

---

# Clearstream API Developer Guide

**8 February 2022**

## Change log

3 June 2021	First draft
10 September 2021	SSL extraction details added
29 September 2021	Rework for Preview
11 November 2021	First version for publication
25 November 2021	Rebranding
8 February 2022	Simplification of DBP integration

## Introduction

This developer guide provides an overview on how to discover and browse 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 customer 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 “Getting Started” section provides some examples which show 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.

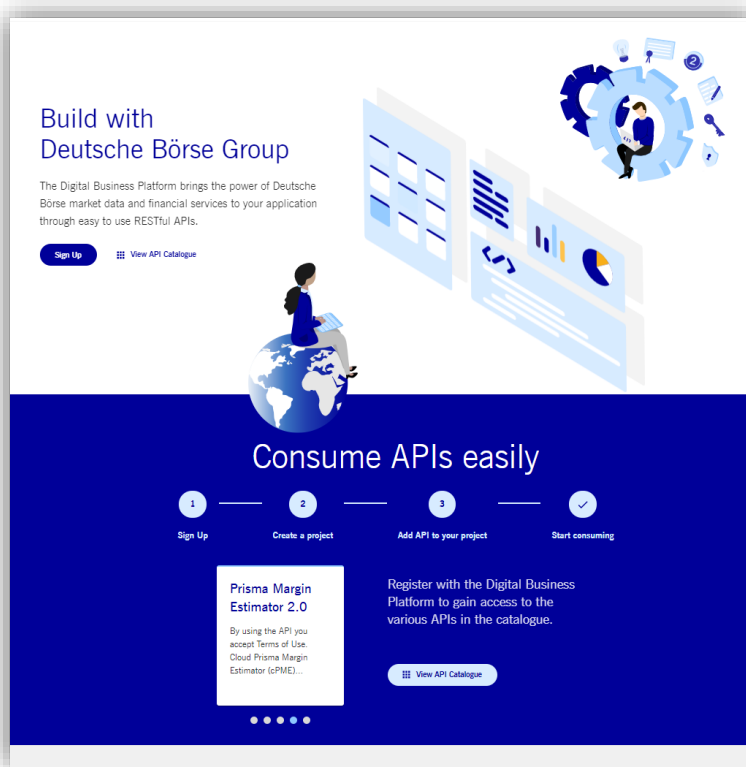
# Table of Contents

Introduction.....	1
Table of Contents .....	2
Pre-requisites .....	3
Digital Business Platform .....	3
Xact Web Portal.....	4
Getting Started.....	6
API Consumer - Permanent Credentials .....	6
Clearstream API Playground .....	6
Getting an OAuth2.0 Bearer Token (access_token) .....	7
Accessing the Playground API .....	8
Troubleshooting .....	9
I cannot get my OAuth2.0 client to work, what is missing? How does the support page work?.....	9
When trying to retrieve a bearer token (access_token), HTTP 4xx errors (client-errors) are returned.....	10
Using curl I am unable to retrieve an access_token and I get SSL/TLS errors. ....	11
Mutual TLS is not part of the OAuth2.0 base specification. As a result, it is not supported by my library.....	11
Example code.....	12
Bash.....	12
JavaScript (Browser).....	14

# Pre-requisites

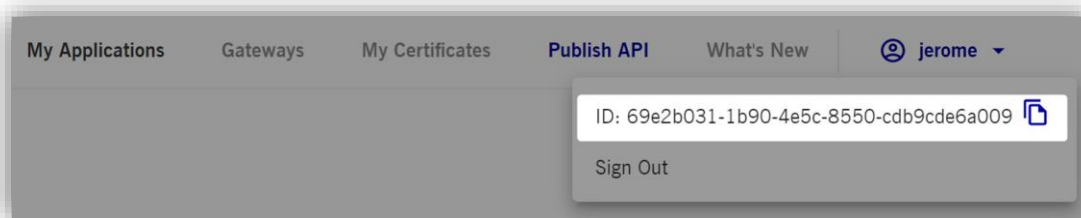
## Digital Business Platform

In order to browse through the Clearstream API catalogue developers must register on 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 specifications.<sup>1</sup>

To be able to link the an Xact API-Consumer account (see Getting started section), developers need to take note of the **DBP-User-ID**. This ID can be found in user dropdown in the upper right corner if logged into the DBP platform.



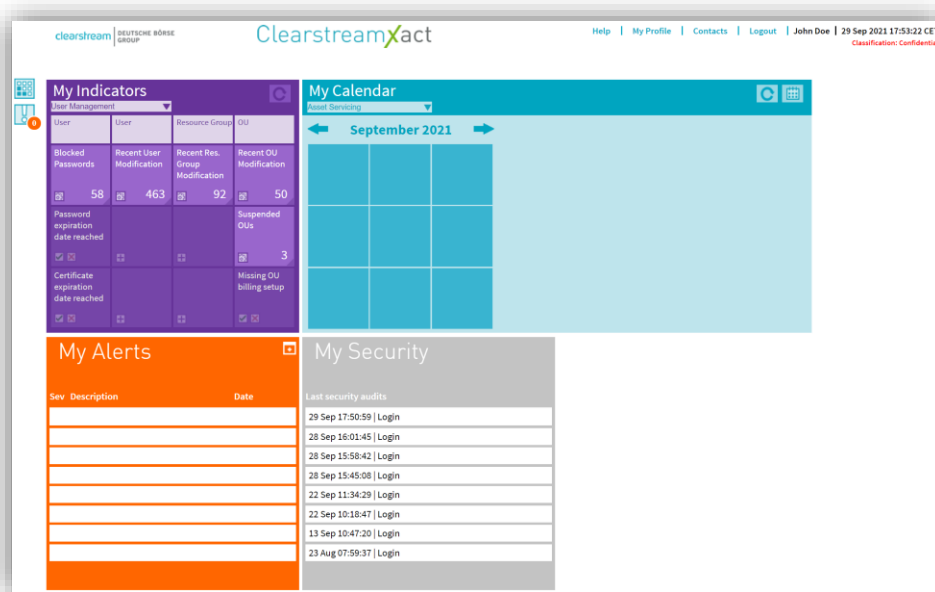
<sup>1</sup> Please note that for pre-production environments (such as the OCCT customer test environment), a staging deployment of the DBP is available here: <https://console.cstest.dbpapi.com>  
Clearstream API Developer Guide  
Clearstream Banking S.A.

## Xact Web Portal

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

### Xact Web Portal subscription

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.

To manage API access and generate API consumer credentials, existing Xact Web Portal customers must send an MT599 SWIFT message to Clearstream Connect to enable the Clearstream API service for their Organisation Unit (OU).

Future customers as well as other interested parties are advised to contact the Clearstream Connectivity Helpdesk directly for additional information. ([connect@clearstream.com](mailto:connect@clearstream.com))

## Xact API Consumer creation

“API Consumer” credentials can be created by following the standard Xact Web Portal user creation procedure<sup>2</sup>. The user type must be set to “API Consumer”. 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.

The screenshot shows the 'User' creation page in the Xact Web Portal. The page title is 'User'. The user type is set to 'API Consumer'. The 'Grant Selection' section shows the following table:

Clearstream APIs	Functional Roles	Resource Groups	Resources
<input checked="" type="checkbox"/> Clearstream API Playground (prod)	API Access	Clearstream API Playground (prod)	Clearstream API Playground (prod)

The 'User properties' table is as follows:

Service name	Property name	Category	Updatable by	Value
User Management	Smart Card	Credentials Types	All Types of Admin Users	Allowed
User Management	Software Key Store	Credentials Types	All Types of Admin Users	Allowed
User Management	User Mgt N-Eyes Principle	N-Eyes	All Types of Admin Users	2 eyes
User Management	User Time Zone	System	All Types of Admin Users	CET
User Management	Preferred Language	Session	All Types of Admin Users	English
Clearstream APIs	DBP User id	DBP	All Types of Admin Users	164c06c3-340e-4f56-80e4-0a7ccaace34

In the grant-selection section, an API consumer can be granted one or more Clearstream APIs. In addition to this general coarse-grained API access, it is important to grant specific fine-grained roles if needed by the API consumer later-on. For example, for most user management endpoints, the SCIM Admin or SCIM Read-Only roles are required. In contrary the Clearstream API playground endpoints do not have such a fine-grained access management.

Please note that during the creation of an API consumer, it is important to specify the **DBP-User-ID** in the API consumer’s user properties.

After the temporary credential creation of the API-Consumer, new temporary client certificates and private keys<sup>3</sup> will be downloaded automatically. These temporary credentials are required for an initial Xact Web Portal login as explained in the next section.

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

<sup>3</sup> Certificates and keys are stored inside a P12 key-store  
Clearstream API Developer Guide  
Clearstream Banking S.A.

# Getting Started

## API Consumer - Permanent Credentials

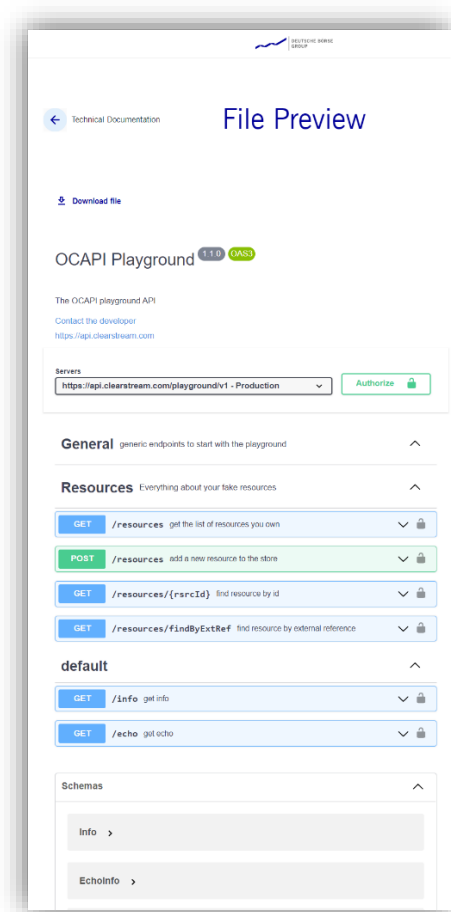
Once the temporary credentials of the “API Consumer” have been retrieved, an initial login with the API-Consumer to <https://xact.clearstream.com> is necessary to generate permanent credentials that remain valid for 2 years. This one-time login is part of the default Xact Web Portal security concept documented in the Xact Web Portal User Manual.<sup>4</sup>

Before first login to Xact	After first login Xact
<b>Temporary</b> SSL/SIGN certificates and private keys (inside P12)	<b>Permanent</b> SSL/SIGN certificates and private keys (inside P12)
<b>Temporary</b> login password	<b>Permanent</b> login password

Once permanent credentials have been generated, they can be used to consume granted APIs.

## Clearstream API Playground

It is recommended to start with the “Clearstream API Playground” that offers a variety of synthetic endpoints and is offered free of charge. The OpenAPI specification of this API is available on the DBP portal under the “Technical Documentation” tab.



<sup>4</sup> See: <https://www.clearstream.com/resource/blob/1311454/6a3813957a058c3138ba6fcdeb6dde71/xactusermanual-en-data.pdf>  
Clearstream API Developer Guide  
Clearstream Banking S.A.



## 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 flow<sup>5</sup> but has hardened it by enforcing a strict mutual TLS connection (Xact SSL certificate/key) in addition to a consumer/user password.<sup>6</sup>

More information on this can be found in the “Troubleshooting” section at the end of this document.

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, the section Example codes provides complete examples using Bash scripting and JavaScript.

```
curl
  --key ${SSL_KEY_FILE}
  --pass ${SSL_KEY_ENCRYPTION_PASSWORD}
  --cert ${SSL_CERT_FILE}
  --data 'grant_type=password'
  --data 'scope=allow ocapi-playground-v1'
  --data 'username=${API_CONSUMER_USERID}'
  --data 'password=${API_CONSUMER_PASSWORD}'
  https://api.clearstream.com/authmanager/oauth2/access_token
```

Running the above command yields a JSON response containing the access\_token, the granted scopes as well as a refresh token.

```
{
  "access_token": "9mkBME7gEGJ2Bf9C6E_VixrcZ4.uIBXbb1CY4NUxXg-TYmwBVF_Pys",
  "refresh_token": "9mkBME7gEGJ2Bf9C6E_VixrcZ4.ayun9r5RLBn0XqbTgC_baBV12po",
  "scope": "allow ocapi-playground-v1",
  "token_type": "Bearer",
  "expires_in": 3599
}
```

Those familiar with the OAuth 2.0 resource owner credentials flow may notice some differences in the token request compared to the default OAuth 2.0 approach.

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 normally required OAuth 2.0 basic-auth header to any random placeholder value is allowed and will simply be ignored by the Clearstream API infrastructure<sup>7</sup>.

<sup>5</sup> Sometimes referred to as the OAuth2.0 password flow

<sup>6</sup> 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.

<sup>7</sup> 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
--key ${SSL_KEY_FILE}
--pass ${SSL_KEY_ENCRYPTION_PASSWORD}
--cert ${SSL_CERT_FILE}
--header 'Authorization: Bearer ${ACCESS_TOKEN}'
https://api.clearstream.com/playground/v1/info
```

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

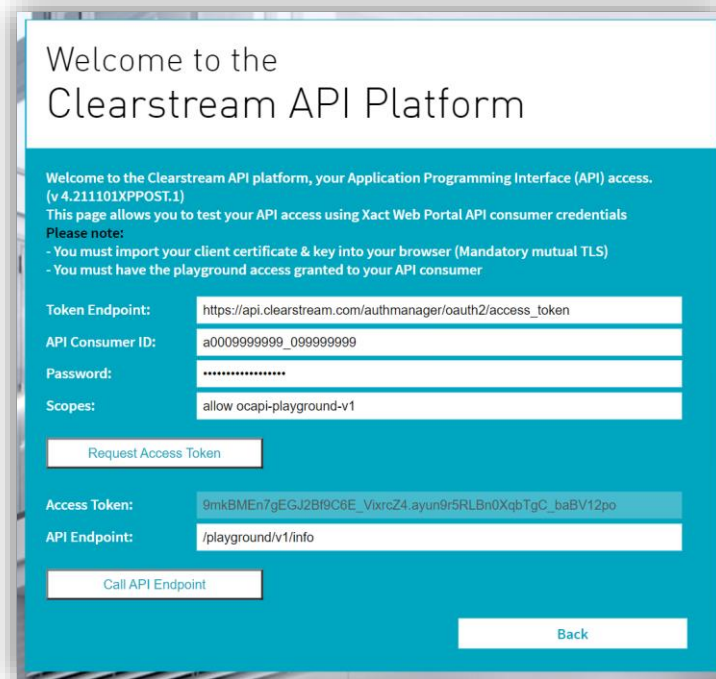
```
{
  "info_str": "Welcome to OCAPI Playground API ... ",
  "version": "1.211101XP.5"
}
```

## Troubleshooting

I cannot get my OAuth2.0 client to work, what is missing?

How does the support page work?

If there are problems with your OAuth2.0 client, please check the Clearstream API support page which provides a working web client. Analysing the network traffic between this client and the Clearstream API platform should help to better understand the setup. To use this client, the API consumer user must be granted access to the Clearstream Playground API as detailed in the “Pre-Requisite” chapter.



Welcome to the  
Clearstream API Platform

Welcome to the Clearstream API platform, your Application Programming Interface (API) access.  
(v 4.211101XPOST.1)  
This page allows you to test your API access using Xact Web Portal API consumer credentials

Please note:  
- You must import your client certificate & key into your browser (Mandatory mutual TLS)  
- You must have the playground access granted to your API consumer

Token Endpoint:

API Consumer ID:

Password:

Scopes:

Access Token:

API Endpoint:

The SSL certificate and key from the Xact credentials P12 must be imported into the browser. By default, the Xact Web Portal P12 cannot be imported as it is encrypted with algorithms that are not natively supported by Windows as well as browser key stores.

To generate a P12 key store that can be imported containing only the SSL certificate and key use the following bash and openssl commands:

```
P12='api-consumer.p12'
P12_PASSWORD='vSB6tONHxLRY7Q40'

openssl pkcs12 -in ${P12} -nocerts -nomacver -out tmp/xact-keys.pem -passin pass:${P12_PASSWORD}
-passout pass:${P12_PASSWORD}
csplit --silent -z -f tmp/xact-key- tmp/xact-keys.pem '/Bag Attributes/' '{*}'
openssl pkcs12 -in ${P12} -passin pass:${P12_PASSWORD} -out tmp/xact-certs.pem -nokeys
csplit --silent -z -f tmp/xact-cert- tmp/xact-certs.pem '/Bag Attributes/' '{*}'

SSL_KEY_FILE=""
SSL_CERT_FILE=""
if ! grep -q ssl "tmp/xact-key-00";
then
  SSL_KEY_FILE="tmp/xact-key-01"
  SSL_CERT_FILE="tmp/xact-cert-01"
else
  SSL_KEY_FILE="tmp/xact-key-00"
  SSL_CERT_FILE="tmp/xact-cert-00"
fi
echo ""
echo ""

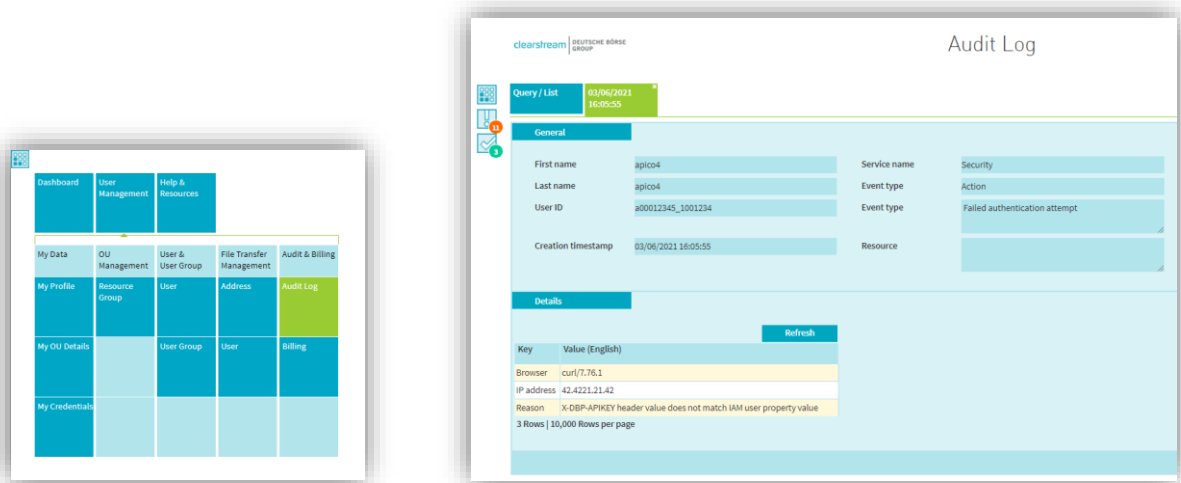
openssl pkcs12 -export -out tmp/consumer-ssl-only.p12 -in ${SSL_CERT_FILE} -inkey ${SSL_KEY_FILE}
-passin pass:${P12_PASSWORD} -passout pass:${P12_PASSWORD}
```

### When trying to retrieve 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.



### 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.

### Mutual TLS is not part of the OAuth2.0 base specification. As a result, it is not supported by my library.

The current OAuth2.0 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 putting a proxy in between your client and our infrastructure. This proxy should then wrap the client's one-way TLS into mutual TLS (two-way) before contacting the Clearstream API infrastructure.

## Example code

This section will provide some ready to use code examples for your convenience. Clearstream Banking does not take any responsibility in case of issues, problems or damages caused as a result of these example which are intended for demonstration only. 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

```
#!/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
# 3) openssl - cryptography swiss knife. Needed to deconstruct the Xact WebCrypto P12 for CURL PEM usage
# 4) csplit - splitting files by a separator

echo " -----"
echo " -- Demo: Accessing Clearstream Playground API with bash using Xact P12 --"
echo " -----"

ENVIRONMENT='https://api.clearstream.com'
P12='api-consumer.p12'
USERNAME='a000999999_00999999'
LOGIN_PASSWORD='xactPassword'
P12_PASSWORD='p12Password'

# -----

echo "ENVIRONMENT=${ENVIRONMENT}"
echo "P12=${P12}"
echo "LOGIN_PASSWORD=${LOGIN_PASSWORD}"
echo "P12_PASSWORD=${P12_PASSWORD}"
echo ""

# -----

echo ">> 1) Deleting previous temporary files"
rm -rf tmp
mkdir tmp
echo ""
echo ""

# -----

echo ">> 2) Creating PEM files from P12 for easier CURL handling"

openssl pkcs12 -in ${P12} -nocerts -nomacver -out tmp/xact-keys.pem -passin pass:${P12_PASSWORD} -passout
pass:${P12_PASSWORD}
csplit --silent -z -f tmp/xact-key- tmp/xact-keys.pem '/Bag Attributes/' '{*}'
openssl pkcs12 -in ${P12} -passin pass:${P12_PASSWORD} -out tmp/xact-certs.pem -nokeys
csplit --silent -z -f tmp/xact-cert- tmp/xact-certs.pem '/Bag Attributes/' '{*}'

SSL_KEY_FILE=""
SSL_CERT_FILE=""
if ! grep -q ssl "tmp/xact-key-00";
then
    SSL_KEY_FILE="tmp/xact-key-01"
    SSL_CERT_FILE="tmp/xact-cert-01"
else
    SSL_KEY_FILE="tmp/xact-key-00"
    SSL_CERT_FILE="tmp/xact-cert-00"
fi
echo ""
echo ""

# -----

# See next page

echo ">> 3) Request access token from authorization server"
```

```

ACCESS_TOKEN_RESP_CALL=$(curl \
  --insecure \
  --key ${SSL_KEY_FILE} \
  --pass ${P12_PASSWORD} \
  --cert ${SSL_CERT_FILE} \
  --data "grant_type=password" \
  --data "scope=allow ocapi-playground-v1" \
  --data "username=${USERNAME}" \
  --data "password=${LOGIN_PASSWORD}" \
  ${ENVIRONMENT}/authmanager/oauth2/access_token)
echo ""
echo ""

# -----

echo ">> 4) Parse access token from response"
ACCESS_TOKEN=$(echo $ACCESS_TOKEN_RESP_CALL | jq -r .access_token)
echo ">> $ACCESS_TOKEN"
echo ""
echo ""

# -----

echo ">> 5) Access API with access_token"
API_RESP_CALL=$(curl \
  --insecure \
  --key ${SSL_KEY_FILE} \
  --pass ${P12_PASSWORD} \
  --cert ${SSL_CERT_FILE} \
  --header "Authorization: Bearer $ACCESS_TOKEN" \
  ${ENVIRONMENT}/playground/v1/info)
echo ""
echo ""

# -----

echo ">> 6) API Response body"
echo $API_RESP_CALL | jq --color-output
echo ""
echo ""

```

## JavaScript (Browser)

For this demonstration to work, the API Consumer credential needs to be imported into your browser key store. (See Troubleshooting section) The browser will take care of the mutual TLS connection to the Clearstream API infrastructure.

This same code is used in the Clearstram API Support page (<https://api.clearstream.com/support.html>) to help test and troubleshoot connectivity issues.

```
<script>
  let baseDomain = https://api.clearstream.com

  // This support page allows troubleshooting token retrieval and API endpoint request
  // issues using a web frontend.
  // The Clearstream API platform supports the OAuth2 'password' grant type also known as
  // the "Resource Owner Credentials" flow
  function requestAccessToken(tokenEndpoint, userid, password, scopes) {
    fetch(tokenEndpoint,
      {
        method: 'POST',
        body: 'grant_type=password&scope=' + encodeURIComponent(scopes) + '&username=' + userid +
        '&password=' + password,
        headers: {
          'X-Requested-With': 'fetch',
          'Content-Type': 'application/x-www-form-urlencoded'
        }
      })
    .then(response => {
      if (response.ok) {
        response.json().then(function(data) {
          alert("Token Received")
          document.getElementById('access-token').value =
data.access_token;
        });
      } else {
        alert("Token Request Failed!");
      }
    })
    .catch(function(err) {
      console.error(err);
    });
  }

  // Endpoints protected by the Clearstream API platform require the OAuth access_token
  // to be provided as a Bearer token
  // Please consult the Clearstream API developer guide for a full overview including more
  // additional example code.
  function callApiEndpoint(apiEndpoint) {
    fetch(baseDomain + apiEndpoint,
      {
        redirect: 'follow',
        headers: {
          'Content-Type': 'application/json',
          'Authorization': 'Bearer ' + document.getElementById('access-
token').value
        }
      })
    .then(response => {
      if (response.ok) {
        response.text().then(function (text) {
          alert(text);
        });
      } else {
        alert("API Call Failed!");
      }
    })
    .catch(function(err) {
      console.error(err);
    });
  }
}
</script>
```