selected by client_id and resource_owner.	/oau/th/tokens/enst/ore/get	GET	content-type: application/x-www-form-urlencoded	resource_owner=<resource_owner>&client_key=<client_key>&format=xml	Authentication is done via ssl mutual authentication.
	Response			status: 200	<values xmlns="http://ns.l7tech.com/2012/11/otk-tokenstore"> <value content-index="integer"> <token>value</token> <secret>value</secret> <text>value</text> <expiration>value</expiration> <rtoken>value</rtoken> < charset="UTF-8" /> <scope>value</scope> <resource_owner>.value</resource_owner> <client_key>value</client_key> <client_name>value</client_name>	resource_owner : all tokens that have been granted by this resource_owner. client_key : the client_key of which all long-living tokens should be retrieved. format (optional) : Either 'xml' or 'json'. The response is returned in the requested format.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params	Comment
					/client_name> <status>value</status><created>value</created><custom>URL-Encoded-JSON-structure</custom> </value> </values>	
Response					<pre>status: {"values": { "value": [ { "index": 1, "client_key": "value", "scope": "value", "content": "rtoken": "value", "created": "-type": 1234567890, "status": "value", "application": "expiration": 1234567890, "token": "value", "resource_owner": "value", "/json; "secret": "value", "expiration": "charset": 1234567890, "client_name": "value", "encoding": "UTF-8 "custom": "URL-Encoded-JSON-Structure"} ]} }</pre>	A JSON response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
Error-Response					<pre>x-ca-err: { "error": "invalid_request", "error_description": "Missing or 410910 unknown parameters" } 3</pre> <p>Pragma : no-cache Cache-Control : no-store status: 400 content -type: application /json; charset =UTF-8</p>	If any required parameters or headers are missing, the request will fail.
Error-Response					<pre>x-ca-err: { "error": "invalid_request", "error_description": "The path 410930 component of the URL is invalid" } 3</pre> <p>Pragma : no-cache Cache-Control : no-store status: 400 content -type:</p>	The path component is invalid.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					application/json; charset=UTF-8	
Error-Response				x-ca-err: 4	{ "error":"invalid_request", "error_description":"SSL with client 410020 authentication is required" }	Authentication failed.
Error-Response				x-ca-err: 5	{ "error":"invalid_request", "error_description":"The client 410020 certificate is not valid" }	The client could not be authenticated.
Error-Response				Allow: GET		The HTTP method is not valid
				Pragma: no-		<b>Allow</b> : This header is required by <a href="#">RFC 2616</a>

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				cache Cache- Control : no- store status: 405		( <a href="https://tools.ietf.org/html/fc2616">https://tools.ietf.org/html/fc2616</a> ) and contains a comma separated list of valid HTTP methods.
	Error- Response			x-ca- err: 410900 0 Pragma : no- cache Cache- Control : no- store status: 500		Unknown error.
<a href="#">get_oauth_to_ke_n_s_ta_tu_s_</a>	<a href="#">Get the values of selected by token status and resource_o</a>	/oau /tok /get	GET	content-type: application/x-www-form-urlencoded	resource_owner=<resource_owner>&-type:&status=<status>&offset=0&format=xml	Authentication is done via ssl mutual authentication.  <b>resource_owner</b> : all tokens that have been granted by this resource_owner. <b>status</b> : get all tokens with the given status. <b>offset</b> (optional) : The API returns up to 100 values. This parameter is the query offset which can be used for pagination purposes. <b>format</b> (optional) : Either 'xml' or 'json'. The response is returned in the requested format.
	Response				status: <values xmlns=" <a href="http://ns.l7tech.com/2012/11/otk-tokenstore">http://ns.l7tech.com/2012/11/otk-tokenstore</a> "> <value index="integer"> <token>value</token> <secret>value</secret> <expiration>value</expiration> <rtoken>value</rtoken> <rexpiration>value</rexpiration>	An XML response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					text <scope>value</scope> /xml; <resource_owner>.value< charset /resource_owner><client_key>value< =UTF-8 /client_key><client_name>value< /client_name><status>value</status> <created>value</created> <custom>URL-Encoded-JSON-structure< /custom> </value> </values>	
Response					status: {"values": { "value": [ { "index": 1, 200 "client_key": "value", "scope": "value", content "rtoken": "value", "created":: -type: 1234567890, "status": "value", applica "expiration": 1234567890, "token":: tion "value", "resource_owner": "value", /json; "secret": "value", "rexpiration": charset 1234567890, "client_name": "value" =UTF-8 "custom":"{URL-Encoded-JSON- Structure}" } ]} }	A JSON response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
Error-Response					x-ca- { "error":"invalid_request", err: "error_description":"Missing or 410910 unknown parameters" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	If any required parameters or headers are missing, the request will fail.
Error-Response					x-ca- { "error":"invalid_request", err: "error_description":"The path 410930 component of the URL is invalid" } 3 Pragma : no- cache Cache- Control : no- store	The path component is invalid.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params	Comment
					status: 400 content -type: application /json; charset =UTF-8	
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"SSL with client 410020 authentication is required" } 4 Pragma : no- cache Cache- Control : no- store status: 403 content -type: application /json; charset =UTF-8	Authentication failed.
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The client 410020 certificate is not valid" } 5 Pragma : no- cache Cache- Control : no- store status: 401 content -type: application /json; charset =UTF-8	The client could not be authenticated.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
	Error-Response			Allow: GET Pragma : no- cache Cache- Control : no- store status: 405		The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
	Error-Response			x-ca- err: 410900 0 Pragma : no- cache Cache- Control : no- store status: 500		Unknown error.
get_temporar_y_t	Get the temporary tokens selected by a token.	/oau /tok /get	GET	content-type: application/x-www-form-urlencoded	token= <token>& format=xml	Authentication is done via ssl mutual authentication.  <b>token</b> : the request_token (OAuth 1.0) or authorization_code (OAuth 2.0) to be retrieved. <b>format</b> (optional) : Either 'xml' or 'json'. The response is returned in the requested format.
	Response			status: 200	<values xmlns="http://ns.l7tech.com/2012/11/otk-tokenstore"> <value content-index="integer"> <token>value</token> <secret>value</secret> <text>value</text> <expiration>value</expiration> <xml>value</xml> <scope>value</scope> <resource-owner>value</resource-owner> <client-key>value</client-key> <client-name>value</client-name>	An XML response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					/client_name> <created>value</created> <callback>value</callback> <verifier>value</verifier> <custom>URL-Encoded-JSON-structure</custom> </value> </values>	
	Response				status: {"values": { "value": [ { "index": 1, "client_key": "value", "scope": "value", "content": "created": 1234567890, "expiration": "-type": 1234567890, "token": "value", "application": "callback": "value", "verifier": "value", "resource_owner": "value", "secret": "/json; "value": "value", "client_name": "value", "charset": "custom": "URL-Encoded-JSON- =UTF-8 Structure"} ] } }	A JSON response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
	Error-Response				x-ca-err: { "error": "invalid_request", "error_description": "Missing or 411010 unknown parameters" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	If any required parameters or headers are missing, the request will fail.
	Error-Response				x-ca-err: { "error": "invalid_request", "error_description": "The path 411030 component of the URL is invalid" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type:	The path component is invalid.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
Error-Response					application/json; charset=UTF-8	
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"SSL with client 410020 authentication is required" }  4 Pragma : no- cache Cache- Control : no- store status: 403 content -type: application/json; charset=UTF-8	Authentication failed.
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The client 410020 certificate is not valid" }  5 Pragma : no- cache Cache- Control : no- store status: 401 content -type: application/json; charset=UTF-8	The client could not be authenticated.
Error-Response				Allow: GET Pragma : no-		The HTTP method is not valid <b>Allow</b> : This header is required by <a href="#">RFC 2616</a>

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				cache Cache- Control : no- store status: 405		( <a href="https://tools.ietf.org/html/fc2616">https://tools.ietf.org/html/fc2616</a> ) and contains a comma separated list of valid HTTP methods.
	Error- Response			x-ca- err: 411000 0 Pragma : no- cache Cache- Control : no- store status: 500		Unknown error.
get_temporar	Get the values of temporary tokens selected by token and optional verifier. This is valid in the context of OAuth 1.0	/oau /tok /tokens /selected /by token /and /optional /verifier.	GET	content token=<token>&-type: application/x-www-form-urlencoded	content token=<token>&-type: application/x-www-form-urlencoded	Authentication is done via ssl mutual authentication.
	Response			status: <values xmlns=" <a href="http://ns.l7tech.com/2012/11/otk-tokenstore">http://ns.l7tech.com/2012/11/otk-tokenstore</a> "> <value content index="integer"> <token>value</token>	status: <values xmlns=" <a href="http://ns.l7tech.com/2012/11/otk-tokenstore">http://ns.l7tech.com/2012/11/otk-tokenstore</a> "> <value content index="integer"> <token>value</token>	An XML response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
	Response			-type: <secret>value</secret> text <expiration>value</expiration> /xml; <scope>value</scope> charset <resource_owner>.value<=UTF-8	-type: <secret>value</secret> text <expiration>value</expiration> /xml; <scope>value</scope> charset <resource_owner>.value<=UTF-8	

ID	Operation	URL-Path	HTTP Method	HTTP Headers	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					<pre> status: {"values": { "value": [ { "index": 1, 200   "client_key": "value", "scope": "value", content "created": 1234567890, "expiration": -type: 1234567890, "token": "value", applica "callback": "value", "verifier": "value", tion   "resource_owner": "value", "secret": /json;   "value", "client_name": "value", charset "custom":"{URL-Encoded-JSON- = UTF-8 Structure}" } ]} } </pre>	A JSON response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
Error-Response					<pre> x-ca- { "error":"invalid_request", err:   "error_description":"Missing or 411010 unknown parameters" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8 </pre>	If any required parameters or headers are missing, the request will fail.
Error-Response					<pre> x-ca- { "error":"invalid_request", err:   "error_description":"The path 411030 component of the URL is invalid" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8 </pre>	The path component is invalid.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
Error-Response				x-ca-err: 4	{ "error":"invalid_request", "error_description":"SSL with client authentication is required" }  Pragma : no- cache Cache- Control : no- store status: 403 content -type: applica tion /json; charset =UTF-8	Authentication failed.
Error-Response				x-ca-err: 5	{ "error":"invalid_request", "error_description":"The client certificate is not valid" }  Pragma : no- cache Cache- Control : no- store status: 401 content -type: applica tion /json; charset =UTF-8	The client could not be authenticated.
Error-Response				Allow: GET Pragma : no- cache Cache- Control		The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org /html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.

ID	Operation	URL- Path	HTTP Method	HTTP Header	Body/ Query Params	Comment
					(for GET attach params to URL-PATH)	
					: no-store status: 405	
	Error-Response		x-ca-err:	411000	0	Unknown error.
			Pragma		: no-cache	
			Cache-Control		: no-store	
					status: 500	
<u>ar</u>	<u>Get the values of temporary tokens selected by client_id and resource_owner.</u>	/oau	GET	content-type: application/x-www-form-urlencoded	client_key=<client_key>&resource_owner=<resource_owner>&	Authentication is done via ssl mutual authentication.
<u>y</u>		th				This request only returns a result if any temporary token issued has not been used. Usually this table is empty since clients exchange the temporary token for a long living token immediately.
<u>t</u>		/tok				
<u>ok</u>		enst				
<u>e</u>		ore				
<u>n</u>		/get				
<u>c</u>		Tem				
<u>id</u>		urlenco				
<u>r</u>		ded				
<u>o</u>						
						client_key(optional) : the oauth_consumer_key (OAuth 1.0) or client_id (OAuth 2.0) of which all temporary tokens should be retrieved. This can be used without 'resource_owner'. resource_owner(optional) : all tokens that

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
						have been granted by this resource_owner. This can be used without 'client_key'. <b>format</b> (optional) : Either 'xml' or 'json'. The response is returned in the requested format.
Response					<pre> status: &lt;values xmlns="http://ns.l7tech.com 200   /2012/11/otk-tokenstore"&gt; &lt;value content index="integer"&gt; &lt;token&gt;value&lt;/token&gt; -type: &lt;secret&gt;value&lt;/secret&gt; text   &lt;expiration&gt;value&lt;/expiration&gt; /xml;  &lt;scope&gt;value&lt;/scope&gt; charset &lt;resource_owner&gt;. value&lt; =UTF-8 /resource_owner&gt; &lt;client_key&gt;value&lt; /&gt;&lt;client_key&gt; &lt;client_name&gt;value&lt; /&gt;&lt;client_name&gt; &lt;created&gt;value&lt; /&gt;&lt;created&gt; &lt;callback&gt;value&lt;/callback&gt; &lt;verifier&gt;value&lt;/verifier&gt; &lt;custom&gt;URL- Encoded-JSON-structure&lt;/custom&gt; &lt; /&gt;&lt;/value&gt; &lt;/values&gt;</pre>	An XML response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
Response					<pre> status: {"values": { "value": [ { "index": 1, 200   "client_key": "value", "scope": "value", content "created": 1234567890, "expiration": -type: 1234567890, "token": "value", appli "callback": "value", "verifier": "value", tion   "resource_owner": "value", "secret": "value", /json;  "value", "client_name": "value", charset "custom": "{URL-Encoded-JSON- =UTF-8 Structure}" } ]} }</pre>	A JSON response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
Error-Response					<pre> x-ca- { "error":"invalid_request", err:   "error_description":"Missing or 411010 unknown parameters" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: appli cation</pre>	If any required parameters or headers are missing, the request will fail.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					/json; charset =UTF-8	
Error-Response		x-ca-err: 411030 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	{ "error":"invalid_request", "error_description":"The path component of the URL is invalid" } 411030 component of the URL is invalid" 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: applica tion /json; charset =UTF-8	The path component is invalid.		
Error-Response		x-ca-err: 410020 4 Pragma : no- cache Cache- Control : no- store status: 403 content -type: applica tion /json; charset =UTF-8	{ "error":"invalid_request", "error_description":"SSL with client authentication is required" } 410020 authentication is required" 4 Pragma : no- cache Cache- Control : no- store status: 403 content -type: applica tion /json; charset =UTF-8	Authentication failed.		
Error-Response		x-ca-err: 410020 5 Pragma : no-	{ "error":"invalid_request", "error_description":"The client certificate is not valid" } 410020 certificate is not valid" 5 Pragma : no-	The client could not be authenticated.		

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params	Comment
					cache Cache- Control : no- store status: 401 content -type: applica tion /json; charset =UTF-8	
	Error- Response				Allow: GET Pragma : no- cache Cache- Control : no- store status: 405	The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
	Error- Response				x-ca- err: 411000 0 Pragma : no- cache Cache- Control : no- store status: 500	Unknown error.
t	<b>Registers a new JWT on the OAuth server.</b>	/oau POST			content resource_owner=<resource_owner>& -type: (id_token) /tok on the enst OAuth ore server. /jwt _regi ster	Authentication is done via ssl mutual authentication. The API will replace /update an existing JWT if the combination of salt=<salt>& sub=<sub>& urlenco ded shared_secret=<shared_secret>& shared_secret_type=<shared_secret_type>

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
						'resource_owner - azp' already exists in the case of an id_token.
						<b>resource_owner</b> : the resource_owner for whom this token was issued. <b>azp</b> : a single value or space-separated list of valid users for this token. <b>iss</b> : the issuing server URL, such as: <a href="http://server.ca.com">server.ca.com</a> ( <a href="http://server.ca.com">http://server.ca.com</a> ). <b>jwt</b> : the complete JWT. <b>sub</b> (optional) : the 'subject' if the JWT is an id_token. <b>salt</b> (optional) : the salt value used to generate the 'subject' value if the JWT is an id_token. <b>shared_secret</b> (optional) : the shared secret used to sign the JWT. <b>shared_secret_type</b> (optional) : the shared secret type used to sign the JWT. For example, the algorithm that was used: 'HS256'.
Response					status: persisted 200 content -type: text /plain; charset =UTF-8	
Error-Response		x-ca-err:			{ "error":"invalid_request", "error_description":"Missing or 411110 unknown parameters" } 3 Pragma	If any required parameters or headers are missing, the request will fail.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params	Comment
					: no-cache Cache-Control : no-store status: 400 content-type: application/json; charset=UTF-8	
Error-Response				x-ca-err: 3	{ "error": "invalid_request", "error_description": "The path component of the URL is invalid" }	The path component is invalid.
Error-Response				x-ca-err: 4	{ "error": "invalid_request", "error_description": "SSL with client authentication is required" }	Authentication failed.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					application/json; charset=UTF-8	
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The client certificate is not valid" }	The client could not be authenticated.
Error-Response				5	Pragma: no-cache	
Error-Response				Cache-Control: no-store	Cache-Content-Type: application/json; charset=UTF-8	
Error-Response				status: 401	Allow: POST	The HTTP method is not valid
Error-Response				status: 405	Pragma: no-cache	<b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
Error-Response				x-ca-err: 411100	0	Unknown error.
				Pragma: no-cache	Cache-Control: no-store	
				status: 500		

ID	Operation	URL-Path	HTTP Method	HTTP Header	Query Params (for GET attach params to URL-PATH)	Comment
lookup_jwt	Looks up a JWT (id_token) from the OAuth server.	/oau/th/tok/ens/store/jwt/_lookup	GET	content-type: application/x-www-form-urlencoded	jwt=<jwt>&azp=<azp>&sub=<sub>&resource_owner=<resource_owner>&format=<format>	<p>Authentication is done via ssl mutual authentication. The API will return a JWT based on given parameters.</p> <p><b>jwt</b> (optional) : the complete JWT. This parameter has to be used o its own if it is used!</p> <p><b>azp</b> (optional) : a single or space separated list of valid users for this token. This parameter MUST be used with 'sub' or 'resource_owner'.</p> <p><b>sub</b> (optional) : the 'sub'ject if the JWT is an id_token. This parameter MUST be used with 'azp'</p> <p><b>resource_owner</b>(optional) : the resource_owner for whom this token was issued. This parameter MUST be used with 'azp'</p> <p><b>format</b> (optional) : Either 'xml' or 'json'. The response is returned in the requested format.</p>
Response					status: <values xmlns="http://ns.l7tech.com/2012/11/otk-tokenstore"> <value content-index="integer"> <type: <resource_owner>value<text /></resource_owner> <azp>value</azp></xml> <iss>value</iss> <expiration>value<charset /><expiration> <jwt>value</jwt> =UTF-8 <sub>value</sub> <jwt_id>value</jwt_id> <salt>value</salt> <shared_secret>value</shared_secret> <shared_secret_type>value</shared_secret_type> </value> </values>	An XML response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).

ID	Operation	URL-Path	HTTP Method	HTTP Headers	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
	Response				<pre>status: {"values": { "value": { "sub": "value", 200   "resource_owner": "value", "salt": content "value", "shared_secret_type": "value", -type: "azp": "value", "jwt": "value", "jwt_id": "text " value", "iss": "value", "index": integer, /xml;   "shared_secret": "value", "expiration": "charset value } }}} =UTF-8</pre>	A JSON response that includes the element 'values'. 'value' appears once, multiple times, or not at all (indicating no values found).
	Error-Response				<pre>x-ca- { "error":"invalid_request", err:   "error_description":"Missing or 411210 unknown parameters" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type:  applica tion /json; charset =UTF-8</pre>	If any required parameters or headers are missing, the request will fail.
	Error-Response				<pre>x-ca- { "error":"invalid_request", err:   "error_description":"The path 411230 component of the URL is invalid" } 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type:  applica tion /json; charset =UTF-8</pre>	The path component is invalid.
						Authentication failed.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
Error-Response				x-ca-err: 4	{ "error":"invalid_request", "error_description":"SSL with client authentication is required" }  Pragma : no-cache Cache-Control : no-store status: 403 content-type: application/json; charset=UTF-8	
Error-Response				x-ca-err: 5	{ "error":"invalid_request", "error_description":"The client certificate is not valid" }  Pragma : no-cache Cache-Control : no-store status: 401 content-type: application/json; charset=UTF-8	The client could not be authenticated.
Error-Response				Allow: GET		The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.

ID	Operation	URL- Path	HTTP Method	HTTP Header	Query Params (for GET attach params to URL-PATH)	Comment
				: no-store status: 405		
	Error-Response		x-ca-err: 411200 0 Pragma : no-cache Cache-Control : no-store status: 500			Unknown error.
remove_jwt		Removes a JWT from the OAuth server.	/oau /tok enst ore /jwt _re mov al	DELETE -type: aplica tion/x- www- form- urlenco ded	content jwt=<jwt>& azp=<azp>& logout_apps=false	<p>Authentication is done via ssl mutual authentication.</p> <p>Only one of the parameters can be and has to be used in addition to 'logout_apps'.</p> <p><b>jwt</b> (optional) : the complete JWT.</p> <p><b>azp</b> (optional) : the azp value.</p> <p><b>logout_apps</b> (optional) : [MAG]: if set to 'true', all oauth token issued to 'azp' will be removed. This parameter is valid in the context of MAG and Mobile SSO only!</p>
	Response				status: removed 200 content -type: text /plain	
	Error-Response		x-ca-err: 411310 3 Pragma	{ "error":"invalid_request", "error_description":"Missing or unknown parameters" }		If any required parameters or headers are missing, the request will fail.

ID	Operation	URL-Path	HTTP Method	HTTP Header	Query Params	Comment
				: no-cache Cache-Control : no-store status: 400 content-type: application/json; charset=UTF-8		
Error-Response			x-ca-err: 3	{ "error": "invalid_request", "error_description": "The path component of the URL is invalid" }		The path component is invalid.
Error-Response			x-ca-err: 4	{ "error": "invalid_request", "error_description": "SSL with client authentication is required" }		Authentication failed.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					application/json; charset=UTF-8	
Error-Response				x-ca-err:	{ "error":"invalid_request", "error_description":"The client certificate is not valid" }	The client could not be authenticated.
Error-Response				5	Pragma: no-cache	
Error-Response				Cache-Control: no-store	Cache-Content-Type: application/json; charset=UTF-8	
Error-Response				status: 401	Allow: DELETE	The HTTP method is not valid
Error-Response				status: 405	Pragma: no-cache	<b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
Error-Response				status: 500	x-ca-err: 411300	Unknown error.
				0	Pragma: no-cache	
				Cache-Control: no-store	Cache-Content-Type: application/json; charset=UTF-8	
				status: 500	Allow: DELETE	The HTTP method is not valid

## Register a Token

Registers a temporary token.

---

```
/store?
token=<value>&expiration=<value>&client_key=<value>&client_name=<value>&callback=<value>&scope=<value>(&resource_owner=<value>)?(&secret=<value>)?
```

**token:** the token, either a *request\_token* (OAuth 1.0) or an *authorization\_code* (OAuth 2.0) to be stored

**expiration:** the date this token expires (the OAuth Toolkit is using ms)

**client\_key:** the *client\_key* issued for this token

**client\_name:** the client issued for this token

**callback:** the callback uri. "redirect\_uri" in OAuth 2.0

**scope:** the scope used with OAuth 2.0

**resource\_owner:** the *resource\_owner* who granted this token. This is optional because the initial *OAuth token* was not granted when it was issued

**secret:** the secret for this token. This is optional because OAuth 2.0 does not have one.

---

Registers access\_tokens used with OAuth 1.0

---

```
/store?
token=<value>&secret=<value>&expiration=<value>&client_key=<value>&status=<value>&temp_token=<value>?
```

**token:** the *access\_token*

**secret:** the shared secret for this token

**expiration:** the date this token expires (the OAuth Toolkit is using ms)

**client\_key:** the *client\_key* issued for this token

**status:** either ENABLED or DISABLED. DISABLED tokens will cause a request to fail

**temp\_token:** the temporary token exchanged for this token

---

Registers access\_token used with OAuth 2.0.

---

```
store?
token=<value>&expiration=<value>&client_key=<value>&resource_owner=<value>&status=<value>&temp_token=<value>(&rtoken=<value>&rexpiration=<value>)?
```

**token:** the *access\_token*

**expiration:** expiration: the date the token expires (the OAuth Toolkit is using ms)

**client\_key:** the *client\_key* issued for this token

---

---

```
store?
token=<value>&expiration=<value>&client_key=<value>&resource_owner=<value>&status=<value>&s
(&rtoken=<value>&rexpiration=<value>)?
```

**resource\_owner:** the *resource\_owner* who granted this token

**status:** either ENABLED or DISABLED. DISABLED tokens will cause a request to fail

**scope:** the scope

**client\_name:** the client name issued for this token

**secret:** the secret for this token. This is optional because it is only used for token type "MAC"

**rtoken:** the refresh token is available

**rexpiration:** the refresh token expiration date (if available)

---

**Response:**

- status: 200, content-type: text/plain, body: persisted
- status: 400, content-type: text/plain, body: Token could not be persisted

## Update a Token



You can also update the status of an access token by submitting the parameters "status" and "token".

---

### APIs

---

/update?token=<value>&resource\_owner=<value>

---

/update?token=<value>&expiration=<value>&verifier=<value>

---

### Parameters

**token:** the token to be updated. This is a temporary token (*request\_token*).

**resource\_owner:** the *resource\_owner* assigned to the token

**expiration:** when the token expires (in milliseconds)

**verifier:** the verifier assigned to this token

---

### Response:

- status: 200, content-type: text/plain, body: {no-of-tokens} token(s) updated
- status: 400, content-type: text/plain, body: Tokens could not be updated

## Revoke a Token

This API works for long-living tokens only.

### APIs

---

```
/revoke?client_key=<value>&resource_owner=<value>
```

---

```
/revoke?client_key=<value>
```

---

```
/revoke?resource_owner=<value>
```

---

```
/revoke?token=<value>
```

---

### Parameters

**client\_key:** revokes the token of this *client\_key*

**resource\_owner:** the resource owner who granted the token to be revoked

**token:** the token to be revoked

---

### Response:

- status: 200, content-type: text/plain, body: {no-of-tokens} token(s) revoked
- status: 400, content-type: text/plain, body: Token could not be revoked

## Delete a Token

---

```
/delete?temp_token=<value>/token=<value>/rtoken=<value>/client_key=<value>
```

---

Only one of these parameters are allowed at once.

**temp\_token:** the temporary token to delete

**token:** the access\_token to delete. This will include an assigned refresh\_token

**rtoken:** the refresh token to delete. This will include an assigned access\_token

**client\_key:** all tokens issued for this *client\_key*, includes temporary and long living tokens

---

### Response:

- status: 200, content-type: text/plain, body: {no-of-tokens} token(s) deleted
- status: 400, content-type: text/plain, body: Token could not be deleted

## Retrieve Token Values

These APIs will use the given parameters as search parameter to select only certain tokens. All of these APIs accept the additional parameter "format". "Format" can either take the value "xml" (which is the default) or "json". If it is "json", the values will be returned as a json message.

---

APIs`/get``/get?token=<value>/rtoken=<value>/resource_owner=<value>/client_key=<value>/status=<value>``/get?client_key=<value>&resource_owner=<value>``/get?token=<value>/rtoken=<value>/resource_owner=<value>/client_key=<value>/status=<value>``/get?client_key=<value>&resource_owner=<value>`

---

When no parameters are specified, all values of all long-living tokens will be returned.

---

Parameters

**token:** all values of the given token (which is an *access\_token*)

**rtoken:** all token values of the given *refresh\_token*

**resource\_owner:** all token values the given *resource\_owner* has granted

**client\_key:** all token values of tokens issued for the given *client\_key*

**status:** all token values according to the given status (ENABLED | DISABLED)

---

Response:

- status: 200, content-type: text/xml, body
- status: 200, content-type: application/json, body
- status: 400, content-type: text/plain, body: Non-valid parameter list for this operation

## Retrieve Temporary Token Values

These APIs will use the given parameters as search parameter to select only certain temporary tokens. All of these APIs accept the additional parameter "format". "Format" can either take the value "xml" (which is the default) or "json". If it is "json", the values will be returned as a json message.

---

APIs`/getTemp``/getTemp?token=<value>/resource_owner=<value>/client_key=<value>``getTemp?token=<value>&verifier=<value>/client_key=<value>&resource_owner=<value>`

---

When no parameters are specified, all values of all temporary tokens are returned.

---

Parameters

**token:** all token values of this temporary token

**Parameters****resource\_owner:** all token values of temporary tokens granted by this *resource\_owner***client\_key:** all temporary token values issued for given *client\_key***Response:**

- status: 200, content-type: text/xml, body
- status: 200, content-type: application/json, body
- status: 400, content-type: text/plain, body: Non-valid parameter list for this operation

## Sessionstore API

This API allows you to manage storage. It provides access to a generic interface used to persist and cache short lived data. The API is disabled by default.

All endpoints described will start here:

*<Gateway\_host\_and\_port>/oauth/session/\**

For example:

<http://my.securespan.gateway.com:8080/oauth/session/store>

- [create\\_session \(see page 348\)](#)
- [get\\_session \(see page 351\)](#)
- [delete\\_session \(see page 353\)](#)
- [delete\\_expired\\_session \(see page 355\)](#)

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
<b>Creates a new session. The data is persisted but also cached to increase performance.</b>	<b>/oauth/session</b>	/oaut h /sessi on /stor e	POST	content- type: application/x-www-form-urlencoded	cacheKey=<cacheKey>& value=<value>& cacheAge=360000& cacheId=defaultCache& cacheMaxEntries=cacheMaxEntries& cacheMaxSize=100000& dbAge=360000	<p>Authentication is done via ssl mutual authentication.</p> <p>This API updates an existing session of the combination 'cacheKey - cacheId'.</p> <p><b>cacheKey</b> : The unique identifier of this session.</p> <p><b>value</b> : The value to be cached for this session. It will be URLEncoded</p> <p><b>cacheAge</b>(optional) : The maximum cache lifetime of the value in seconds.</p> <p><b>cacheId</b> (optional) : The id</p>

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
						<p>to identify this session store, e.g.: myAppCache.</p> <p><b>cacheMaxEntries</b>(optional) : The maximum number of cached entries for this cache.</p> <p><b>cacheMaxSize</b>(optional) : The maximum size in bytes per session value.</p> <p><b>dbAge</b> (optional) : The maximum persistent lifetime of the value in seconds.</p>
	Response				status: 200 content-type: text/plain; charset= UTF-8	Key \${cacheKey} added to session.
	Error-Response				x-ca-err: { "error": "4200103 invalid_request", "Pragma": "error_description": "no-cache Missing or unknown Cache- parameters" } Control: no-store status: 400 content-type: application/json; charset= UTF-8	If any required parameters or headers are missing, the request will fail.
	Error-Response				x-ca-err: { "error": "4200303 invalid_request", "Pragma": "error_description": "The no-cache path component of the Cache- URL is invalid" } Control: no-store status: 400 content-type:	The path component is invalid.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					application/json; charset=UTF-8	
	Error-Response				x-ca-err: { "error": "4200204 invalid_request", Pragma: "error_description": "SSL no-cache with client authentication Cache- is required" } Control: no-store status: 403 content-type: application/json; charset= UTF-8	Authentication failed.
	Error-Response				x-ca-err: { "error": "4200205 invalid_request", Pragma: "error_description": "The no-cache client certificate is not Cache- valid" } Control: no-store status: 401 content-type: application/json; charset= UTF-8	The client could not be authenticated.
	Error-Response				Allow: POST, PUT Pragma: no-cache Cache- Control: no-store status: 405	The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616 (https://tools.ietf.org/html/rfc2616)</a> and contains a comma separated list of valid HTTP methods.
	Error-Response				x-ca-err: 4200000 Pragma: no-cache	Unknown error.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					Cache-Control: no-store status: 500	
get_session	<b>Get a session.</b>	/oauth h /sessi on /get	GET	content-type: application/x-www-form-urlencoded	cacheKey=<cacheKey>& cacheId=defaultCache& cacheAge=360000	<p>Authentication is done via ssl mutual authentication.</p> <p><b>cacheKey</b> : The unique identifier of this session.</p> <p><b>cacheId</b> (optional) : The id to identify this session store, e.g.: myAppCache.ca</p> <p><b>cacheAge</b>(optional) : The maximum cache lifetime of the value in seconds.</p>
	Response				status: 200 content-type: text/xml	The session value was found in the cache. 'value' and 'location' will not appear if no data was found
	Response				status: 200 content-type: text/xml	The session value was found in the database. 'value' and 'location' will not appear if no data was found
	Error-Response				x-ca-err: { "error": "4201103 invalid_request", Pragma: "error_description": "no-cache Missing or unknown Cache-parameters" } Control: no-store status: 400 content-type: application/json; charset= UTF-8	If any required parameters or headers are missing, the request will fail.
	Error-Response					The path component is invalid.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					x-ca-err: { "error":"4201303 invalid_request", Pragma: "error_description":"The no-cache path component of the Cache- URL is invalid" } Control: no-store status: 400 content-type: application/json; charset= UTF-8	
	Error-Response				x-ca-err: { "error":"4200204 invalid_request", Pragma: "error_description":"SSL no-cache with client authentication Cache- is required" } Control: no-store status: 403 content-type: application/json; charset= UTF-8	Authentication failed.
	Error-Response				x-ca-err: { "error":"4200205 invalid_request", Pragma: "error_description":"The no-cache client certificate is not Cache- valid" } Control: no-store status: 401 content-type: application/json; charset= UTF-8	The client could not be authenticated.
	Error-Response				Allow: GET Pragma:	The HTTP method is not valid <b>Allow :</b> This header is

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				no-cache Cache- Control: no-store status: 405		required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
	Error-Response			x-ca-err: 4201000 Pragma: no-cache Cache- Control: no-store status: 500		Unknown error.
<a href="#">delete_session</a>	<b>Delete a session.</b>	/oaut	DELETE	content-type: application/x-www-form-urlencoded	cacheKey=<cacheKey>&cacheId=defaultCache	Authentication is done via ssl mutual authentication.
		n		status: 200 content-type: text/plain; charset=UTF-8		<b>cacheKey</b> : The unique identifier of this session. <b>cacheId</b> (optional) : The id to identify this session store, e.g.: myAppCache.
	Response					
	Error-Response			x-ca-err: { "error":"4202103 invalid_request", Pragma: "error_description":"no-cache Missing or unknown Cache- parameters" } Control: no-store status: 400 content-type: application/json; charset=UTF-8		If any required parameters or headers are missing, the request will fail.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
	Error-Response				x-ca-err: { "error":"4202303 invalid_request", Pragma: "error_description":"The no-cache path component of the Cache- URL is invalid" } Control: no-store status: 400 content-type: application/json; charset= UTF-8	The path component is invalid.
	Error-Response				x-ca-err: { "error":"4200204 invalid_request", Pragma: "error_description":"SSL no-cache with client authentication Cache- is required" } Control: no-store status: 403 content-type: application/json; charset= UTF-8	Authentication failed.
	Error-Response				x-ca-err: { "error":"4200205 invalid_request", Pragma: "error_description":"The no-cache client certificate is not Cache- valid" } Control: no-store status: 401 content-type: application/json; charset= UTF-8	The client could not be authenticated.
	Error-Response				Allow: DELETE Pragma:	The HTTP method is not valid <b>Allow :</b> This header is

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				no-cache Cache- Control: no-store status: 405		required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
	Error-Response			x-ca-err: 4202000 Pragma: no-cache Cache- Control: no-store status: 500		Unknown error.
<hr/>						
<a href="#">delete_expire</a>	<b>Deletes expired sessions.</b>	/oaut	DELETE	content-type: application/x-www-form-urlencoded		Authentication is done via ssl mutual authentication. Expired sessions are deleted from the database. This does not include sessions that are in the cache.
<a href="#">d</a>				h		
<a href="#">_s</a>				/sessi		
<a href="#">es</a>				on		
<a href="#">si</a>				/dele		
<a href="#">o</a>				teExp		
<a href="#">n</a>				ired		
	Response				status: deleted 200 content-type: text /plain; charset= UTF-8	
	Error-Response				x-ca-err: { "error": " 4203103 invalid_request", Pragma: "error_description": " no-cache Missing or unknown Cache- parameters" } Control: no-store status: 400 content-type:	If any required parameters or headers are missing, the request will fail.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/Query Params (for GET attach params to URL-PATH)	Comment
					application/json; charset=UTF-8	
	Error-Response				x-ca-err: { "error":"4203303 invalid_request", Pragma: "error_description":"The no-cache path component of the Cache- URL is invalid" } Control: no-store status: 400 content-type: application/json; charset=UTF-8	The path component is invalid.
	Error-Response				x-ca-err: { "error":"4200204 invalid_request", Pragma: "error_description":"SSL no-cache with client authentication Cache- is required" } Control: no-store status: 403 content-type: application/json; charset=UTF-8	Authentication failed.
	Error-Response				x-ca-err: { "error":"4200205 invalid_request", Pragma: "error_description":"The no-cache client certificate is not Cache- valid" } Control: no-store status: 401 content-type:	The client could not be authenticated.

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					application/json; charset=UTF-8	
	Error-Response			Allow: DELETE Pragma: no-cache Cache-Control: no-store status: 405	The HTTP method is not valid <b>Allow</b> : This header is required by <a href="https://tools.ietf.org/html/rfc2616">RFC 2616</a> ( <a href="https://tools.ietf.org/html/rfc2616">https://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.	
	Error-Response			x-ca-err: 4203000 Pragma: no-cache Cache-Control: no-store status: 500	Unknown error.	

## Portal Storage API

The API allows you to synchronize with the SaaS API Portal to manage OAuth clients and client keys (client\_id, oauth\_consumer\_key).

All endpoints described start here:

`<Gateway_host_and_port>/portal/storage`

- [portal\\_sync \(see page 357\)](#)
- [portal\\_lookup\\_apikey \(see page 361\)](#)

ID	Operation	URL-Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
: Create, update or delete API Keys and OAuth Clients	/portal/storage	POST	content { "bulkSync": "false", "deletedIds": [ "3c2acfb5-8803-4c0c-8de3-a9224cad2595", "066f33d1-7e45-4434-be69-5aa7d20934e1" ], "entityType": "APPLICATION", "incrementStart": 1453156182682, "newOrUpdatedEntities": [ { "apis": [ { "id": "7e4535be-76d6-4696-8268-c9226acf792a" } ], "createdBy": null, "custom": "{ \"data\": \"value\" }", "id": "9ecd3853-c9bb-417b-8a69-d75c53aeeb5f", "key": "l7xxab6a89e9598c42809d99debecba8c576", "name": "My OAuth Client", "type": "APPLICATION" } ] }	-type: application/json	-	Authentication is done via ssl mutual authentication.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					<pre>"label": "app1", "mag": { "masterKeys": [ { "environment": "iOS", "masterKey": "a value" }, { "environment": "android", "masterKey": "a value" } ], "redirectUri": "a redirect uri", "scope": "a list of scopes" }, "modifiedBy": null, "oauthCallbackUrl": null, "oauthScope": null, "oauthType": null, "organizationId": "de0d455c-89f9-4602-9494-20f6ae62c5d3", "organizationName": "org2", "secret": "4e5ac053df584ef08e7f5232aed1c26f", "status": "active" } ] }</pre>	
Response					<pre>status: { "portal_sync": { "syncType": "incremental bulk", "deletedIdsError": [], "content": "newOrUpdatedEntitiesError": [] } }</pre> <p>-type: application/json; charset= UTF-8</p>	
Error-Response					<pre>x-ca-err: { "error": "invalid_request", "error_description": "SSL with client authentication is required" }</pre> <p>400520 4 Pragma : no- cache Cache- Control : no- store status: 403 content -type: application/json; charset= UTF-8</p>	Authentication failed.
Error-Response					<pre>x-ca-err: { "error": "invalid_request", "error_description": "The client certificate is not valid" }</pre> <p>400520 5 Pragma : no- cache Cache- Control</p>	The client could not be authenticated.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					: no-store status: 401 content-type: application/json; charset=UTF-8	
Error-Response		x-ca-err:	400530	3	{ "error":"invalid_request", "error_description":"The path component of the URL is invalid" }	The path component is invalid.
Error-Response		x-ca-err:	400530	0	{ "portal_sync": { "syncType":incremental bulk", "deletedIdsError": [a comma separated list of ApiKey ID's that could not be deleted], "newOrUpdatedEntitiesError": [a comma separated list of ApiKey ID's that could not be updated or created], } }	The API Keys and/or OAuth Clients could not be processed.

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
Error-Response		x-ca-err:	{ "error":"invalid_request", "error_description":"Missing or unknown parameters" }	400510 3 Pragma : no-cache Cache-Control : no-store status: 400 content-type: application/json; charset= UTF-8	If the JSON Sync message was missing or could not be processed.
Error-Response		Allow:	POST, PUT	Pragma : no-cache Cache-Control : no-store status: 405	The HTTP method is not valid. <b>Allow</b> : This header is required by <a href="#">RFC 2616</a> ( <a href="http://tools.ietf.org/html/rfc2616">http://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
Error-Response		x-ca-err:	400500 0	Pragma : no-cache Cache-Control : no-store status: 500	Unknown error.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
portal_lookup_by_key	Retrieve an API Key record	/portal/storage/age	GET		content apikey=<apikey>&-type: serviceid=<serviceid>	Authentication is done via ssl mutual authentication.
	Response				<p>status: &lt;values xmlns="http://ns.l7tech.com/2012/11/otk-clientstore"&gt; &lt;value&gt; &lt;id&gt;9ecd3853...&lt;/id&gt; &lt;key&gt;l7xxa...a8c576&lt;/key&gt;</p> <p>-type: &lt;secret&gt;4e5ac05...1c26f&lt;/secret&gt;</p> <p>text &lt;service&gt;7e4535be-...f792a&lt;/service&gt;</p> <p>/xml; &lt;accountPlanMappingId&gt;de0d455...ae62c5d3&lt;/accountPlanMappingId&gt; &lt;label&gt;AppName&lt;=UTF-8 /label&gt; &lt;status&gt;active&lt;/status&gt;</p> <p>&lt;xmlbase64&gt;PGw3OkFwaUtleSB4b...Hk+DQo8L2w3OkFwaUtleT4=&lt;/xmlbase64&gt; &lt;/value&gt; &lt;/values&gt;</p>	<p><b>apikey</b> : The API Key to look up from the data storage.</p> <p><b>serviceid</b>(optional) : If this parameter is provided a result will only be returned if the API Key is valid for that Service ID.</p>
Error-Response					{ "error":"invalid_request", "error_description":"Missing or unknown parameters" }	

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
			x-ca-err:	400510 3 Pragma : no-cache Cache-Control : no-store status: 400 content-type: application/json; charset= UTF-8	If any required parameters or headers are missing, the request will fail.
Error-Response		x-ca-err:	{ "error": "invalid_request", "error_description": "SSL with client authentication is required" }	400520 4 Pragma : no-cache Cache-Control : no-store status: 403 content-type: application/json; charset= UTF-8	Authentication failed.
Error-Response		x-ca-err:	{ "error": "invalid_request", "error_description": "The client certificate is not valid" }	400520 5 Pragma : no-cache Cache-Control : no-	The client could not be authenticated.

ID	Operation URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
				store status: 401 content -type: applica tion /json; charset =UTF-8	
Error- Response		x-ca- err: 400530 3 Pragma : no- cache Cache- Control : no- store status: 400 content -type: text /xml; charset =UTF-8	{ "error":"invalid_request", "error_description":"The path component of the URL is invalid" }		The path component is invalid.
Error- Response		Allow: GET Pragma : no- cache Cache- Control : no- store status: 405			The HTTP method is not valid <b>Allow</b> : This header is required by <a href="#">RFC 2616</a> ( <a href="http://tools.ietf.org/html/rfc2616">http://tools.ietf.org/html/rfc2616</a> ) and contains a comma separated list of valid HTTP methods.
Error- Response		x-ca- err: 400500 0			Unknown error.

ID	Operation	URL- Path	HTTP Method	HTTP Header	HTTP Body/ Query Params (for GET attach params to URL-PATH)	Comment
					Pragma : no- cache Cache- Control : no- store status: 500	

## OAuth Client Assertions

Locate the OAuth Client assertions in the Policy Assertions/XML Security folder.

Use the following OAuth Client assertions to configure the Gateway as an OAuth client to consume OAuth-protected resources.

- [Retrieve OAuth 1.0 Token Assertion \(see page 364\)](#)
- [Consume OAuth 1.0 Resource \(see page 365\)](#)
- [Retrieve OAuth 2.0 Token Assertion \(see page 366\)](#)
- [Refresh OAuth 2.0 Token Assertion \(see page 366\)](#)

## Retrieve OAuth 1.0 Token Assertion

This assertion implements the two-stage handshaking process of OAuth 1.0.

Given the following:

- The user has an account on a website service that stores photos. To access the website, the user provides user credentials.
- The consumer is a third party application that wants to obtain access to the protected resource (photos) on a user's behalf.
- The user does not share the credentials to access the protected resource with the consumer.
- The consumer application is registered with the protected resource. The protected resource identifies the application by generating a consumer key and consumer secret.

In stage one, the consumer application requests access to the photo site. The request is directed to an authentication server and includes the unique consumer key identifying the application. The authentication server grants a temporary **OAuth request token**. The consumer application redirects the user to a callback URL on the photo site containing an **OAuth verifier**, where the user can explicitly allow (authorizes) the consumer application access to the photos.

In the second stage, the consumer application requests an **OAuth access token**, using the consumer key, the request token, and the verifier. The photo website grants the access token. The consumer application uses the access token to access the protected resource.

## Context Variables Set

---

oauth.access\_token  
oauth.access\_token\_secret  
oauth.auth\_req\_url  
oauth.full\_access\_token\_response  
oauth.full\_oauth\_token\_response  
oauth.oauth\_token  
oauth.oauth\_token\_secret

---

## Context Variables Used

---

oauth\_callback  
oauth\_token  
oauth\_token\_secret  
oauth\_verifier  
request.http.method

---

## Consume OAuth 1.0 Resource

The Consume OAuth 1.0 Resource assertion creates the OAuth authorization header. The header can be added to the HTTP Authorization header to consume OAuth 1.0 protected resources.

## Context Variables Set

---

oauth.resource\_authorization\_header

---

## Context Variables Used

---

access\_token  
access\_token\_secret  
request.http.method

---

## Retrieve OAuth 2.0 Token Assertion

The Retrieve OAuth 2.0 Token assertion is used to retrieve an access token from the authorization server. The token response from the authorization server may include a refresh token.

For authorization code and implicit grant types, this assertion implements the two-stage handshaking process of OAuth 2.0. It first returns the authorization request URL in a context variable. The client can then redirect the user-agent to the authorization request URL. After the authorization server authenticates the resource owner and access is granted, this assertion will be called back to perform the second stage handshaking, which will then return access token.

### Context Variables Set

---

oauth.access\_token

---

oauth.auth\_req\_url

---

oauth.full\_token

---

oauth.refresh\_token

---

## Refresh OAuth 2.0 Token Assertion

The Refresh OAuth 2.0 Token assertion is used to refresh an access token. An access token can be refreshed only if the authorization server issued a refresh token.

### Context Variables Set

---

oauth.access\_token

---

oauth.full\_token

---

oauth.refresh\_token

---

## Encapsulated Assertions

To find OTK related encapsulated assertions:

1. In the Policy Manager, go to **Tasks, Extensions and Add-Ons, Manage Encapsulated Assertions**.
2. Type OTK to filter the list. All OAuth encapsulated assertions start with OTK.
3. Select the encapsulated assertion and click **Properties** to view details.

In the following descriptions, the term "encas" is used as an abbreviation of "encapsulated assertion".

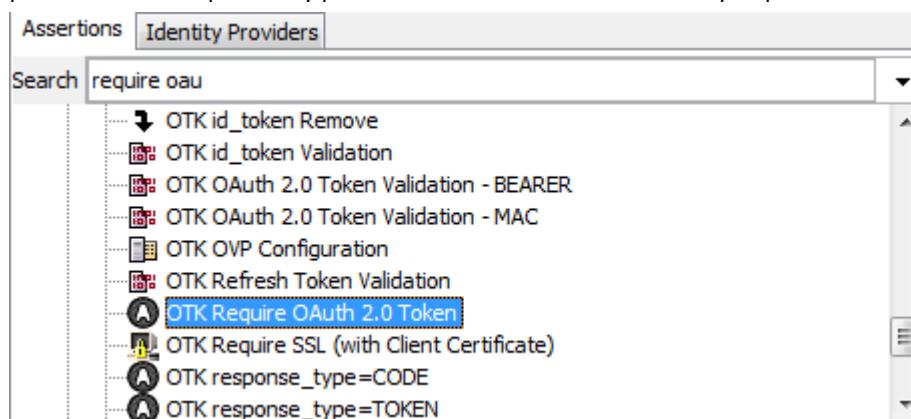
Key encapsulated assertions include:

- [OTK Require OAuth 2.0 Token \(see page 367\)](#)
- [OTK Access Token Retrieval \(see page 368\)](#)
- [OTK Client Persist \(see page 369\)](#)
- [OTK SCOPE Verification \(see page 371\)](#)

This is not a comprehensive list of all encapsulation assertions.

## OTK Require OAuth 2.0 Token

Use this encapsulated assertion to allows access only when a valid access\_token is presented by the client. The assertion searches for access\_tokens presented in the authorization header as a query parameter or as a post body parameter. Use this assertion as early as possible in an API policy.



This assertion uses the OTK Access Token Retrieval assertion to find the incoming access\_token. Refer to that description for error messages.

Drag the assertion into a policy and configure the properties shown in the table below.

Properties	Parameter Type	Notes
Name		
Required scope_req	String	If SCOPE is not required, this value can be empty.
SCOPE(s) required	String	A space separated list of required SCOPES. An access_token is only accepted if it has been granted with those SCOPE values.
Cache validation result (s)	Integ	This value cannot be empty. Represents the time in seconds for which an access_token is cached. The assertion initially validates an access_token. The validation result is then cached until the cache period expires. This increases performance, but also enables clients to use potentially expired access_tokens. The cache_lifetime value extends the lifetime of the token. A value of 0 indicates no caching is performed.
Is this a one-time access-token?	Bool	Default value: <b>false</b> . To allow an access_token to be considered valid only once for this endpoint, set this value to <b>true</b> . This setting is rare, but enables special use cases.
scope_fail		

Properties	Parameter	Type	Notes
Name			
Fail if this SCOPE was granted?		Bool	Default value: <b>false</b> . Set this value to “true” if a request should fail in the case that an access_token has been granted for at least one the specified SCOPE values listed above
Access Token	given_acce ss_token	String	Optional. The hardcoded value of an access token or a context variable representing an access_token. Use this property if an access_token is made available, but not by the client, or if the access_token is passed using a non-standard mechanism.

## Context Variables

The encas sets the following context variables:

Context Variable	Notes
<code> \${session.client_id}</code>	The client_id of the client that has received the token
<code> \${session.subscriber_id}</code>	the username of the resource_owner that has granted access for the client and therefore to access the endpoint
<code> \${session.scope}</code>	the SCOPE that was granted for this access_token
<code> \${session.expires_at}</code>	the expiration time in seconds
<code> \${access_token}</code>	the access_token that was used with this request

## OTK Access Token Retrieval

This encas is used within OTK Require OAuth 2.0 Token where it extracts an access\_token from a request. It works on given input variables and searches for an access\_token. When the token is found, it is passed in as an authorization header value:

authorization: Bearer <token>, authorization: MAC <mac-values>

Input Field	Parameter	Type	Notes
Name			
Allow Authorization Header	allow_heade r	Bool	Indicates whether to accept an access_token within the authorization header.
Allow Parameter er	allow_QUE ue	Bool	Indicates whether to accept and access_token as query or from post parameter.
Authentic ation Header	auth_head er	String	The content of the authorization header to be used. Values with scheme “Bearer” and “MAC” are accepted

Input Field	Parameter Name	Type	Notes
Paramet er	auth_para m_token	Strin g	The content of the variable that contains the access_token to be used
Access Token	given_acce ss_token	Strin g	The hardcoded value of an access token or a context variable representing an access_token. Allows you to provide an access_token other than from a well known location. This is useful if the access_token is held somewhere but not passed in with a request.

## Context Variables

The encas sets the following context variables:

Context Variable	Notes
<code> \${access_tok en}</code>	The client_id of the client that has received the token
<code> \${auth_head er}</code>	The username of the resource_owner that has granted access to the client and therefore access to the endpoint
<code> \${auth_sche me}</code>	The token scheme. One of the following values: <ul style="list-style-type: none"> <li>▪ Bearer</li> <li>▪ MAC</li> </ul>

The assertion creates an error if no access token is found or if multiple locations within the request contained a token.

```

HTTP Header:
Status: 401
Content-type: application/json
Pragma: no-cache
Cache-Control: no-store
HTTP Body:
{
  "error": "invalid_request",
  "error_description": "Missing or duplicate token"
}

```

## OTK Client Persist

The OTK model allows multiple OAuth clients and more than one client ID per client. The OTK Client Persist encas persists a new OAuth client. By default there is no need to use this encas directly. It is used within OAuth Manager to register a new client. Nevertheless, it may be useful for testing purposes or special cases.

Input Field	Parameter Name	Flag
Client Identifier	client_ident	Required.
Client ID	client_key	new client ID

---

Client Secret	client_secret	new client ID
Redirect URI	callback	new client ID optional
Description	description	new client
Environment	environment	new client ID optional
Expiration	expiration	new client ID
Client Name	client_name	new client
Organization	org	new client
Registered By	registered_by	Required.
Valid SCOPE	scope	new client ID optional
Status	client_status	new client ID
Client Type	type	new client
New Client	persist_client	Boolean
New Client ID	persist_client_key	Boolean
New Client including client ID	persist_client_and_key	Boolean

## Create a New Client

To create a new client:

1. Set all required values.
2. Set all values flagged as "new client" (unless optional).
3. Set New Client boolean to **true**.
4. Ensure that the following values are unique:
  - Client Identifier
  - Client Name
  - Client Name with Organization

## Create a New Client ID

To create a new client ID:

1. Set all required values.
2. The Client Identifier value must match an existing value.
3. Set all values flagged as "new client ID" (unless optional).

4. Set New Client ID to **true**.
5. Ensure that the Client ID value is unique.

## Create a New Client with a New Client ID

To create a new client with a new client ID:

1. Set all values (unless optional).
2. Set New Client including client ID to **true**.

## Context Variables

The encas sets the following context variables to the values shown if no error occurs:

Context Variable	Value
<code> \${status}</code>	200
<code> \${result}</code>	persisted
<code> \${content_type}</code>	text/plain

If an error occurs:

```
Status = 403
Content-type = application/json
Message:
{
  "error": "Client registration failed",
  "error_description": "The client could not be registered"
}
```

## OTK SCOPE Verification

Use this encapsulated assertion whenever an API should process a request depending on the SCOPE that was granted to an access\_token. Multiple instances of this encas can be used within the same policy.

The MAG policies use this encas at the endpoint /opened/connect/v1/userinfo. Depending on the granted SCOPE the endpoint returns an email address, profile information, and other values.

In a policy position this encas is always used after using “OTK Require OAuth 2.0 Token”.

Input Field	Parameter	Notes
Granted SCOPE	scope. granted	Insert the values that were granted to the used access_token. In combination with “OTK Require OAuth 2.0 Token” simply use the example shown
Required SCOPE	scope. required	Insert the SCOPE value that must be found in the list of the granted SCOPE values. The encas will fail if the required SCOPE cannot be found.

Input Field	Parameter Notes
Fail if SCOPE is found (true false)?	fail Set this value to "true" if a request should fail in the case that an access_token has been granted for at least one the specified SCOPE values listed above.

## Context Variables

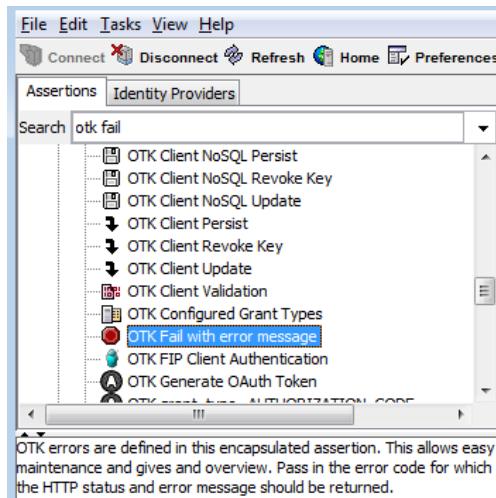
No context variables are set by this encas.

## Error Codes

### How to Add Error Codes to a Policy

Error codes are structured as follows: 4-digits (identifies the API) + 3-digits (identifies the actual error).

Drag the **OTK Fail with error message** encapsulated assertion into a policy to provide standardized error responses when error conditions are met. Using this single encapsulated assertion replaces having to define each error code explicitly in each policy.



For protected APIs, you can create your own codes by editing the **OTK Fail with error message** policy fragment located in OTK-version/validation-verification/. Use codes in the 700 to 899 range. Do not change or overwrite the existing error codes.

To add error handling to a policy:

1. In the Assertions pane, locate the **OTK Fail with error message** assertion.
2. Drag the assertion into your policy where you logically want to trigger the error. On release, the configuration dialog appears.

3. For **Error Code**, provide the 3-digit number that identifies the actual error. Use existing codes, or create custom codes in the 700 to 899 range.  
Refer to the Error Codes below for examples.

4. For **API prefix for error code**, provide a 4-digit number to identifier your API or use \${apiPrefix} to reference a pre-defined API prefix.

```

2  ⚡ Comment: Default assertions for a typical OAuth 2.0 protected API
3  ✗ Error Customize Error Response Override default gateway error response
4  🚫 POST GET OTK HTTP Method Validation Specify valid HTTP methods here
5  🔒 SSL OTK Require SSL (with Client Certificate) Require SSL, optionally require a client certificate
6  🔑 Optional OTK FIP Client Authentication Required if previous line was configured to require a client certificate!
7  🌐 OAuth token OTK Require OAuth 2.0 Token Searches for and validates an OAuth 2.0 access_token
8  💬 Comment: === Implement API related validation here
9  ✅ initialize Set Context Variable myparam as String to: ${request.http.parameter.myparam} Initialize variable here
10  📄 parameters At least one assertion must evaluate to true check for required parameters
11  ✅ Compare Variable: ${myparam} is not empty; If Multivalued all values must pass
12  🚫 error All assertions must evaluate to true Missing or empty required parameter
13  📝 Add Audit Details: "Required parameter 'myparam' is empty or missing"
14  ⚡ 103 OTK Fail with error message Missing parameter
15  💬 Comment: === Implement API related logic here
16  💬 Comment: Backend call, transformations, ...

```

Commonly returned codes are:

1. xxxx**201**: The client authentication failed
2. xxxx**202**: The resource\_owner authentication failed
3. xxxx**203**: SSL was required but not used by the client
4. xxxx**204**: SSL with client authentication was required but not used by the client
5. xxxx**990**: The token has expired
6. xxxx**991**: The access\_token has not been granted for the required SCOPE
7. xxxx**992**: No access\_token was included in the request, An access\_token was included more than once
8. xxxx**993**: The token is disabled which means that associated client is disabled
9. xxxx**000**: Indicates that something unexpected went wrong

## Error Handling

The Customize Error Response assertion (line 3) is required and sets up the error handling mechanism. By modifying this assertion, you can override the default error handling response. It precedes the OTK Require OAuth 2.0 Token assertion (line 7) in a policy. The complete list of error codes and messages are referenced in the OTK Fail with error message encapsulated assertion (line 14).

By default, error codes are returned in the HTTP response message header `x-ca-err` field. For example:

```
x-ca-err: 3007103
```

Additionally:

- `allow` is added to the response if the given HTTP method is not valid. It contains a comma-separated list of valid methods
- `www-authenticate` is added to the response if the given client or resource owner credentials are missing or could not be validated

The message body contains:

```
{ "error": "invalid_request", "error_description": "Missing or duplicate parameters" }
```

## Error Codes

The following error codes are defined in the OTK Fail with error message encapsulated assertion.

Error groups contain one or more APIs. To find an error, click the group then search for the specific error code.



When an unexpected `content_type` is encountered, an error is returned, but the error code is inaccurate.

For error codes 1000-3000, see the CA Mobile API Gateway wiki; [wiki.ca.com/mag](https://wiki.ca.com/mag) (<https://wiki.ca.com/mag>)

- [3000/ request\\_authorization\\_init \(see page 375\)](#)
- [3001/ request\\_authorization\\_login \(see page 380\)](#)
- [3002/ request\\_authorization\\_consent \(see page 382\)](#)
- [3003/ request\\_token\\_password\\_flow, request\\_token\\_code\\_flow, request\\_token\\_refresh\\_flow, request\\_token\\_client\\_creds\\_flow, request\\_token\\_jwt\\_flow, request\\_token\\_saml\\_flow \(see page 383\)](#)
- [3004/ revoke\\_token \(see page 388\)](#)
- [3005/ client\\_details\\_export \(see page 389\)](#)
- [3006/ resource\\_owner\\_logout \(see page 390\)](#)
- [3007/ resource\\_owner\\_session\\_status \(see page 392\)](#)
- [4000/ get\\_client\\_id\\_filter, get\\_client\\_id\\_ident, get\\_client\\_id\\_org, get\\_client\\_id\\_name, get\\_client\\_id, get\\_all\\_client\\_id, get\\_client\\_by\\_ident, get\\_client\\_by\\_clientkey, get\\_client\\_registered\\_by, get\\_client\\_org, get\\_client\\_by\\_name\\_org, get\\_all\\_client \(see page 393\)](#)
- [4001/ persist\\_client, persist\\_client\\_id, persist\\_client\\_and\\_client\\_id \(see page 395\)](#)
- [4002/ delete\\_client, revoke\\_client\\_id \(see page 396\)](#)
- [4003/ update\\_client, update\\_client\\_id, update\\_client\\_id\\_registered\\_by \(see page 397\)](#)

- [4100/ persist临时 \(see page 398\)](#)
- [4101/ persist\\_token\\_oauth1 \(see page 399\)](#)
- [4102/ persist\\_token\\_oauth2 \(see page 399\)](#)
- [4103/ update\\_token\\_status \(see page 400\)](#)
- [4104/ update\\_oauth1\\_token\\_owner \(see page 401\)](#)
- [4105/ update\\_oauth1\\_token\\_verifier \(see page 401\)](#)
- [4106/ revoke\\_oauth\\_token \(see page 402\)](#)
- [4107/ delete\\_oauth\\_token \(see page 403\)](#)
- [4108/ disable\\_oauth\\_token \(see page 404\)](#)
- [4109/ get\\_oauth\\_token, get\\_oauth\\_token\\_param, get\\_oauth\\_token\\_cid\\_ro, get\\_oauth\\_token\\_status\\_ro \(see page 404\)](#)
- [4110/ get\\_temporary\\_token\\_t, get\\_temporary\\_token\\_t\\_v, get\\_temporary\\_token\\_cid\\_ro \(see page 405\)](#)
- [4111/ register\\_jwt \(see page 406\)](#)
- [4112/ lookup\\_jwt \(see page 407\)](#)
- [4113/ remove\\_jwt \(see page 408\)](#)
- [4200/ delete\\_expired\\_session \(see page 408\)](#)
- [4201/ delete\\_session \(see page 409\)](#)
- [4202/ get\\_session \(see page 410\)](#)
- [4203/ create\\_session \(see page 411\)](#)
- [5000/ validate\\_client \(see page 411\)](#)
- [5001/ validate\\_token \(see page 413\)](#)
- [5002/ validate\\_refresh\\_token \(see page 414\)](#)
- [5003/ token\\_revocation \(see page 416\)](#)
- [5004/ validate\\_id\\_token, create\\_id\\_token \(see page 417\)](#)

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
3000/ request_authorization_init	<b>3000</b> <b>103</b>	invalid_r equest	inva lid	repeat the request including all required parameters and/or headers - The request did not include the specified parameters for the API - The request included duplicate parameters - The request included invalid parameters	Repeat the request including all required parameters and/or headers - The request did not include the specified parameters for the API - The request included duplicate parameters - The request included invalid parameters

Group/ API_ID	Code Category Info Reasons	How to resolve
	<p>request included required headers or parameters but without value, empty</p> <p><b>3000</b> invalid_r inva - The mag- <b>107</b> equest lid identifier is mag not - associated iden with a tifie device r</p>	Repeat the request using a valid mag-identifier
	<p><b>3000</b> invalid_r inva - The <b>108</b> equest lid referenced status or contact the system administrator to mag device is in do so - status iden 'registered' tifie and has to r be changed to 'activated' before this request can succeed</p>	Either use MAG Manager to change the device status or contact the system administrator to do so
	<p><b>3000</b> invalid_r inva - None of <b>114</b> equest lid the redi registered rect redirect_uri _uri 's were used - no redirect_uri given: open redirect_uri 's are not supported and therefore the redirect_uri has to be provided - no</p>	Repeat the request using a valid redirect_uri

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
				<p>redirect_uri  given: if  the client  is of type  'public' a  registered  redirect_uri  has to be  provided  - no  redirect_uri  given: if  multiple  redirect_uri  's are  registered  one has to  be passed  in  - The  format is  not valid, e.  g.: it does  not have a  scheme (<a href="http://https://wiki.l7tech.commediawiki/index.php/OTK_3_4_0_API">http://https://wiki.l7tech.commediawiki/index.php/OTK_3_4_0_API</a>),  myscheme:  //)</p>	
<b>3000</b>	invalid_scope	inval	- No scope	Repeat the request using valid scope values	
<b>115</b>	cope	lid	value	sco matched a pe registered was one for this req client uest - Multiple ed scopes were requested but not separated by a space ( ' ) character	Repeat the request using a valid response_type

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
	<b>3000</b>	invalid_r	uns	- The response type was malformed or is unknown	
	<b>116</b>	esponse upp	response_t type	- The response type requested but the scope did not include 'openid'	
				- The response type requested but the request did not include a 'nonce'	
	<b>3000</b>	unauthor una	- The client nt	Verify the configuration of the client and repeat the request.	
	<b>117</b>	ized_clie	uth	oriz valid for the used client response_t type	
				- The client this may not be req valid due uest to the 'type'	
				- A confidential client using an implicit grant must have a registered redirect_uri	
				- A public client must	

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
				have a registered redirect_uri	
	<b>3000</b>	invalid_r	inva	- The client is of type public but has no registered redirect_uri	Check if the client is configured as expected
	<b>130</b>	equest	lid	- The client is using an implicit grant type but has no registered redirect_uri	
	<b>3000</b>	invalid_r	inva	- The given client_id or client_secr is not valid - The client_id and/or client_secr is missing - The credentials were not provided as base64 encoded value - The given client_id is configured as 'Master Key' - The client_id is disabled - The client_id has	Repeat the request using valid credentials. If the error still occurs contact the system administrator
	<b>201</b>	equest	lid		

Group/ API_ID	Code Category Info Reasons	How to resolve
	expired (if it had a limited lifetime)	
	<b>3000</b> invalid_r miss - The client miss - The client Repeat the request using 'https' <b>203</b> equest ing did not use SSL 'https' but 'http'	
3001/	<b>3001</b> invalid_r inva <b>000</b> equest lid	
request_authoriz		
ation_login	<b>3001</b> invalid_r inva - The request did not include all required parameters and/or headers <b>103</b> equest lid - The request included duplicate parameters - The request included required headers or parameters but without value, empty	Repeat the request including all required parameters and/or headers
	<b>3001</b> invalid_r inva - The session lifetime has expired <b>110</b> equest lid - The session has been granted and therefore used	Repeat the authorization request and have the resource_owner authenticate before the session times out

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
	<b>3001</b>	invalid_r	inva	- None of <b>114</b> equest lid the redi registered rect redirect_uri _uri 's were used - For missing redirect_uri : Open redirect_uri 's are not supported and therefore the the redirect_uri has to be provided - For missing redirect_uri : If the client is of type 'public' a registered redirect_uri has to be provided - The format is not valid, e. g.: two redirect_uri 's were included	Repeat the request using a valid redirect_uri
	<b>3001</b>	authenti	aut	- The <b>123</b> cation_e hen resource_o rror ticat wner has ion denied to was login. deni 'action' ed was set to 'cancel' instead of 'login'	Convince the resource_owner to login
	<b>3001</b>	invalid_r	inva	- The given <b>202</b> equest lid username reso or	Repeat the request using valid credentials. If the error still occurs contact the system administrator

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
				urce password own is not valid er - The username and/or password is missing - The social login credentials were invalid - The cookie has expired	
	<b>3001</b>	invalid_r	miss	- The client	Repeat the request using 'https'
	<b>203</b>	equest	ing	did not use SSL 'https' but 'http'	
3002/ <a href="#">request_authorization_consent</a>	<b>3002</b>	invalid_r	inva		
	<b>000</b>	equest	lid		
	<b>3002</b>	invalid_r	inva	- The	Repeat the request including all required
	<b>103</b>	equest	lid	request did not include parameters and/or headers	
				met headers ers and/or parameters as specified for the API	
				- The request included duplicate parameters	
				- The request included required headers or parameters but without value, empty	

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
	<b>3002</b>	invalid_r	inva	- Access to	Repeat the authorization process and have the
	<b>110</b>	equest	lid	resources	resource_owner authenticate before the
			sess	was	session times out
			ion	granted	
				but after	
				the session	
				timed out	
				- Access to	
				resources	
				was denied	
				but after	
				the session	
				timed out	
				- The	
				session has	
				been	
				granted	
				and	
				therefore	
				used	
	<b>3002</b>	authoriza	aut	- The	Convince the resource_owner to grant access
	<b>124</b>	tion_erro	hori	resource_o	
		r	zati	wner has	
			on	denied	
			was	access to	
			deni	resources.	
			ed	'action'	
				was set to	
				'Denied'	
				instead of	
				'Grant'	
	<b>3002</b>	invalid_r	miss	- The client	Repeat the request using 'https'
	<b>203</b>	equest	ing	did not use	
			SSL	'https' but	
				'http'	
	<b>3003</b>	invalid_r	inva		
request_token_p	<b>000</b>	equest	lid		
assword_flow,					
request_token_c					
ode_flow,					
request_token_re					
fresh_flow,					
request_token_cl					
ient_creds_flow,					

Group/ API_ID	Code Category Info Reasons	How to resolve
request_token_j wt_flow, request_token_s aml_flow	<p><b>3003</b> invalid_r inva - The request did not include all required parameters and/or headers</p> <p><b>103</b> equest lid - The request included duplicate parameters</p> <p>- The request included required headers or parameters but without value, or empty</p>	Repeat the request including all required parameters and/or headers
	<p><b>3003</b> invalid_r inva - The mag-identifier is not associated with a device</p> <p><b>107</b> equest lid - The given authorization code has already been used</p>	Repeat the request using a valid mag-identifier
	<p><b>3003</b> invalid_r inva - The given authorization code has already been used</p> <p><b>113</b> equest lid - The given authorization code has already been used</p>	Repeat the authorization process using a valid grant code

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
				invalid - The given refresh_tok en has expired - The given refresh_tok en has been revoked	
	<b>3003</b>	invalid_r	inva	- The code	Repeat the request using a valid redirect_uri
	<b>114</b>	quest	lid	to was redi requested rect using a _uri different redirect_uri	
	<b>3003</b>	invalid_s	inva	- No scope	Repeat the request using valid scope values
	<b>115</b>	cope	lid	value sco matched a pe registered was one for this req client uest - Multiple ed scopes were requested but not separated by a space ( ' ) character - The refresh_tok en request included other or more SCOPE values than initially issued	
	<b>3003</b>	unauthor	una	- The client	Use a valid client, the one that has receive he
	<b>117</b>	ized_clie	uth	nt was not oriz the ed recipient clie of the nt given for token	refresh_token initially

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
			this req uest		
	<b>3003</b>	unsuppo	the - and		Repeat the request using a valid grant_type
	<b>119</b>	rted_gra	give invalid		
		nt_type	n grant_type		
			gran was used		
		t_ty			
		pe			
		is			
		not			
		sup			
		port			
		ed			
	<b>3003</b>	invalid_r	inva	- The given	Repeat the request using valid credentials. If
	<b>201</b>	equest	lid	client_id or	the error still occurs contact the system
			clie	client_secr	administrator
		nt	et is not		
			valid		
			- The		
		client_id	and/or		
			client_secr		
		et is	et is		
		missing	missing		
		- The	- The		
		credentials	credentials		
		were not	were not		
		provided	provided		
		as base64	as base64		
		encoded	encoded		
		value	value		
		- The given	- The given		
		client_id is	client_id is		
		configured	configured		
		as 'Master	as 'Master		
		Key'	Key'		
		- The client	- The client		
		is disabled	is disabled		
		- The	- The		
		client_id	client_id		
		has	has		
		expired (if	expired (if		
		it had a	it had a		
		limited	limited		
		lifetime)	lifetime)		
	<b>3003</b>	invalid_r	inva	- The given	Repeat the request using valid credentials. If
	<b>202</b>	equest	lid	username	the error still occurs contact the system
			reso	or	administrator

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
				<p>urce password            own is not valid            er - The            username            and/or            password            is missing            - The JWT            (id_token)            has expired            - The JWT            (id_token)            has an            invalid            signature            - The JWT            (id_token)            is            unknown.            Only JWT            that were            issued by            the server            can be used            - The JWT            (id_token)            was            revoked            - The SAML            token has            expired            - The SAML            token has            has an            invalid            signature</p>	
	<b>3003</b>	invalid_r	miss	- The client	Repeat the request using 'https'
	<b>203</b>	equest	ing	did not use SSL	'https' but 'http'
	<b>3003</b>	invalid_r	inva	- The	The token (client) has to be enabled or revoked
	<b>993</b>	equest	lid	refresh_tok	and a new one requested.
				toke en is n disabled which means that associated client is disabled	

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
3004/revoke_token	<b>3004</b> invalid_r inva <b>000</b> equest lid				
	<b>3004</b> invalid_r inva - The request did not include all required parameters and/or headers			Repeat the request including all required parameters and/or headers	
	<b>103</b> equest lid - The request included duplicate parameters			- The request included duplicate parameters	
				- The request included required headers or parameters but without value, or empty	
	<b>3004</b> unauthorized - The client was not authorized to make the request			Use a valid client	Contact the system administrator
	<b>117</b> ized_clie uth - Neither a Bearer token nor a refresh token was given for this request				
	<b>3004</b> unsupported - The given token type is not supported			Contact the system administrator	
	<b>118</b> rted upp - Neither a Bearer token nor a refresh token was used				

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
	<b>3004</b>	invalid_r	inva	- The given client_id or client_secret is not valid	Repeat the request using valid credentials. If the error still occurs contact the system administrator
	<b>201</b>	equest	lid	- The client_id and/or client_secret is missing	- The credentials were not provided as base64 encoded value - The client is disabled - The client_id has expired (if it had a limited lifetime)
	<b>3004</b>	invalid_r	miss	- The client did not use SSL 'https' but 'http'	Repeat the request using 'https'
<a href="#">3005/ client_details_export</a>	<b>3005</b>	invalid_r	inva		
	<b>000</b>	equest	lid		
	<b>3005</b>	invalid_r	inva	- The request did not include required parameters and/or headers	Repeat the request including all required parameters and/or headers
	<b>103</b>	equest	lid	- The request included duplicate parameters	- The request included duplicate parameters

Group/ API_ID	Code Category Info Reasons	How to resolve
	request included required headers or parameters but without value, empty	
	<b>3005</b> invalid_r inva - The <b>132</b> equest lid server is serv searching er for an conf unknown igur certificate atio n	Verify that the server has been configured to search for its public SSL cert via the correct certificate alias
	<b>3005</b> invalid_r inva - The given <b>201</b> equest lid client_id is clie unknown nt - The client is disabled	Check with OAuth Manager for verify the current state of the client
	<b>3005</b> invalid_r inva - The given <b>202</b> equest lid username reso or urce password own is not valid er - The username and/or password is missing	Repeat the request using valid credentials. If the error still occurs contact the system administrator
	<b>3005</b> invalid_r miss - The client <b>203</b> equest ing did not use SSL 'https' but 'http'	Repeat the request using 'https'
3006/ resource_owner_ logout	<b>3006</b> invalid_r inva <b>000</b> equest lid	
	<b>3006</b> invalid_r inva - The <b>103</b> equest lid request did para not include met headers ers and/or parameters as	Repeat the request including all required parameters and/or headers

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
				specified for the API - The request included duplicate parameters - The request included required headers or parameters but without value, empty	
	<b>3006</b>	invalid_r	inva	- The mag- 107 equest lid identifier is identifier	Repeat the request including a valid mag- mag not - associated iden with a tifie device r - The device was registered as a mobile client (SCOPE=ms so) but no mag- identifier was included
	<b>3006</b>	unauthor	una	- The client 117 ized_clie uth is not a nt oriz valid user ed of this clie token nt for this req uest	Use a valid client whose client_id is found within the 'azp' key of the id_token. For mobile clients the client has to be valid for the given mag-identifier
	<b>3006</b>	invalid_r	inva	- The given 201 equest lid client_id or the error still occurs contact the system clie client_secr administrator nt et is not valid	Repeat the request using valid credentials. If the error still occurs contact the system administrator

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
				<ul style="list-style-type: none"> <li>- The client_id and/or client_secret is missing</li> <li>- The credentials were not provided as base64 encoded value</li> <li>- The client is disabled</li> <li>- The client_id has expired (if it had a limited lifetime)</li> </ul>	
	<b>3006</b>	invalid_r	miss	- The client request did not use SSL 'https' but 'http'	Repeat the request using 'https'
<b>3007/</b>	<b>3007</b>	invalid_r	inva		
	<b>000</b>	equest	lid		
		resource_owner_			
		session_status			
	<b>3007</b>	invalid_r	inva	- The request did not include required parameters and/or headers	Repeat the request including all required parameters and/or headers
	<b>103</b>	equest	lid	- The request included duplicate parameters	
	<b>3007</b>	invalid_r	miss	- The client request did not use SSL 'https' but 'http'	Repeat the request using 'https'
	<b>203</b>	equest	ing		

**3007** invalid\_r inva - The token Request a new token  
**990** equest lid has expired  
          acce - The token  
          ss\_t does not  
          oke exist  
          n

**3007** invalid\_r inva - The token Request a new token that is scoped  
**991** equest lid has not accordingly  
          acce been  
          ss\_t granted for  
          oke the  
          n required  
          SCOPE

**3007** invalid\_r inva - No Use the access\_token either within the  
**992** equest lid access\_tok authorization header, as query parameter or as  
acce en was post body  
ss\_t included in  
oke the request  
n - An  
access\_tok  
en was  
included  
more than  
once  
parameter

**3007** invalid\_r inva - The token The token (client) has to be enabled or revoked  
**993** equest lid is disabled and a new one requested.  
toke which  
n means that  
associated  
client is  
disabled

```
client_id_filter,    4000 invalid_r inva
get_client_id_ide
nt,
get_client_id_or
g,
get_client_id_na
me,
get_client_id,
get_all_client_id,
get_client_by_id
ent,
get_client_by_cli
```

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
entkey, get_client_regist ered_by, get_client_org, get_client_by_na me_org, get_all_client	<b>4000</b> invalid_r <b>103</b> equest	inva	- The request did not include required parameters and/or headers	Repeat the request including all required parameters and/or headers	
		lid	- The request included duplicate parameters		
		req	- The request included required headers or parameters but without value, empty		
	<b>4000</b> invalid_r <b>204</b> equest	inva	- The requester did not use mutual ssl	Repeat the request using mutual SSL	
		lid			
		req			
		uest			
	<b>4000</b> invalid_cl <b>205</b> ient	inva	- The given client certificate is unknown	Repeat the request using valid credentials. If the error still occurs contact the system administrator	
		lid			
		clie			
		certi			
		ficat			
		e			
	<b>4000</b> invalid_r <b>303</b> equest			Repeat the request using a valid path	

Group/ API_ID	Code Category Info Reasons	How to resolve
		inva - The API lid was used ope with an rati invalid on ending path value
4001/ persist_client, persist_client_id, persist_client_an d_client_id	<b>4001</b> invalid_r inva <b>000</b> equest lid	
	<b>4001</b> invalid_r inva - The <b>103</b> equest lid request did parameters and/or headers para not include met headers ers and/or parameters as specified for the API - The request included duplicate parameters - The request included required headers or parameters but without value, empty	Repeat the request including all required parameters and/or headers
	<b>4001</b> invalid_r inva - The given <b>201</b> equest lid client_iden the error still occurs contact the system clie references administrator nt an unknown client - The given client is unknown	Repeat the request using valid credentials. If the error still occurs contact the system
	<b>4001</b> invalid_r clie - The <b>300</b> equest nt 'client_iden coul t' already	Repeat the request using unique values.

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
			d	exist not - The be 'client_key' pers already iste exist d - The combinatio n of 'name' and 'organizatio n' already exists	
	<b>4001</b>	invalid_r	clie	- The given	Repeat the request using a valid value for
	<b>301</b>	equest	nt	value for	'persist_type'
				coul	'persist_typ
			d	e' is invalid	
			not		
			be		
			pers		
			iste		
			d		
	<b>4001</b>	invalid_r	inva	- The API	Repeat the request using a valid path
	<b>303</b>	equest	lid	was used	
			ope	with an	
			rati	invalid	
			on	ending	
			path	value	
4002/ delete_client, revoke_client_id	<b>4002</b>	invalid_r	inva		
	<b>000</b>	equest	lid		
	<b>4002</b>	invalid_r	inva	- The	Repeat the request including all required
	<b>103</b>	equest	lid	request did	parameters and/or headers
			para	not include	
			met	headers	
			ers	and/or	
			parameters		
			as		
			specified		
			for		
			the		
			-		
			The		
			request		
			included		
			duplicate		
			parameters		
			-		
			For		
			'revoke':		
			The		
			request		

Group/ API_ID	Code Category Info Reasons	How to resolve
	<p>included 'client_iden t' and 'client_key' - The request included required headers or parameters but without value, empty</p>	
	<p><b>4002</b> invalid_r inva - The API <b>303</b> equest lid was used ope with an rati invalid on ending path value</p>	Repeat the request using a valid path
4003/ update_client, update_client_id, update_client_id _registered_by	<p><b>4003</b> invalid_r inva <b>000</b> equest lid</p>	
	<p><b>4003</b> invalid_r inva - The <b>103</b> equest lid request did para not include met headers ers and/or parameters as specified for the API - The request included duplicate parameters - The request included required headers or parameters</p>	Repeat the request including all required parameters and/or headers

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
				but without value, empty	
	<b>4003</b>	invalid_r	inva	- The API	Repeat the request using a valid path
	<b>303</b>	equest	lid	was used ope with an rati invalid on ending path value	
<a href="#">4100/ persist_temporary</a>	<b>4100</b>	invalid_r	inva		
	<b>000</b>	equest	lid		
	<b>4100</b>	invalid_r	inva	- The	Repeat the request including all required
	<b>103</b>	equest	lid	request did para not include met headers ers and/or parameters as specified for the API - The request included duplicate parameters - The request included required headers or parameters but without value, empty	parameters and/or headers
	<b>4100</b>	invalid_r	inva	- The	
	<b>204</b>	equest	lid	requester req did not use uest mutual ssl	Repeat the request using mutual SSL
	<b>4100</b>	invalid_cl	inva	- The given	Repeat the request using valid credentials. If
	<b>205</b>	ient	lid	client clie certificate nt is unknown certi ficat e	the error still occurs contact the system administrator

Group/ API_ID	Code Category Info Reasons	How to resolve
	<b>4100</b> invalid_r inva - The API <b>303</b> equest lid was used ope with an rati invalid on ending path value	Repeat the request using a valid path
4101/ persist_token_oauth1	<b>4101</b> invalid_r inva <b>000</b> equest lid	
	<b>4101</b> invalid_r inva - The <b>103</b> equest lid request did parameters and/or headers para not include met headers ers and/or parameters as specified for the API - The request included duplicate parameters - The request included required headers or parameters but without value, empty	Repeat the request including all required parameters and/or headers
	<b>4101</b> invalid_r inva - The API <b>303</b> equest lid was used ope with an rati invalid on ending path value	Repeat the request using a valid path
4102/ persist_token_oauth2	<b>4102</b> invalid_r inva <b>000</b> equest lid	
	<b>4102</b> invalid_r inva - The <b>103</b> equest lid request did parameters and/or headers para not include met headers ers and/or parameters	Repeat the request including all required parameters and/or headers

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
				as specified for the API - The request included duplicate parameters - The request included required headers or parameters but without value, empty	
	<b>4102</b>	invalid_r	inva	- The API	Repeat the request using a valid path
	<b>303</b>	equest	lid	was used ope with an rati invalid on ending path value	
<a href="#">4103/ update_token_status</a>	<b>4103</b>	invalid_r	inva		
	<b>000</b>	equest	lid		
	<b>4103</b>	invalid_r	inva	- The	Repeat the request including all required
	<b>103</b>	equest	lid	request did para not include met headers ers and/or parameters as specified for the API - The request included duplicate parameters - The request included required headers or parameters	parameters and/or headers

Group/ API_ID	Code Category Info Reasons	How to resolve
	but without value, empty	
	<b>4103</b> invalid_r inva - The API <b>303</b> equest lid was used ope with an rati invalid on ending path value	Repeat the request using a valid path
4104/ update_oauth1_t oken_owner	<b>4104</b> invalid_r inva <b>000</b> equest lid	
	<b>4104</b> invalid_r inva - The <b>103</b> equest lid request did parameters and/or headers para not include met headers ers and/or parameters as specified for the API - The request included duplicate parameters - The request included required headers or parameters but without value, empty	Repeat the request including all required parameters and/or headers
	<b>4104</b> invalid_r inva - The API <b>303</b> equest lid was used ope with an rati invalid on ending path value	Repeat the request using a valid path
	<b>4105</b> invalid_r inva <b>000</b> equest lid	

Group/ API_ID	Code Category Info Reasons	How to resolve
<a href="#">update_oauth1_token_verifier</a>	<p><b>4105</b> invalid_r inva - The request did not include all required parameters and/or headers</p> <p><b>103</b> equest lid met headers and/or parameters as specified for the API</p> <p>- The request included duplicate parameters</p> <p>- The request included required headers or parameters but without value, or empty</p>	Repeat the request including all required parameters and/or headers
	<p><b>4105</b> invalid_r inva - the given verifier already has an associated verifier</p> <p><b>125</b> equest lid token</p>	Repeat the authorization flow and set the verifier once only
	<p><b>4105</b> invalid_r inva - The API operation with an invalid path value</p> <p><b>303</b> equest lid</p>	Repeat the request using a valid path
<a href="#">4106/ revoke_oauth_token</a>	<p><b>4106</b> invalid_r inva</p> <p><b>000</b> equest lid</p>	
	<p><b>4106</b> invalid_r inva - The request did not include all required parameters and/or headers</p> <p><b>103</b> equest lid met headers and/or parameters as specified for the API</p>	Repeat the request including all required parameters and/or headers

Group/ API_ID	Code Category Info Reasons	How to resolve
	<p>specified for the API</p> <ul style="list-style-type: none"> <li>- The request included duplicate parameters</li> <li>- The request included required headers or parameters but without value, empty</li> </ul>	
	<p><b>4106</b> invalid_r inva - The API <b>303</b> equest lid was used ope with an rati invalid on ending path value</p>	Repeat the request using a valid path
<a href="#">4107 / delete_oauth_token</a>	<p><b>4107</b> invalid_r inva <b>000</b> equest lid</p> <p><b>4107</b> invalid_r inva - The <b>103</b> equest lid request did para not include met headers ers and/or parameters as specified for the API</p> <ul style="list-style-type: none"> <li>- The request included duplicate parameters</li> <li>- The request included required headers or parameters</li> </ul>	Repeat the request including all required parameters and/or headers

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
				but without value, empty	
	<b>4107</b>	invalid_r	inva	- The API	Repeat the request using a valid path
	<b>303</b>	equest	lid	was used ope with an rati invalid on ending path value	
<a href="#">4108/ disable_oauth_token</a>	<b>4108</b>	invalid_r	inva		
	<b>000</b>	equest	lid		
	<b>4108</b>	invalid_r	inva	- The	Repeat the request including all required
	<b>103</b>	equest	lid	request did para not include met headers ers and/or parameters as specified for the API - The request included duplicate parameters - The request included required headers or parameters but without value, empty	parameters and/or headers
<a href="#">oauth_token, get_oauth_token</a>	<b>4109</b>	invalid_r	inva		
	<b>000</b>	equest	lid		

Group/ API_ID	Code Category Info Reasons	How to resolve
_param, get_oauth_token _cid_ro, get_oauth_token _status_ro	<b>4109</b> invalid_r inva - The request did not include all required parameters and/or headers. <b>103</b> equest lid - The request included duplicate parameters or required headers or parameters but without value, or empty.	Repeat the request including all required parameters and/or headers.
	<b>4109</b> invalid_r inva - The API path was used with an invalid or ending path value.	Repeat the request using a valid path.
4110/	<b>4110</b> invalid_r inva <b>000</b> equest lid	
	get_temporary_token_t, get_temporary_token_t_v, get_temporary_token_cid_ro	

Group/ API_ID	Code Category Info Reasons	How to resolve
	<b>4110</b> invalid_r inva - The request did not have all required parameters and/or headers. <b>103</b> equest lid - The request included duplicate parameters or required headers or parameters but without value, or empty.	Repeat the request including all required parameters and/or headers.
	<b>4110</b> invalid_r inva - The API path was used with an invalid ending path value. <b>303</b> equest lid - The request included duplicate parameters or required headers or parameters but without value, or empty.	Repeat the request using a valid path value.
<a href="#">4111 / register_jwt</a>	<b>4111</b> invalid_r inva <b>000</b> equest lid	<b>4111</b> invalid_r inva - The request did not have all required parameters and/or headers. <b>103</b> equest lid - The request included duplicate parameters or required headers or parameters but without value, or empty.

Group/ API_ID	Code Category Info Reasons	How to resolve
	included required headers or parameters but without value, empty	
	<b>4111</b> invalid_r inva - The API <b>303</b> equest lid was used ope with an rati invalid on ending path value	Repeat the request using a valid path
<a href="#">4112 / lookup_jwt</a>	<b>4112</b> invalid_r inva <b>000</b> equest lid	
	<b>4112</b> invalid_r inva - The <b>103</b> equest lid request did para not include met headers ers and/or parameters as specified for the API - The request included duplicate parameters - The request included required headers or parameters but without value, empty	Repeat the request including all required parameters and/or headers
	<b>4112</b> invalid_r inva - The API <b>303</b> equest lid was used ope with an rati invalid on ending path value	Repeat the request using a valid path

Group/ API_ID	Code Category Info Reasons	How to resolve
4113/ remove_jwt	<b>4113</b> invalid_r inva <b>000</b> equest lid	<p><b>4113</b> invalid_r inva - The request did not have all required parameters and/or headers.</p> <p><b>103</b> equest lid - The request included duplicate parameters.</p> <p>The request included required headers or parameters but without value, empty.</p>
	<b>4113</b> invalid_r inva - The API was used with an invalid path value.	Repeat the request using a valid path
4200/ delete_expired_session	<b>4200</b> invalid_r inva <b>000</b> equest lid	<p><b>4200</b> invalid_r inva - The request did not have all required parameters and/or headers.</p> <p><b>103</b> equest lid - The request included duplicate parameters.</p> <p>The request included required headers or parameters but without value, empty.</p>

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
				duplicate parameters - The request included required headers or parameters but without value, empty	
	<b>4200</b>	invalid_r	inva	- The request did not use mutual SSL	Repeat the request using mutual SSL
	<b>204</b>	equest	lid	requester	
				req did not use mutual ssl	
	<b>4200</b>	invalid_cl	inva	- The given client certificate is unknown	Repeat the request using valid credentials. If the error still occurs contact the system administrator
	<b>205</b>	ient	lid	client certificate is unknown	
				certif	
				icat	
				e	
	<b>4200</b>	invalid_r	inva	- The API path was used with an invalid ending path value	Repeat the request using a valid path
	<b>303</b>	equest	lid	was used with an invalid ending path value	
4201/ delete_session	<b>4201</b>	invalid_r	inva		
	<b>000</b>	equest	lid		
	<b>4201</b>	invalid_r	inva	- The request did not include required parameters and/or headers	Repeat the request including all required parameters and/or headers
	<b>103</b>	equest	lid	request did not include required headers and/or specified parameters as specified for the API	
				- The request included duplicate parameters	
				- The request	

Group/ API_ID	Code Category Info Reasons	How to resolve
	included required headers or parameters but without value, empty	
	<b>4201</b> invalid_r inva - The API <b>303</b> equest lid was used ope with an rati invalid on ending path value	Repeat the request using a valid path
<a href="#">4202 / get_session</a>	<b>4202</b> invalid_r inva <b>000</b> equest lid	
	<b>4202</b> invalid_r inva - The <b>103</b> equest lid request did para not include met headers ers and/or parameters as specified for the API - The request included duplicate parameters - The request included required headers or parameters but without value, empty	Repeat the request including all required parameters and/or headers
	<b>4202</b> invalid_r inva - The API <b>303</b> equest lid was used ope with an rati invalid on ending path value	Repeat the request using a valid path

Group/ API_ID	Code Category Info Reasons	How to resolve
4203/ create_session	<p><b>4203</b> invalid_r inva  <b>000</b> equest lid</p> <p><b>4203</b> invalid_r inva - The request did not include all required parameters and/or headers</p> <p><b>103</b> equest lid - The request included duplicate parameters</p> <p>The request included required headers or parameters but without value, empty</p>	Repeat the request including all required parameters and/or headers
	<p><b>4203</b> invalid_r inva - The API path was used with an invalid value</p> <p><b>303</b> equest lid - The request included required headers or parameters but without value, empty</p>	Repeat the request using a valid path
5000/ validate_client	<p><b>5000</b> invalid_r inva - The request did not include all required parameters and/or headers</p> <p><b>103</b> equest lid - The client_key parameter is missing</p>	Repeat the request including all required parameters and/or headers
	<p><b>5000</b> invalid_s inva - the given scope is not valid</p> <p><b>115</b> cope lid - The scope requested by the client was not valid</p>	Repeat the request using valid scope values
		Verify that the client has the expected status

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
	<b>5000</b>	invalid_r	inva	- the	
	<b>126</b>	equest	lid	expected	
			stat	client	
			us	status is	
			not	the	
			actual	one	
	<b>5000</b>	invalid_r	expi	- The client	Check if your requested values are correct
	<b>127</b>	equest	red	id has	
			clie	expired	
			nt	- The client	
			id	id has been	
			removed		
			due to its		
			expiration		
	<b>5000</b>	invalid_cl	inva	- The	Check if your requested values are correct
	<b>128</b>	ient	lid	client_nam	
			clie	e is invalid	
			nt		
			nam		
			e		
	<b>5000</b>	invalid_r	inva	- The	Check if the environment for the client id is
	<b>129</b>	equest	lid	environme	configured as expected
			envi	nt is not	
			ron	valid	
			men	- The client	
			t	has been	
			configured		
			for a		
			different		
			one		
	<b>5000</b>	invalid_r	inva	- The client	Check if the client type for the client is
	<b>130</b>	equest	lid	type is not	configured as expected
			clie	the actual	
			nt	one	
			type		
	<b>5000</b>	invalid_r	inva	- The	Check if your requested values are correct
	<b>201</b>	equest	lid	client_id is	
			clie	a master-	
			nt	key	
			- The		
			client_secr		
			et	is invalid	
	<b>5000</b>	invalid_r	inva	- The	Repeat the request using mutual SSL
	<b>204</b>	equest	lid	requester	
			req	did not use	
			uest	mutual ssl	
	<b>5000</b>	invalid_cl			
	<b>205</b>	ient			

Group/ API_ID	Code Category Info Reasons	How to resolve
	inva - The given lid client clie certificate nt is unknown certi ficat e	Repeat the request using valid credentials. If the error still occurs contact the system administrator
5001/ validate_token	<b>5001</b> invalid_r inva <b>000</b> equest lid	
	<b>5001</b> invalid_r inva - The <b>103</b> equest lid request did parameters and/or headers para not include met headers ers and/or parameters as specified for the API - The request included duplicate parameters - The request included required headers or parameters but without value, empty	Repeat the request including all required parameters and/or headers as specified for the API - The request included duplicate parameters - The request included required headers or parameters but without value, empty
	<b>5001</b> invalid_r inva - The <b>204</b> equest lid requester req did not use uest mutual ssl	Repeat the request using mutual SSL
	<b>5001</b> invalid_cl inva - The given <b>205</b> ient lid client clie certificate nt is unknown certi ficat e	Repeat the request using valid credentials. If the error still occurs contact the system administrator
	<b>5001</b> invalid_r inva <b>990</b> equest lid acce	Request a new token

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
				ss_t - The token oke has expired n - The token does not exist	
	<b>5001</b>	invalid_r	inva	- The token	Request a new token that is scoped
	<b>991</b>	equest	lid	has not acce been ss_t granted for oke the n required SCOPE	accordingly
	<b>5001</b>	invalid_r	inva	- No	Use the access_token either within the
	<b>992</b>	equest	lid	access_tok acce en was ss_t included in oke the request n - An access_tok en was included more than once	authorization header, as query parameter or as post body parameter
	<b>5001</b>	invalid_r	inva	- The token	The token (client) has to be enabled or revoked
	<b>993</b>	equest	lid	is disabled	and a new one requested. toke which n means that associated client is disabled
<a href="#">validate_refresh_token</a>	<b>5002</b>	invalid_r	inva	- The	Repeat the request including all required
	<b>103</b>	equest	lid	request did para not include met headers ers and/or parameters as specified for the API - The request included duplicate parameters - The request included required	parameters and/or headers

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
				headers or parameters but without value, empty	
	<b>5002</b>	invalid_s	inva	- No scope	Repeat the request using valid scope values
	<b>115</b>	cope	lid	value	
			sco	matched a	
			pe	valid one	
			was	for this	
			req	refresh_tok	
			uest	en	
			ed		
	<b>5002</b>	unauthor	una	- The	Repeat the request using a valid client
	<b>117</b>	ized_clie	uth	refresh_tok	
			nt	oriz en has	
			ed	been	
			clie	issued to a	
			nt	different	
			for	client_id	
			this		
			req		
			uest		
	<b>5002</b>	invalid_r	inva	- The	Repeat the request using mutual SSL
	<b>204</b>	equest	lid	requester	
			req	did not use	
			uest	mutual ssl	
	<b>5002</b>	invalid_cl	inva	- The given	Repeat the request using valid credentials. If
	<b>205</b>	ient	lid	client	the error still occurs contact the system
			clie	certificate	administrator
			nt	is unknown	
			certi		
			ficat		
			e		
	<b>5002</b>	invalid_r	inva	- The token	Request a new token
	<b>990</b>	equest	lid	has expired	
			acce	- The token	
			ss_t	does not	
			oke	exist	
			n		
	<b>5002</b>	invalid_r	inva	- The token	The token (client) has to be enabled or revoked
	<b>993</b>	equest	lid	is disabled	and a new one requested.
			toke	which	
			n	means that	
				associated	
				client is	
				disabled	

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
5003/ token_revocation	<b>5003</b> <b>103</b>	invalid_r equest	inva lid	- The request did not include required parameters and/or headers met headers and/or specified for the API - The request included duplicate parameters - The request included required headers or parameters but without value, empty	Repeat the request including all required parameters and/or headers
	<b>5003</b> <b>117</b>	unauthor ized_clie	una uth	- The token has been issued to a different client_id	Repeat the request using a valid client
	<b>5003</b> <b>118</b>	unsuppo rted	uns upp	- The token_type_hint values for token_type must be either 'access_token' or 'refresh_token'	Repeat the request using the correct value for token_type_hint
	<b>5003</b> <b>204</b>	invalid_r equest	inva lid	- The requester did not use mutual ssl	Repeat the request using mutual SSL

Group/ API_ID	Code	Category	Info	Reasons	How to resolve
	<b>5003</b>	invalid_client	inval	- The given client certificate is unknown	Repeat the request using valid credentials. If the error still occurs contact the system administrator
	<b>205</b>	invalid_request	lid	- The request did not include required parameters and/or headers	
5004/ validate_id_token, create_id_token	<b>5004</b>	invalid_request	inval	- The request included duplicate parameters	Repeat the request including all required parameters and/or headers
	<b>103</b>	unauthorized_client	uth	- The request included required headers or parameters but without value, empty	
	<b>5004</b>	unauthorized_client	una	- The token has been issued to a different client recipient	Repeat the request using a valid client or request a valid JWT
	<b>117</b>	invalid_id_token	lid	- The id_token has expired	
	<b>120</b>	invalid_request	id_t	- The id_token has missing claims	Repeat the request including a valid id_token

Group/ API_ID	Code Category Info Reasons	How to resolve
	<b>5004</b> invalid_r inva - The JWT <b>121</b> equest lid has an JWT invalid signature - The JWT has been created with a different shared secret	Repeat the request using a valid JWT
	<b>5004</b> invalid_r inva - If an <b>122</b> equest lid access_tok creating the JWT/ id_token JWT en was / given a id_t nonce has oke to be n available req also uest - If the id_token had other missing claims - If the JWT's signature could not be created	Repeat the request including valid values for creating the JWT/ id_token
	<b>5004</b> invalid_r inva - If an <b>131</b> equest lid access_tok creating the id_token id_t en was oke given a n nonce has req to be uest available also	Repeat the request including valid values for creating the id_token
	<b>5004</b> invalid_r inva - The <b>204</b> equest lid requester req did not use uest mutual ssl	Repeat the request using mutual SSL
	<b>5004</b> invalid_cl inva - The given <b>205</b> ient lid client the error still occurs contact the system clie certificate administrator nt is unknown certi ficat e	Repeat the request using valid credentials. If the error still occurs contact the system administrator