# User documentation for export client of the LSR API Search Service

# Document information

Project	Lehrstellenregister (LSR)
Responsible	Schweizerische Dienstleistungszentrum Berufsbildung SDBB
Version	0.1 (see change tracking)
Issue from	
State	Draft
Distribution list	

#### **Table of Contents**

- Document information
  - Table of Contents
  - Change tracking
- General information on the LSR API Search service
  - Prerequisite
  - Context
    - Apprenticeship places
    - Apprenticeships (LEFIs)
  - Data restrictions
  - Professions
  - Document scope
  - General overview of the Identity Server (IDService)
  - General overview of the LSR API Search Service
- Obtaining and using a token
  - Getting a user account
  - Procedure to obtain the token
  - Procedure to obtain the refresh token
- Description of the API
  - Description of API parameters
  - Description of API response
  - Tool Swagger
  - Tool Postman
- API planning

# Change tracking

Version	Date	From	What
0.1	28.10.2019	ALB	Initial version
0.2	25.11.2019	PG	Additional information
0.3	28.11.2019	ALB/PG	Final version

# General information on the LSR API Search service

# **Prerequisite**

Here is the correspondence between the English designations used in this document and the German designation:

German designation	English designation
Lehrort	Apprenticeship location
Betriebliche Bildung (Lehrort + Beruf)	Apprenticeship
Lehrstellenausschreibung	Apprenticeship place

Bewerbungskontakt	Application contact
Unternehmen	Enterprise
Lehrbeginn	School year

#### Context

The LSR project corresponds to the implementation of a new apprenticeship place's management system. This system replaces the existing one in which data was made available to third parties in the form of a data export via a SOAP service. This service is replaced by a new API that allows you to retrieve the data managed in LSR.

This documentation therefore concerns users who wish to retrieve data on an ad hoc basis to integrate it into their own management system, and not to retrieve and use it in real time. That's why clients are called "export users".

The API allows you to get 2 types of data:

- · apprenticeship places
- apprenticeships (LEFIs)

The API is called "Search service" and provides 2 end points for getting each type of data.

## Apprenticeship places

The apprenticeship places data provided by the API are the same as those provided by the current SOAP service, but also including the management of 2 school years. The LSR system makes it possible to manage the data of the apprenticeship places for 2 school years in order to facilitate the transition phase from one school year to the next. Each canton thus defines the period of publication of the apprenticeship places for its canton for each of the 2 school years. Depending on the time of year, places for a single year or for 2 school years can be published by the canton and therefore made available to third parties via the API. There is no common date for all cantons. The data available depends on the publication periods defined by each canton.

The information provided refers to:

- the apprenticeship location
- the application contact
- the profession
- the apprenticeship itself (language). This information is not specific for each school year.
- the apprenticeship places. This information is specific for each school year.

The data in the result is limited to apprenticeship places with at least one open place for one school year

The data in the result is sorted: Location code of the apprenticeship location, no SBFI and variant asc.

The end point to get apprenticeship places data is named: ExportApprenticeshipPlaces

# **Apprenticeships (LEFIs)**

The list of potential apprenticeships (LEFIs) is also available. It corresponds to the professions authorized to train for enterprises. This list of professions does not depend on whether or not the company currently offers open apprenticeship places.

Each county can choose if the LEFIs for its own county can be provided to third parties. The data available by the API depends on this cantonal choice.

The information provided refers to:

- · the apprenticeship location
- the application contact
- the profession
- the apprenticeship itself (language). This information is not specific for each school year.

The data in the result is sorted: Location code of the apprenticeship location, n° SBFI and variant asc.

The end point to get apprenticeship places data is named: ExportApprenticeships

# **Data restrictions**

To be able to use the API you must have a user who is authorized (see the paragraph "Getting a user account" below).

The right to obtain data is restricted only on an organizational basis. Each client of the API is responsible for retrieving only the data which he is allowed to according to the contract with the SDBB (based for example on Swisdoc number for selection professions).

For the moment, no restrictions on the number of requests to the API pro client are made. But the number of requests will nevertheless be monitored.

#### **Professions**

The data management in LSR is based on the SBFI professions system. Nevertheless, the correspondence with the SDBB Swissdoc system is carried out. You can therefore search for data according to these 2 systems.

In the case of Swissdoc, you can provide a partial number and thus retrieve all data relating to professions whose Swissdoc number begins with the character string.

# **Document scope**

This document is a documentation to be able to use this API, it is written in English because it is intended for a technical target audience.

This document must contain all the necessary elements for the technical implementation in the system of a canton by readers with the necessary knowledge of APIs.

This document is the first user documentation for using an API to be written by SDBB. If any elements are unclear, not detailed enough or missing from the document, any feedback is welcome.

# **General overview of the Identity Server (IDService)**

An identity server is a core part of any identity and access control infrastructure. It is the central database that stores user credentials. It also provides a means for establishing policies and permissions regarding who has access to which information.

Identity Server enables following features in your application:

#### Authentication as a Service :

Centralized login logic and workflow for all the applications.

Single Sign-on / Sign-out

Single sign-on (and out) over multiple application types.

## Access Control for APIs

Issue access tokens and refresh tokens for APIs for various types of clients, e.g. server to server, web applications.

#### OpenID Connect (OIDC)

The IDService provided by SDBB uses the OpenID Collect (OIDC) protocol.

OpenID Connect (OIDC) is an authentication layer, an authorization framework. The standard is controlled by the OpenID Foundation.

It uses simple JSON Web Tokens (JWT),

OIDC protocol exposes discovery documents.

The discovery document for SDBB IDService can be reached at URL https://id-test.sdbb.ch/auth/realms/SDBB\_Test/.well-known/openid-configuration which will give all information about the IDService on Test environment.

▲ URLs for the QUAL and PROD environment will be provided later

#### General overview of the LSR API Search Service

The described endpoint only allow LSR data to be read, no change is possible. The endpoint therefore only allow you to send requests of type GET to read the resources.

In return for a request, you obtain an http code and one or more apprenticeship objects in json.

The LSR API is versioned, the URL of the API includes the version number.

The LSR API is available on several environments:

- production
- quality
- test

# Obtaining and using a token

To be able to use the API, you need a token, derived from

- a username
- a password
- a client id
- and a client secret.

## Getting a user account

The user is created by the LSR-Administrator. A user is created for each client of the API and for each environment, in order to avoid errors on the environment used.

A naming convention has been defined for the username:

<environment TEST, QUAL or nothing for PROD>\_LSR\_<Export Client name>

In order to be able to create this user, an email address must be provided by each client. Thus the LSR-administrator can create the user. An email will be sent to the email address provided. This e-mail will contain a link which allows to define a password.

The LSR administrator also authorizes the user to use the API. The authorization to use the API relates to the two end points.

From the 4 information above, the user obtains an access\_token that is valid for the life time specified in authentication under expires in (25 min). After this period, the client must request a new access\_token using the refresh\_token provided.

The client\_id and client secret are common to all users for this API.

A For the test environment, at the beginning only one single dummy user is created for all clients (see section "Tool Swagger"). Each client must send to SDBB e-mail addresses in order to create users for TEST environment to Datenmanagement@sdbb.ch. The generic user will be deleted once specific users are created.

## Procedure to obtain the token

Here is the procedure to get the token: Make a POST request with:

- the following URL: https://id-test.sdbb.ch/auth/realms/SDBB\_Test/protocol/openid-connect/token
- the following parameters:
  - grant\_type
  - username
  - password
  - client id
  - · client secret

In return you get a json with the access token as shown below:

```
"access_token":
    "eyJhbGci0iJSUzIINiIsInR5cCIg0iAiSldUIiwia2lKIIAGICJuVXdqbEktOVRjZmIIOUetN3paZEtjMmmyTDVDTkhGRIJ3YTMmbGROaUVZIne.eyJqdGki0iJhY2zhZDRiMClmZGFlLTRkMjMtYjZiZS@yMGQwNTY@ZTlmODAiLCJleHAiOjEINzIJNj
    Q30DIsIm5iZiIGMCwiaWF0IjoxMTcyMjYzMjgyLCJpc3Mi0iJodHRwczovL2lkLXRlc3Quc2RiYi5jaC9hdXRoL3JIYWctcy9TREJCXIRlc3QiLCJhdMQi0iJMc3JTZXJ2aWNIIiwic3ViIjoiOGU5MGIIOGYtYTUXOS0@ZDczLTImYjUtNmIyYTdiZmM2N
    WMmIIwiddHumJoiQmVncmWyTiwiYXpwIjoiTHMQUZVydmljZSISImFldGhFdditZSI6MCwicZvzczLvb19zdGF0ZSI5IjTzMZyMMq2UTIIYZMHNOWAZSIJWAWLEZSIJWDaLTRhZjgzhzWkHzE2OCISImFjciIcJfziELCJyZWFsoV9hYZNLc3MiniyHtmIyYTdiZmM2N
    WMmIIwiddHumJoiQmvncmWyTiwiXXpwIjoiHmWyTWXSYZXJXWNIIIJfoiHMbyc2Vydm1jZSJ39.
    pnHbxVZ-rSIVD600g8mdkaQtW8DsNFGONIGVIJKg_gINbGlpKRefF8dSaJcXMNZeM905r5wryv_smci5tVx85wiP8N5F37PCyXuqL7bmpjKZtitAUsvIIF6h1H6jR_W-jzd11-g6wsQvsMBYhF7F9jkMCUK-f8PWEpZtRyA8GuY8FscYSRAaQIS7GGKzcb
    NDEtngA9e2PgXJTvstipRatlaARIrNAQErJZqj6z_MVLCVaJU008D0jVsscqof9McR0VFOXFMAQoMh2aM26xpFekWmb66Kna80_FkLV-fdkbkpM09KvSXLqTpVwP8ALMDKdN1BX65tM66WZ_inIPVjg",
    "expires_in": 1500,
    "refresh_expires_in": 1800,
    "refresh_cken":
    "eyJhbGci0JJUZIIMiIsInR5cCIg0iAiSldUIiwia2lKIIAGICJIZDkzZWU00Hi0JMTNjLTQZMWITYmY5MSIkNTgwYjg40DZmMmqifq.eyJqdGki0JJJVWQzMMY2YjgYjk0LTQ3ZDctO6FkMillNGMyMmVIMGZkZTQiLCJleHai0jElNzJMjUw0DIsImS
    iZiIGMcW.daWF0JjoxNTcyMjYzMjgyLCJpc3Mi0iJodHRwczovL2lkLXRlc3QucZRiYi5jaC9hdXRol3JlYMktcy9TREJCXIRlc3QiLCJdMQZlMYY2JYZYLLCJ0eXAi0JSZWZyZXMoIiwiXymIjoITMNyUZVydm1jZSIsImFlddhfdGltZSI6MCwicZwcz2lvb19zdGF0ZSI6IjIzMZYxMmQ3LT11VZMtNGM4ZSIiMDA2LTRhZjgzNzVMtcE2OCIsInJ1VxktX
    2FjYZYzcyIGeyJyDzxLcyGMyDNJZNGYZYYZYLCJ0eXai0JSZWZyZXMoIiwiXymIjoITMNyUZVydm1jZSIsImFldGhfdGltZSI6MCwicZwcz2lvb19zdGF0ZSI6IjIzMZYxMmQ3LT11VZMtNGM4ZSIMDA2LTRhZjgzNzVMtcE2OCIsInJ1VxktX
    Token_type: "bearer",
    "not-before-policy": 1564064994,
    "sesion Satate": "23347267-9ecc-4-8e-b006-4ef8375d3168",
```

Then the token is to be placed in the header of the http requests as follows (Example given for the token "xxx"):

"Authorization" field Value "Bearer xxx"

A detailed example to get the token will be covered below in section about using the Tool Postman

## Procedure to obtain the refresh token

Here is the procedure to get the token: Make a POST request with:

- the following URL: https://id-test.sdbb.ch/auth/realms/SDBB\_Test/protocol/openid-connect/token
- the following parameters:
  - grant\_type
  - refresh\_token
  - client id
  - client secret

In return you get a json with the access token token as shown below:

```
"access_token":
    "eyJhbGciOiJSUzIINiIsInR5cCIgOiAiSldUIwia2lkIiAGICJuVXdqbEktOVRjZmIIOU0tN3paZEtjMmwyTDVDTkhGRIJ3YTMwbGROaUVZIn0.eyJqdGkiOiIzYzViMTUyNC0yMzllLTRkMjUtYjcONiInNzcxZTc3YTJhOTUiLCJleHAiOjEINzJyNj
Q4HDUsImSizIiGMCvtaWF0IjOiNTcyMjYzhzQlLCJpc3MiOiJodHwczovL2lkLR1c3QucZRIYI5JaC9hdXRGL3JYNbctcy9TBEZOLR1c3QlLCJhdmqiOiJMc3TZXJzaWNIIwic3VIJjoiOSU5NGIIOGYtVTUxOS002DcclTlmYyUtMrdy
WhMIIAVdtAWIJOjuMchwmyTiaifYxdywJmjJSTisInFaChdCatScafeGwciz2vcz1vbGF0SZ516IjIzMZyvAmdQSIIIJNyZHNOMAZSIJMQAZITRATjgzWxkHz2COISInFijcifjEitCJZyJEFbSDMyPAJUZAJEFbSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEFBSDMyPAJUZAJEF
```

Then the token is to be placed in the header of the http requests as follows (Example given for the token "xxx"):

"Authorization" field Value "Bearer xxx"

A detailed example to get the refresh token will be covered below in section about using the Tool Postman

# Description of the API

In the detailed description of the API, you will find the following elements:

- Description of the end-point
- Description of the parameters (name, description, mandatory or not, type)
- · Description of the response

The description is available online. For the moment only on the test environment: https://lsr-search-test.sdbbservices.ch/index.html

⚠ Online description for SDBB IDService will be available soon

# **Description of API parameters**

For each parameter, the documentation contains:

- name.
- · description,
- mandatory or not,
- type

# **Description of API response**

In return for an API request, you get:

- an http code indicating whether the request was successful or not:
  - 200 (Ok: GET)
  - 400 (Bad Request)
- objects in json
  - see API resources for a description of the different fields
  - the character set of the response is UTF-8

The maximum number of results allowed in the response is XX. If the number of results is higher than this maximum, you will get an error XX. You should use pagging to retrieve all the results by making several requests, indicating in each one, the number of results in the answer and the number of the page.

# **Tool Swagger**

The documentation is produced with the tool Swagger.

In Swagger, you must be authorized if you want to have access to the description of end points. User / Password must be provided to the clients.

username :<environment TEST, QUAL or nothing for PROD>\_LSR\_<Export Client name>

password: \*\*\*\*\*\*

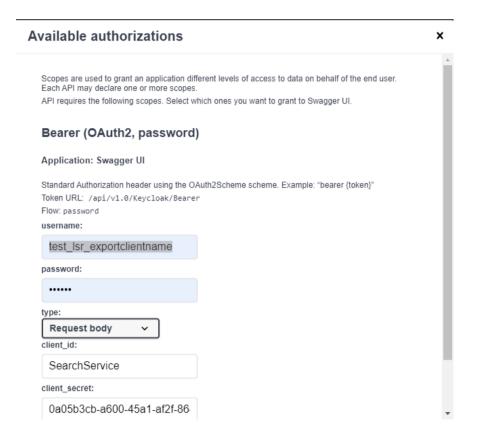
.

Temporary dummy User Name and Password for Test environment:

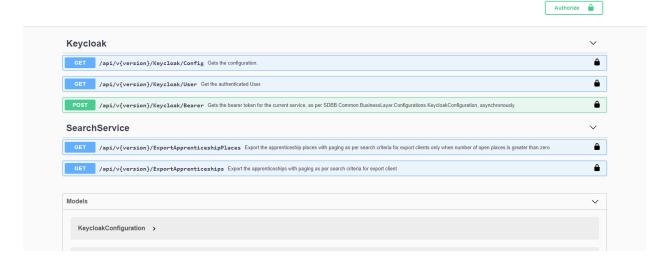
username : test\_lsr\_exportclientname

password: 123456

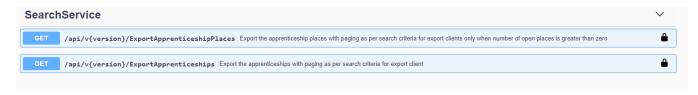
Click on Authorize



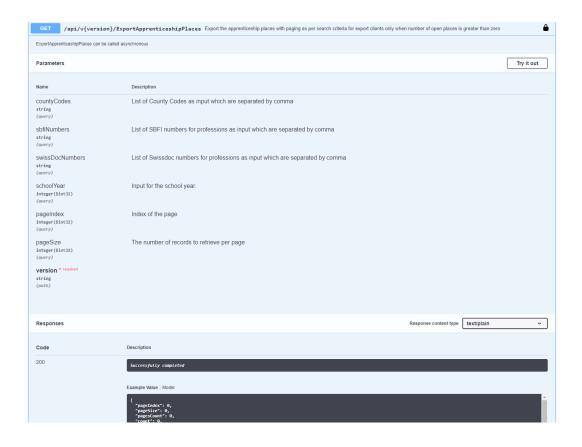
Once it is successfully authorized, Swagger's GUI also allows you to run queries directly from the GUI (action "Try it out") and view the response as shown below

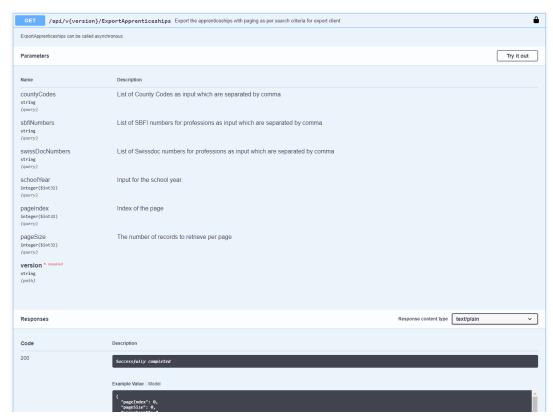


Once authorized, swagger will display the two endpoints under SearchService for Export Clients as shown below:



The Model and its description for each endpoint is available as shown below:





⚠ for the moment, do not download the file swagger.json from the GUI because it is incomplete, but use the one provided by the SDBB (sent via Email).

## **Tool Postman**

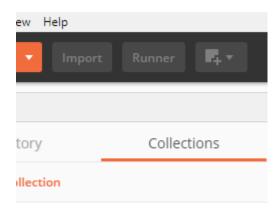
Postman is a free tool which offers graphical interface to send API requests.

Postman Collections are simply a collection of pre-built requests that can be organized into folders, and they can be easily exported and shared with others.

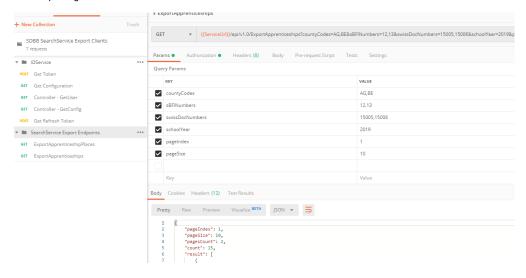
There are two json files named SDBB SearchServiceExportClients.postman\_collection.json and SDBB SDBB SearchServiceExportClients. postman\_environment.json which include the environment variables and pre-build in request for SearchService to get mock data of apprenticeships and apprenticeship places for export clients.

Example for importing SearchService Collections and environment as shown below

- 1. Download and install Postman from the link https://www.getpostman.com/downloads/
- 2. After that open Postman App
- 3. Follow the steps below to open the above two json files SDBB SearchServiceExportClients.postman\_collection.json and SDBB SDBB SearchServiceExportClients.postman\_environment.json provided by SDBB
- 4. Click on Import and choose the two json files



5. After importing there will be a folder created in Collections as below



Environment selected should be as shown below



- 6. SDBB SearchService Export Clients contains pre-defined requests for endpoints related to IDService and SearchService Export endpoints to get mock data.
  - 7. SDBB SearchService ExportClients environment contains the various variables as below
    - 1. ServiceUrl URL of the service whose endpoints need to be called.

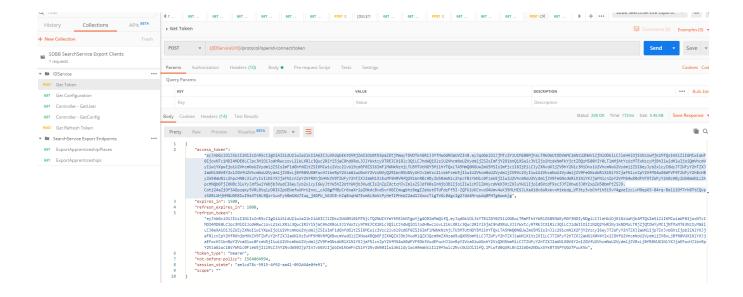
- 2. IDServiceUrl URL of the ID Service provider for authentication and authorization
- 3. ClientId Client Id to get the token
- 4. ClientSecret Client secret to get the token
- 5. UserName user name
- 6. Password password
- 7. crtToken The bearer token which will be used for API request for authorization
- 8. RefreshToken The refresh token which can be used to get the access token again

SDBB SearchServio	e ExportClients	Edit
VARIABLE	INITIAL VALUE	CURRENT VALUE
ServiceUrl	https://lsr-search-test.sdbbservices.ch/	https://lsr-search-test.sdbbservices.ch/
IDServiceUrl	https://id-test.sdbb.ch/auth/realms/SDBB_Test	https://id- test.sdbb.ch/auth/realms/SDBB_Test
ClientId	SearchService	SearchService
ClientSecret	0a05b3cb-a600-45a1-af2f-8640f99ccb8b	0a05b3cb-a600-45a1-af2f-8640f99ccb8b
crtToken	eyJhbGciOiJSUzl1NilsInR5cClgOiAiSldUliwia2lkl iA6lCJuVXdqbEktOVRjZml1OU0tN3paZEtjMmw yTDVDTkhGR1J3YTMwbGROaUVZIn0.eyJqdGki OiJhYzQwYzk4OS1kN	eyJhbGciOiJSUzl1NilsInR5cClgOiAiSldUliwia 2lkliA6lCJuVXdqbEktOVRjZml1OU0tN3paZE tjMmwyTDVDTkhGR1J3YTMwbGROaUVZIn 0.eyJqdGkiOiJjMTc2YzU1MS00M
UserName	test_lsr_exportclientname	test_lsr_exportclientname
Password	123456	123456
RefreshToken	eyJhbGciOiJSUzl1NilsInR5cClgOiAiSldUliwia2lkl iA6lCJuVXdqbEktOVRjZml1OU0tN3paZEtjMmw yTDVDTkhGR1J3YTMwbGROaUVZIn0.eyJqdGki Oil4ZWRjYjJINS04Y	eyJhbGciOiJSUzl1NilsInR5cClgOiAiSldUliwia 2lkliA6lCJuVXdqbEktOVRjZml1OU0tN3paZE tjMmwyTDVDTkhGR1J3YTMwbGROaUVZIn 0.eyJqdGkiOil4ZWRjYjJINS04Y
Globals		Edit

which will be used by IDService and SearchService.

## How to obtain token

The pic below shows the keys and its value that need to be sent in body of POST request to get the bearer token for authentication



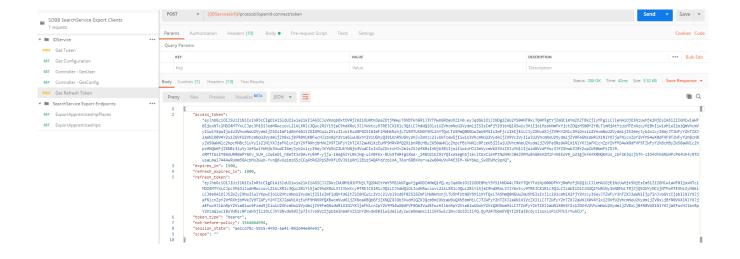
Copy the access\_token from above and paste it in crtToken variable in SearchService Export Client Environment as shown below

SDBB SearchService E	xportClients	Ed
VARIABLE	INITIAL VALUE	CURRENT VALUE
ServiceUrl	https://lsr-search-test.sdbbservices.ch/	https://lsr-search-test.sdbbservices.ch/
IDServiceUrl	https://id-test.sdbb.ch/auth/realms/SDBB_Test	https://id- test.sdbb.ch/auth/realms/SDBB_Test
Clientld	SearchService	SearchService
ClientSecret	0a05b3cb-a600-45a1-af2f-8640f99ccb8b	0a05b3cb-a600-45a1-af2f-8640f99ccb8b
crtToken	eyJhbGciOiJSUzI1NilsInR5cClgOiAiSIdUliwia2lkl iA6lCJuVXdqbEktOVRjZml1OU0tN3paZEtjMmw yTDVDTkhGR1J3YTMwbGROaUVZIn0.eyJqdGki OiJhYzQwYzk4OS1kN	eyJhbGciOiJSUzl1NilsInR5cClgOiAiSldUliwia 2lkliA6lCJuVXdqbEktOVRjZml1OU0tN3paZE tjMmwyTDVDTkhGR1J3YTMwbGROaUVZIn 0.eyJqdGkiOiJjMTc2YzU1MS00M
UserName	test_lsr_exportclientname	test_lsr_exportclientname
Password	123456	123456
RefreshToken	eyJhbGciOiJSUzI1NilsInR5cCIgOiAiSIdUliwia2lkl iA6ICJuVXdqbEktOVRjZml1OU0tN3paZEtjMmw yTDVDTkhGR1J3YTMwbGROaUVZIn0.eyJqdGki OiI4ZWRjYjJINS04Y	eyJhbGciOiJSUzl1NilsInR5cClgOiAiSldUliwia 2lkliA6lCJuVXdqbEktOVRjZml1OU0tN3paZE tjMmwyTDVDTkhGR1J3YTMwbGROaUVZIn 0.eyJqdGkiOil4ZWRjYjJINS04Y

# **Example to call Refresh Token**

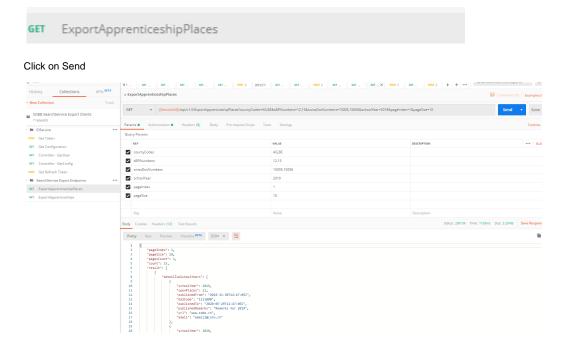
Use the refresh\_token in the above pics to get an access token without login page

Get Refresh Token



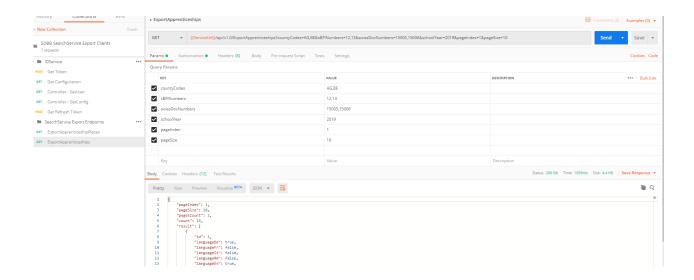
#### Example to call API endpoint for mock data

1. Export Apprenticeships Places :



## 2. Export Apprenticeships:

GET ExportApprenticeships



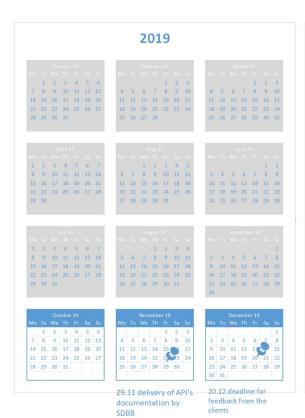
# API planning

### Steps to consider:

- Documentation
  - Delivery of documentation to the Export Clients by SDBB
  - · Provision of feedback by the Export Clients
  - · Provision of a final version of the documentation by SDBB
- MOCK
  - Provision of a MOCK to the clients in order to carry out tests and obtain dummy data in the response. 

     \( \begin{align\*} \) in the context of the mock, the parameters therefore have no effect on the data provided in the response, the data in the response are the same for all users.
  - The two endpoints which return mock data are included in the collection and environment json files for searchservice with this package.
  - Provision of user accounts for TEST
- It's only dummy data. The API response will always be the same and will not take into account the parameters of the request.
- Functional version of the API
  - A functional version of the API and real data will be made available on the QUALITY environment.
  - The data will come from data files delivered by the cantons. The data will not necessarily be updated daily.
  - This version will have been tested internally by SDBB.
- User tests
  - The cantons must carry out tests on the QUALITY environment
  - Support from SDBB is planned
  - Bugs found will be fixed and re-delivered on QUALITY
  - Change requests will be analyzed and planned
- Go Live
  - Provision of user accounts for PROD
  - The Go Live will be concurrent with the opening of LSR to counties

The main milestones are represented below:



			nuary								v 20									
						Su	Mo						Su	Mo						
		1	2	3	4	5						1	2					П		1
6	7	8	9	10	0	12	3	4	5	6	7	8	9	2	3	4	5	6	7	8
13	14	15	16	17	18	19	10	11	12	13	14	15	16	9	10	11	12	13	14	15
20	21	22	23	24	25	26	17	18	19	20	21	22	23	16	17	18	19	20	21	22
27	28	29	30	31			24	25	26	27	28	29		23	24	25	26	27	28	29
														30	31					
			pril	20.						1ay 2	10					9	unei	20		
Mo			prii ، Th			Su	Mo						Su	Mo						
	1.0	1	2	30	Ö	5	WIO	Ö	ive	-""	1	2	3	1	2	3	4	5	6	7
6	7	8	9	16	11	12	12	5	6	7	8	9	10	8	9	10	11	12	13	14
13	14	15	16	17	18	19	11	12	13	14	15	16	17	15	16	17	18	19	20	2:
20	21	22	23	24	25	26	18	19	20	21	22	23	24	22	23	24	25	26	27	28
27	28	29	30	2-4	20	20	25	26	27	28	29	30	31	29	30			20		-
	2.0	2.0	00				2.5	2.0	2.1	2.0		00	01	-	00					
Мо	Tu	We	Th	Fr	Sa	Su	Мо	Tu	We	Th	Fr	Sa	Su	Мо	Tu	We	Th	Fr	5a	Sι
		1	2	3	4	5						1	2		1	2	3	4	5	6
6	7	8	9	10	11	12	3	4	5	6	7	8	9	7	8	9	10	11	12	13
13	14	15	16	17	18	19	10	11	12	13	14	15	16	14	15	16	17	18	19	20
20	21	22	23	24	25	26	17	18	19	20	21	22	23	21	22	23	24	25	26	27
27	28	29	30	31		_	24	25	26	27	28	29	30	28	29	30				
							31													
		0	tobe	20						- 1	20	3				8	- 14	20		
						Su	Mo			em be Th			Su	Mo		We				
		-	1	2	3	4							1	1110	1	2	3	4	5	6
5	6	7	8	9	10	11	2	3	4	5	6	7	8	7	8	9	10	11	12	13
12	13	14	15	16	17	18	9	10	11	12	13	14	15	14	15	16	17	18	19	20
19	20	21	22	23	24	25	16	17	18	19	20	21	22	21	22	23	24	25	26	27
26	27	28	29	30	31	20	23	24	25	26	27	28	29	28	29	30	31	23	20	21
20	21	20	29	30	31		30	24	23	20	21	20	23	20	29	30	31			

April 2020: operational API on

Mai 2020: Acceptance testing on QUAL June 2020 Go Live API

QUAL

2020

17.01 final version of

the documentation and