
Clearstream API Developer Guide

August 2025

Change log

3 June 2021		First draft
10 September 2021	0.1	SSL extraction details added
29 September 2021	0.2	Rework for Preview
11 November 2021	0.3	First version for publication
25 November 2021	0.4	Rebranding
8 February 2022	0.5	Simplification of DBP integration
3 June 2022	0.6	Clarifications on API onboarding + Additional troubleshooting & support sections
20 June 2022	0.7	Removal of DBP user ID references, SSL cert/keys scripts clarified
23 August 2022	0.8	Clarifications on API onboarding + November certificate changes for API consumers
23 September 2022	0.9	Support page improvements
4 November 2022	1.0	Rework for November 2022 API roll-out
15 November 2022	1.1	Adding additional information on CA certificates
20 August 2025	2.1	CA certificates download

Introduction

This developer guide provides an overview on how to discover and integrate with Clearstream APIs as well as how to access and use the Clearstream API setup. The current infrastructure is powered by:

- Deutsche Börse Digital Business Platform (DBP) – The API developer platform and catalogue
- Clearstream XACT – The Clearstream client facing web platform
- Clearstream API Platform – The API component of Xact

This guide is targeted at developers as well as technical project managers and software architects interested in building systems using Clearstream APIs. This guide assumes a basic understanding of various API related topics such as HTTP, REST, OAuth2.0, mutual TLS, etc.

It must be noted that due to the sensitive nature of financial APIs, Clearstream APIs require additional setup steps which are detailed in the “Pre-requisites” chapter of this document.

The “Example code” section provides a script based example which illustrates how to build using Clearstream APIs. OpenAPI specifications are always published to the digital business platform in their most recent version. Some APIs reference additional documentation through their description.

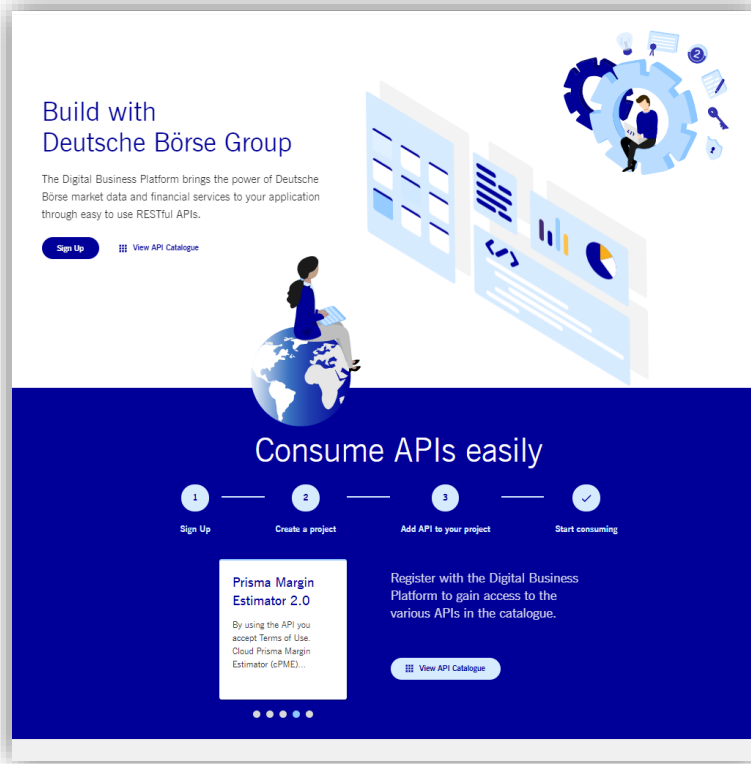
Table of Contents

Change log	2
Introduction.....	1
Table of Contents	2
Pre-requisites	3
Digital Business Platform	3
Xact Web Portal.....	4
Xact Web Portal subscription & MT599 API onboarding messages.....	4
Step 1 – Xact API Consumer Creation	5
Step 2 – Xact API Consumer Creation	6
Getting Started.....	7
Clearstream API Playground	7
Certificate Chains, Certificate Authorities (CAs), Truststores & more.....	8
Getting an OAuth2.0 Bearer Token (access_token)	10
Accessing the Playground API	11
Troubleshooting.....	12
1) I cannot get my OAuth2.0 client to work! How does the support page work?.....	12
2) How do I convert the Xact API consumer PEM files to P12 files?.....	13
3) When requesting a bearer token (access_token), HTTP 4xx errors (client-errors) are returned.....	13
4) Using curl I am unable to retrieve an access_token and I get SSL/TLS errors.	14
5) Mutual TLS is not part of the OAuth2.0 spec and is not supported by my library	14
6) How to troubleshoot and/or debug with CURL commands	14
How to get development support from Clearstream?.....	15
Example code.....	17
Bash / CURL	17

Pre-requisites

Digital Business Platform

To browse through the Clearstream API catalogue developers must consult the digital business platform (DBP). <https://console.developer.deutsche-boerse.com>.



This platform serves as a DBG-wide API inventory and presents all available Clearstream APIs, their descriptions, points of contact as well as Swagger / OpenAPI specifications¹.

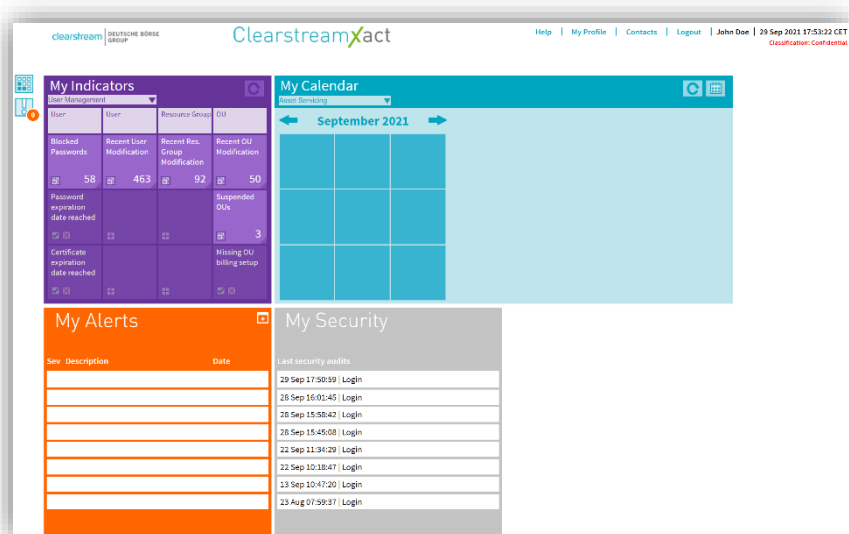
¹ Clients should note that for the pre-production / UAT / OCCT environment (<https://api-t2s-test.clearstream.com/>), a staging deployment of the DBP is available under: <https://console.cstest.dbpapi.com>

Xact Web Portal

This section presents the required Xact Web Portal setup as well as the API consumer credentials setup mandatory for any successful API call.

Xact Web Portal subscription & MT599 API onboarding messages

Clearstream Xact Web Portal is a web-based, connectivity channel offering online access to a variety of post-trade services and more. In addition, Xact Web Portal provides access management for Clearstream APIs published on the Digital Business Platform. By design, Clearstream APIs are treated as business resources and are therefore protected by fine grained user access control mechanisms and API credentials.



Existing Xact Web Portal clients must send an **MT599 Swift** message to the Clearstream address **CEDELULLXXX, (attn.: PRGConnect)** to enable the Clearstream API service for their Organisation Unit (OU). In this message, a request for one or more specific API resources can be included. At least the Clearstream API Playground should be requested for an initial onboarding in production.

The full list of available API resources (including descriptions) is available in the public [DBP API Catalogue](#).

The below example shows the required format of this MT599 Swift message. Clients should make sure that the message contains only Swift compliant characters!

```
ATTN: PRGConnect
SUBJECT: Clearstream APIs
Please add the Clearstream API service
and link the Clearstream API Playground resource
to our Xact OU xxxxxx
Contact details: John Doe, john.doe(at)xxxxxx.xyz, 00123456789
```

For clients already having the Clearstream API service added in their OU, the following shortened MT599 message can be used instead of the above:

```
ATTN: PRGConnect
SUBJECT: Clearstream APIs
Please link the <API-NAME-FROM-DBP-CATALOGUE> API resource
to our Xact OU xxxxxx
Contact details: John Doe, john.doe(at)xxxxxx.xyz, 00123456789
```

Future clients as well as other interested parties are advised to contact the Clearstream Connectivity Helpdesk directly for additional information about how to setup (connect@clearstream.com).

Step 1 – Xact API Consumer Creation

API consumer credentials can be created by following the standard Xact Web Portal user creation procedure². The user type must be set to “API Consumer”³. In the **first step** of the API Consumer creation, the grants of the technical user are configured:

The screenshot shows the 'Create user' form in the Xact Web Portal. The form is titled 'User' and includes fields for personal information (First name, Middle name, Last name, User deputy, Email address), external identifiers (External user ID, Telephone number, Mobile number, Fax number), and user type (API Consumer). Below these fields is the 'Grant Selection' section, which shows a table of available grants. The 'Clearstream APIs' section is expanded, showing a table with columns for Functional Roles, Resource Groups, and Resources. The 'Xact Web Portal User Management API (prod)' is selected. At the bottom, there is a 'User properties' table with columns for Service name, Property name, Category, Updatable by, and Value. The table lists various properties like Smart Card, Software Key Store, N-Eyes, User Time Zone, Preferred Language, and DBP User Id. The form has a 'Reset' button and a 'Submit' button.

In the grant-selection component, an API consumer can be granted one or more Clearstream APIs.

In addition to this general coarse-grained API access, depending on the API it might be necessary to grant specific more fine-grained roles. For example, for most user management API calls to succeed, the SCIM Admin or SCIM Read-Only roles are required in addition to the Xact Web Portal User Management API role.

The screenshot shows the 'User Management' grant selection component. It displays a list of roles with checkboxes. The 'SCIM Admin' role is selected, and the 'SCIM Read-Only' role is also selected.

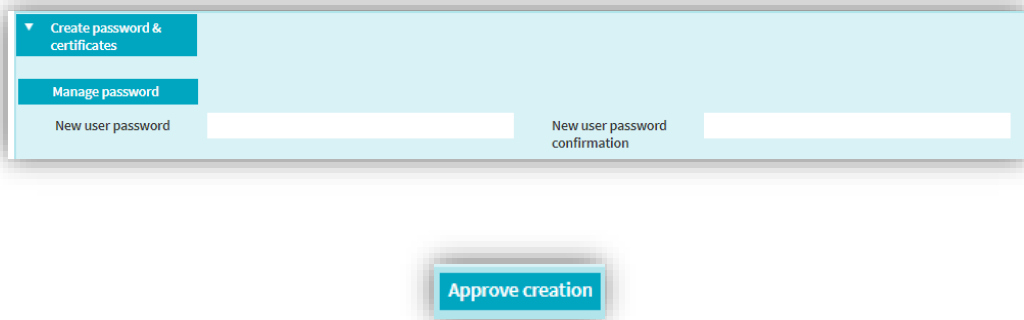
² See: <https://www.clearstream.com/resource/blob/1311454/6a3813957a058c3138ba6fcdeb6dde71/xactusermanual-en-data.pdf>

³ Xact Web Portal OUs need to be granted access to the Clearstream API service and must be configured with at least two valid administrators to create “API Consumer” users.

However, some APIs (such as the Clearstream API playground) do not need additional fine-grained access management. The DBP documentation describes which API endpoint offers which roles and how they can be used.

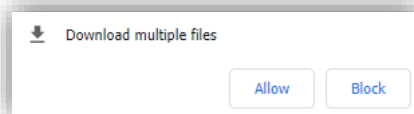
Step 2 – Xact API Consumer Creation

The second step of the API Consumer creation deals with credential generation. After submitting the future API consumer password, **private keys will be locally generated in the browser** for which certificates (that is, CSRs) are then generated and returned. This results in the download of two PEM files containing certificates and keys.



The screenshot shows a web form titled 'Create password & certificates'. It has a sidebar with two tabs: 'Create password & certificates' (selected) and 'Manage password'. The main area contains two input fields: 'New user password' and 'New user password confirmation'. Below the form is a blue button labeled 'Approve creation'.

Your browser might prevent the simultaneous download the 2 PEM files. In this case, the download needs to be confirmed. Google Chrome shows the below pop-up:



2 PEM files will be downloaded containing the:

- ✓ Signing certificate and key
- ✓ **SSL (TLS) certificate and key**

The naming convention of the files is "**ocapi-{USERID}-{TYPE}.pem**". The more important file is the SSL file as it allows establishing a mutual TLS connection with <https://api.clearstream.com> and retrieving OAuth 2.0 tokens as described in the next section.

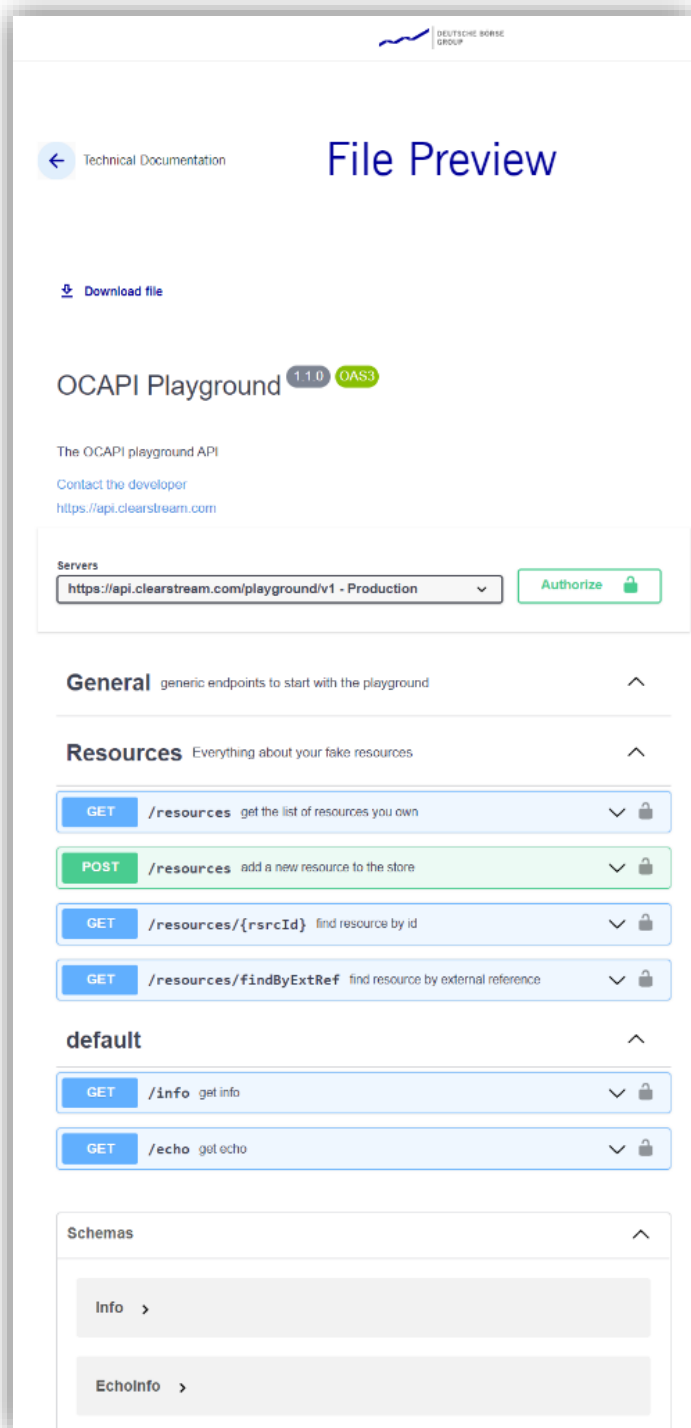


Note: Your certificates will expire after 2 years and need to be replace beforehand. Xact OU administrators can view the certificate expiry date in the credential details screen of the API consumers.

Getting Started

Clearstream API Playground

It is recommended to start with the “[Clearstream API Playground](#)” which offers a variety of synthetic endpoints and is provided free of charge. The OpenAPI specification of this API is available on the DBP portal under the “Technical Documentation” tab.



Certificate Chains, Certificate Authorities (CAs), Truststores & more...

To securely communicate with the Clearstream API Platform it is mandatory to add the necessary CACertificates into your certificate authority chain or truststore.

Running in "insecure mode" without proper the certificate authority chain or truststore defined is not supported. Additionally, any form of certificate pinning is not supported. This includes adding Clearstream server certificates to the truststore.

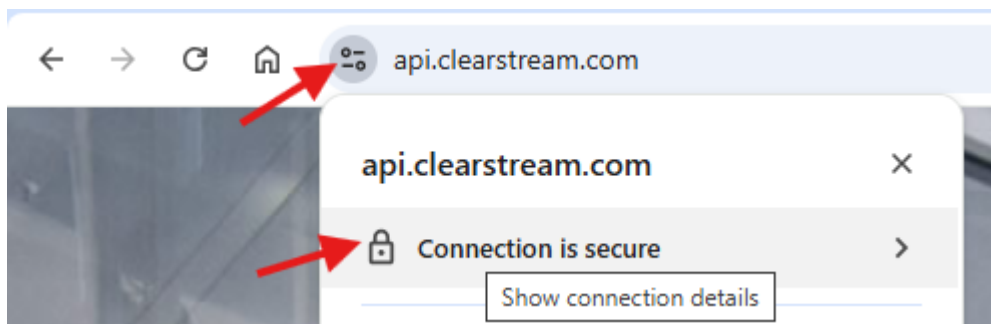
Note: Clearstream reserves the right to update API server certificates at any point in time with no prior announcement.

Server CA Certificates

You can download the server CA certificates using the following procedure:

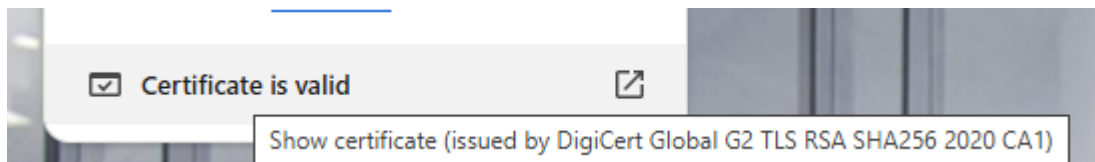
→ <https://api.clearstream.com>

click on the icon to the left of the URL:

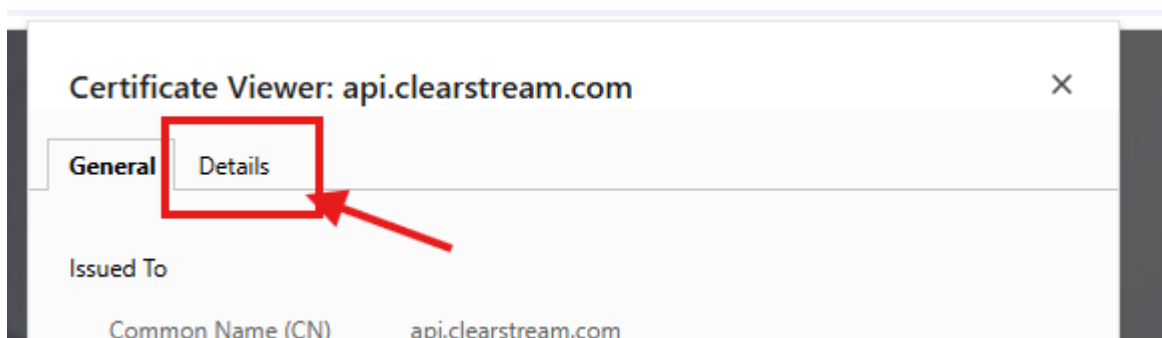


Then click on the "Connection is secure"

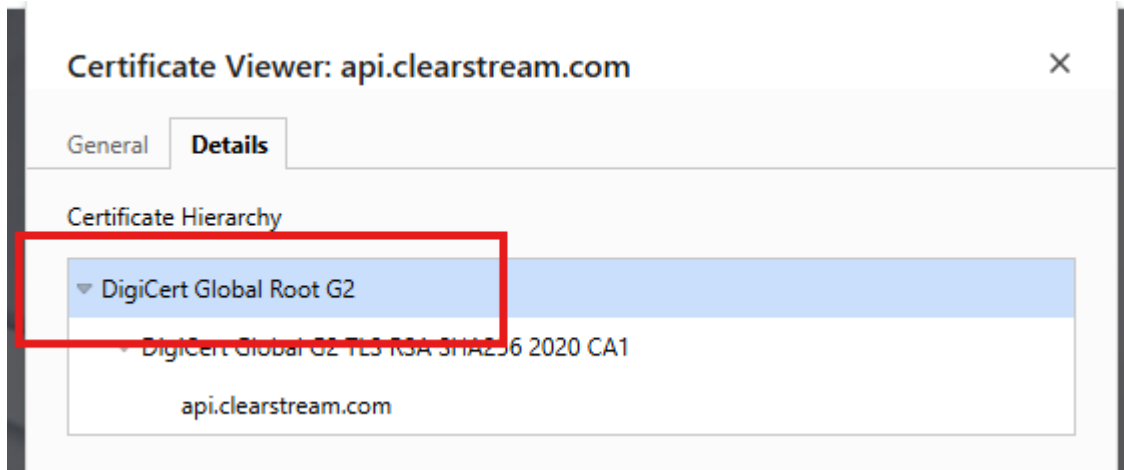
Click on "Certificate is valid":



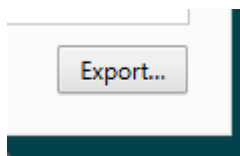
On the certificate viewer click on "Details":



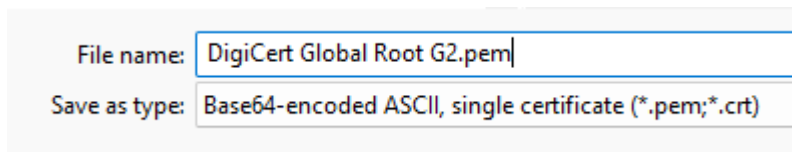
Click on the DigiCert Global Root G2, the entry must be highlighted as below:



Click on the "Export" button:



Save the file as ".pem":



The file contains the **DigiCert** root CA and can be used for the following API client use cases:

- For CURL, it is sufficient to provide this the above file using the "--cacert" parameter
- For JAVA, the certificates from the above file should be added to a P12 or JKS truststore which is then referenced / used in the client HTTPS code.

Client CA Certificates

Note: This section can be ignored in most cases!

The Clearstream API Platform uses mutual TLS for all API calls. As a result, certificates are issued to API consumers during the API consumer creation. If you need to validate the certificate chain of the API consumer certificate, then the client CA certificates are available using the below download link.

→ <https://api.clearstream.com/client-cert-cacerts.pem>

The verification of the client certificate is not necessary for most use cases. For example, CURL trusts client certificates and keys without having the full certificate chain.

Getting an OAuth2.0 Bearer Token (access_token)

Before calling the “playground” API, an OAuth2.0 bearer token needs to be requested from the Xact authorisation and authentication server. Xact has implemented the OAuth 2.0 resource owner credentials flow⁴ but has hardened it by enforcing a strict mutual TLS connection (SSL certificate/key) in addition to a consumer/user password⁵.

To request an OAuth 2.0 access token the mutual TLS curl commands below can be used. Please note that the variables in the below snippet need to be initialised correctly. When requesting a token, the desired scopes need to be specified.

```
curl
--cert ${SSL_PEM}
--cacert ${CA_PEM}
--data "grant_type=password"
--data "scope=allow ocapi-playground-v1"
--data "username=${USERNAME}"
--data "password=${LOGIN_PASSWORD}"
https://api.clearstream.com/authmanager/oauth2/access_token)
```

Running the above command yields a JSON response containing the **JWT access_token**, the granted scopes as well as a refresh token.

```
{
  "access_token": "eyJ0eXAiOiJKV1QiLCJ...\"",
  "refresh_token": "eyJ0eXAiOiJKV1QiLCJ...\"",
  "scope": "allow ocapi-playground-v1",
  "token_type": "Bearer",
  "expires_in": 3599
}
```

The basic authentication header required for the OAuth 2.0 client identification is not required by the Clearstream API infrastructure as clients are instead identified through their mutual TLS and consumer/user login password credentials sent via form data. However, setting this OAuth 2.0 basic-auth header to any random placeholder value is allowed and will simply be ignored by the Clearstream API infrastructure⁶.

⁴ Sometimes referred to as the OAuth2.0 password flow.

⁵ Furthermore, an underlying authentication algorithm paired with a scope validator ensure the validity of any token requests before tokens are issued. Finally, the Clearstream API configuration comes with coarse grained access control to API endpoints based on the token scopes that have been granted.

⁶ For example, in case your OAuth 2.0 library requires you to specify a client id and password.

Accessing the Playground API

Once a valid bearer token (access_token) has been obtained, calling API endpoints is straight forward but still requires a mutual TLS connection.

```
curl
--cert ${SSL_PEM}
--cacert ${CA_PEM} \
--header 'Authorization: Bearer ${ACCESS_TOKEN}'
https://api.clearstream.com/playground/v1/info
```

For the above playground info endpoint, the following JSON response is returned.

```
{
  "info_str": "Welcome to OCAPI Playground API ... ",
  "version": "1.221101XP.12"
}
```

Note:

- A full CURL based BASH example is provided in the "Example Code" chapter.
- In case of errors, more information can be found in the "Troubleshooting" section of this document.

Troubleshooting

1) I cannot get my OAuth2.0 client to work! How does the support page work?

If there are problems with your OAuth2.0 client, please consult the [Clearstream API support](#) page which provides a working web client (see screenshot below).

Analysing the network traffic between this web client and the Clearstream API platform with your Browser's development tools is recommended to understand how API calls should be performed. To use this client, the API consumer must be granted access to at least the Clearstream API Playground as detailed in the "Pre-Requisite" chapter.

The SSL certificate and key from the Xact credentials P12 must be imported into the browser. To convert your PEM files to P12 files for import into your browser or operating system, clients should use the [Clearstream API credential support page](#).

Finally the support page also provides the download links for the server and client CACerts.

The screenshot shows a web client interface for the Clearstream API Platform. The header says "Welcome to the Clearstream API Platform". Below this, there is a section for instructions: "This page allows testing API access using Xact API consumer credentials (v4.2211XP.14)" followed by four bullet points: "API access must be granted to the API consumer in Xact", "You must import your SSL client certificate in your browser (Mandatory mutual TLS)", "Refer to the API credential support page to convert between PEM and P12 files", and "Only if required download Server-CACerts here. (Download Client-CACerts here)".

The interface is divided into two main columns. The left column contains input fields for "Token Server:" (https://api.clearstream.com), "Token Endpoint:" (/authmanager/oauth2/access_token), "API Consumer ID:", "Password:", and "Scopes:" (allow ocapi-playground-v1). Below these is a "Request Access Token" button. Further down are fields for "Token type:", "Access Token:", "Refresh Token:", "Token scopes:", and "Token expiry:". At the bottom left is a "Back" button.

The right column contains a link to the "DBG Digital Business Platform API catalogue". Below this are input fields for "API Server:" (https://api.clearstream.com), "API Endpoint:" (/playground/v1/info), and "Query params:". The "Method:" is set to "GET". The "Request headers:" section shows "Content-type: application/json; charset=UTF-8". The "Request body:" section is a large empty text area. At the bottom right is a "Call API Endpoint" button.

2) How do I convert the Xact API consumer PEM files to P12 files?

To import your certificates and keys into your browser or operating system it might be necessary to convert the PEM files to P12 files. For this the [Clearstream API credential support page](#) provides a converter which allows going from PEM to P12 and vice-versa. This converter is also helpful to generate P12 files for easy integration with Java applications supporting only the P12 format.

Welcome to the Clearstream API Platform

This converter page allows you to convert between PEM and Standard P12 files. (3DES encrypted)
Please note that the above-mentioned P12 files can be imported into most applications, browsers and operating systems.
For API integrations, we recommend the use of PEM files whenever possible.
Please note: This converter runs locally in your Browser. No data is exchanged with Clearstream. Clearstream has no access to your keys at any point in time!

A) Convert PEM to Standard P12 (3DES)

PEM Certificate & Key:

B) Convert Standard P12 (3DES) to PEM

Standard P12: No file chosen

3) When requesting a bearer token (access_token), HTTP 4xx errors (client-errors) are returned

Please make sure your request goes through <https://api.clearstream.com> or the equivalent API portal for test environments. Any OAuth requests sent to <https://xact.clearstream.com> will be rejected.

When requesting a bearer token via <https://api.clearstream.com>, valid scopes must be specified. If invalid scopes are requested, then the bad request error response will contain a list of valid scopes. If this does not solve your issue, please continue with the next troubleshooting step.

If any of the other parameters in the token request is incorrect, a 403 Forbidden error is returned by the authorisation server. Details on what exactly has led to the error can be viewed in the audit log of the Xact OU. This feature is only available to granted Xact OU administrators.



clearstream DEUTSCHE BÄNKE GROUP Audit Log

Query / List 03/06/2021 16:05:55

General

First name	apico4	Service name	Security
Last name	apico4	Event type	Action
User ID	a00012345_1001234	Event type	Failed authentication attempt
Creation timestamp	03/06/2021 16:05:55	Resource	

Details

Key	Value (English)
Browser	curl/7.76.1
IP address	42.4221.21.42
Reason	X-DBP-APIKEY header value does not match IAM user property value

3 Rows | 10,000 Rows per page

4) Using curl I am unable to retrieve an access_token and I get SSL/TLS errors.

Certain older versions of curl are not supported by the Clearstream API infrastructure. Please try upgrading curl to a more recent version.

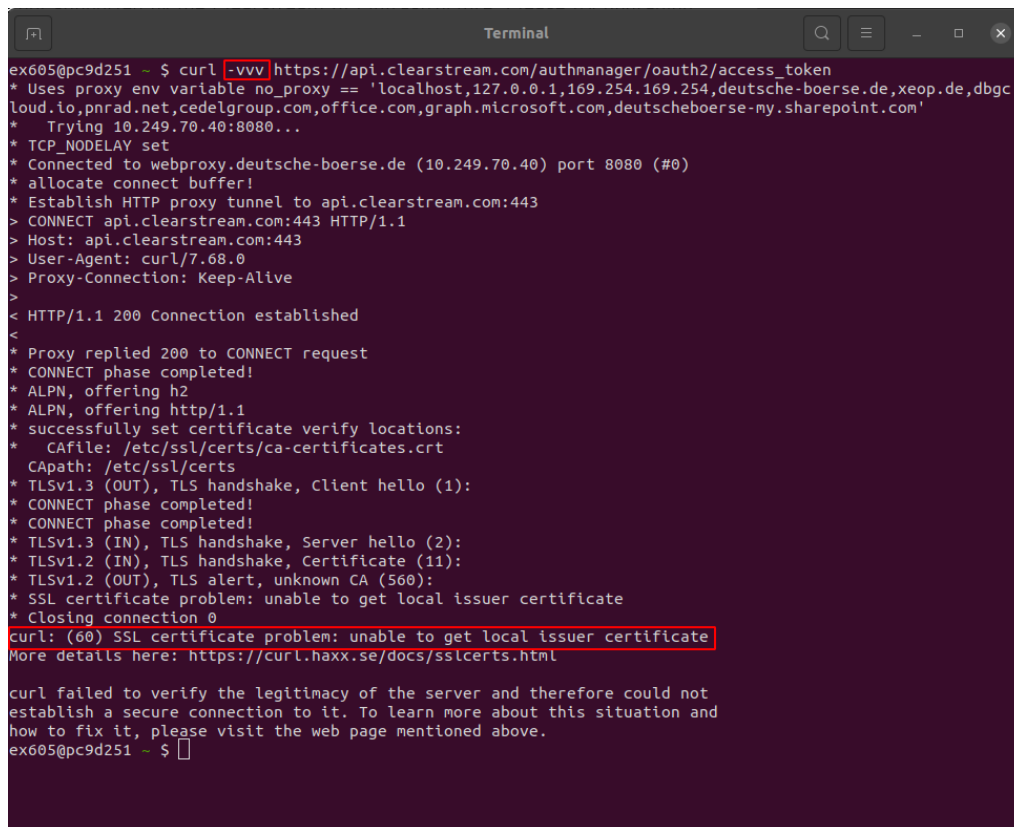
5) Mutual TLS is not part of the OAuth2.0 spec and is not supported by my library

The current OAuth2.0 specification does not deal with mutual TLS. If you need to use a specific library which does not support mutual TLS, and you are not able to modify the HTTP connection accordingly, then we recommend installing a proxy in between your client and our infrastructure. This proxy should then wrap the library's one-way TLS into mutual TLS (two-way) before contacting the Clearstream API infrastructure.

6) How to troubleshoot and/or debug with CURL commands

When using CURL commands to troubleshoot or debug API connectivity issues it is important to use the verbose mode. In this mode additional debug information is provided which is often necessary to fully understand the underlying issue.

The **below invalid example** illustrates how a verbose CURL command (-vvv option) shows a successful proxy connection followed by a SSL error caused by the fact that no client certificate was specified.

A terminal window titled "Terminal" showing a verbose curl command. The command is: curl -vvv https://api.clearstream.com/authmanager/oauth2/access_token. The output shows the connection to a proxy at 10.249.70.40:8080, successful connection, and then an SSL handshake. The error message is: "curl: (60) SSL certificate problem: unable to get local issuer certificate". The terminal also shows the path to the CA file and the certificate path. The error message is highlighted with a red box. The terminal also shows the path to the CA file and the certificate path. The error message is highlighted with a red box. The terminal also shows the path to the CA file and the certificate path. The error message is highlighted with a red box.

```
ex605@pc9d251 ~ $ curl -vvv https://api.clearstream.com/authmanager/oauth2/access_token
* Uses proxy env variable no_proxy == 'localhost,127.0.0.1,169.254.169.254,deutsche-boerse.de,xeop.de,dbgc
loud.io,pnrad.net,cedelgroup.com,office.com,graph.microsoft.com,deutscheboerse-my.sharepoint.com'
* Trying 10.249.70.40:8080...
* TCP_NODELAY set
* Connected to webproxy.deutsche-boerse.de (10.249.70.40) port 8080 (#0)
* allocate connect buffer!
* Establish HTTP proxy tunnel to api.clearstream.com:443
> CONNECT api.clearstream.com:443 HTTP/1.1
> Host: api.clearstream.com:443
> User-Agent: curl/7.68.0
> Proxy-Connection: Keep-Alive
>
< HTTP/1.1 200 Connection established
<
* Proxy replied 200 to CONNECT request
* CONNECT phase completed!
* ALPN, offering h2
* ALPN, offering http/1.1
* successfully set certificate verify locations:
*  CAfile: /etc/ssl/certs/ca-certificates.crt
*  CAPath: /etc/ssl/certs
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
* CONNECT phase completed!
* CONNECT phase completed!
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (OUT), TLS alert, unknown CA (560):
* SSL certificate problem: unable to get local issuer certificate
* Closing connection 0
curl: (60) SSL certificate problem: unable to get local issuer certificate
More details here: https://curl.haxx.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.
ex605@pc9d251 ~ $
```

Note: A correct working example including all required parameters would look like this:

```
curl
--cert ocapi-a000012345_00012345-ssl.pem
--cacert ocapi-a000012345_00012345-cacert.pem
--data "grant_type=password"
--data "scope=allow ocapi-playground-v1"
--data "username=a000012345_00012345"
--data "password=*****" https://api.clearstream.com/authmanager/oauth2/access_token
```

How to get development support from Clearstream?

In case the above troubleshooting steps did not solve your issue and **before contacting** connect@clearstream.com for additional help, please run through the following checklist.

- 1) Visit “My OU Details” and check that the Clearstream API service is granted to your OU.
Check that you have at least one Clearstream API resource linked in the Relationships of your OU.

clearstream | DEUTSCHE BÖRSE GROUP

My OU Details

Details

OU name	My Customer OU	OU type	Customer
Parent OU	Clearstream Entity OU	OU status	Activated
Pending action	None		

OU services

Service name	Clearstream APIs
	Message Exchange
	Settlement
	Cash & Liquidity

Relationships

Relationship type	Relationship name	Entity name
Clearstream API	Xact Web Portal User Management API (prod)	CBL
Clearstream API	Clearstream API Playground (prod)	CBL

- 2) Check that your API consumer user has valid credentials in Xact.
Check that at least one API resource such as the Clearstream API playground is assigned to the API Consumer user.

clearstream | DEUTSCHE BÖRSE GROUP

User

Query / List **Create** **API Consumer**

Details

First name *	API	User ID	a01044330_1000149
Last name *	Consumer	External user ID	
Middle name		User type *	API Consumer
User deputy		User status	Activated
Email address *	api.consumer@example.com	Pending action	None
Telephone number *	123456	Fax number	
Mobile number		Last update timestamp	11/03/2022 14:46:50
Creation timestamp	11/03/2022 14:46:27	Last login	11/03/2022 15:04:05

Credentials information

Credential type	Status	Pending action	Credential expiry timestamp	Serial number SSL	Serial number SIGN	Last update timestamp	Last update user Id
Software Certificate	Activated	None	11/03/2024 14:47:32	1151210352	1151210354	11/03/2022 14:47:33	a01044330_1000149
Password	Activated	None	11/06/2022 14:47:29			11/03/2022 14:47:29	a01044330_1000149

Grant Summary

Clearstream APIs

Clearstream API Playground (prod)

Functional Roles	Resource Groups
API Access	Clearstream API Playground (prod)

- 3) With the SSL key and certificate perform a CURL command against https://api.clearstream.com/authmanager/oauth2/access_token to obtain an access token. In this CURL command check that all required information is provided including the grant_type, scope, username and password. A complete example is available in the "Getting an OAuth2.0 Bearer Token (access_token)" section.
- 4) With the SSL key and certificate perform a CURL command against <https://api.clearstream.com/playground/v1/info> to ensure the obtained access token is working correctly.
- 5) With the SSL key and certificate perform a CURL command against the Clearstream API you are trying to use. In case of errors please ensure that the correct scope was specified in step 3.

If the above checklist did not reveal the problem, please communicate at which step your issue occurs and additionally **you must provide the following debugging information:**

- Timestamps of when you encountered the issue;
- The command history in case you execute CURL or scripting commands; (**verbose** mode – see troubleshooting section);
- Code snippets illustrating your API usage;
- HTTP requests and responses send/receive to/from the Clearstream API platform.

Example code

This section provides a scripting-based example for illustration purposes. Clearstream Banking does not take any responsibility in case of issues, problems or damages caused by the below example code.

Clearstream Banking encourages developers to always follow best practices and to perform regular penetration tests for any systems integrating with the Clearstream API platform.

Bash / CURL

```
#!/usr/bin/env bash

# Script dependencies (tools you need to have installed to run this script
# 1) curl - recent version of curl (Old versions cause issues. E.g. 7.47.0 does not work)
# 2) jq - json parser for bash
echo " -----"
echo " -- Demo: Accessing the Clearstream API Playground with bash/curl --"
echo " -----"

# >>>> PRODUCTION <<<<
# ENVIRONMENT='https://api.clearstream.com'
# SSL_PEM='keystores/ocapi-a00012345_0012345-ssl.pem'
# CA_PEM='keystores/ocapi-a00012345_0012345-cacert.pem'
# USERNAME='a00012345_0012345'
# LOGIN_PASSWORD='*****'

# >>>> PRE-PRODUCTION (OCCT) <<<<
ENVIRONMENT='https://api-t2s-test.clearstream.com'
SSL_PEM='keystores/ocapi-stb2_01014172_1000000-ssl.pem'
CA_PEM='keystores/ocapi-stb2_01014172_1000000-cacert.pem'
USERNAME='stb2_01014172_1000000'
LOGIN_PASSWORD='*****'

# -----
echo "ENVIRONMENT=${ENVIRONMENT}"
echo "SSL_PEM=${SSL_PEM}"
echo "CA_PEM=${CA_PEM}"
echo "USERNAME=${USERNAME}"
echo "LOGIN_PASSWORD=${LOGIN_PASSWORD}"

echo $'\n>> 1) Request access token from authorization server'
ACCESS_TOKEN_RESP_CALL=$(curl \
  --cert ${SSL_PEM} \
  --cacert ${CA_PEM} \
  --data "grant_type=password" \
  --data "scope=allow ocapi-playground-v1" \
  --data "username=${USERNAME}" \
  --data "password=${LOGIN_PASSWORD}" \
  ${ENVIRONMENT}/authmanager/oauth2/access_token)

echo $'\n\n>> 2) Parse access token from response'
ACCESS_TOKEN=$(echo $ACCESS_TOKEN_RESP_CALL | jq -r .access_token)

# https://console.developer.deutsche-boerse.com/apis/9993a0e6-d53a-4bf9-94d4-73e4da49f6dd/technical-documentation
echo $'\n\n>> 3) Access Playground API with access_token'
curl \
  --cert ${SSL_PEM} \
  --cacert ${CA_PEM} \
  --header "Authorization: Bearer ${ACCESS_TOKEN}" \
  ${ENVIRONMENT}/playground/v1/echo?value=Hello%20World"

echo $'\n\n'
```