User documentation of the LSR API

Document information

Project	Lehrstellenregister (LSR)
Responsible	Schweizerische Dienstleistungszentrum Berufsbildung SDBB
Version	1.0 (see change tracking)
Issue from	29.10.2019 (see change tracking)
Status	Pending Approval
Distribution list	

Table of Contents

- Document information
  - Table of Contents
  - Change tracking
- General information on the LSR API
  - Prerequisite
  - Context
  - Rücklieferung vs Volllieferung
  - Business rules
  - Document scope
  - General overview of the Identity Server (IDService)
  - General overview of the LSR API
- 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	26.09.2019	ALB	Initial version
0.2	02.10.19	ALB	Adaptations after meetings with DEV
1.0	29.10.19	ALB/PG	Initial public version

General information on the LSR API

Prerequisite

Here is the correspondence between the English designations used in this document and the German designation used for example in Dat-Rili:

German designation	English designation
Lehrort	Apprenticeship location
Betriebliche Bildung	Apprenticeship
Lehrstellenausschreibung	Apprenticeship place
Bewerbungskontakt	Application contact
Unternehmen	Enterprise

Lehrbeginn	School year

Context

With the implementation of the LSR project, some large companies will be able to manage their apprenticeship places themselves. This management will be based on the authorized professions managed by the cantons.

NB: It is therefore still necessary for the cantons to send all apprenticeships in their canton to LSR.

Large companies will be able to manage themselves:

- the number of open places,
- information for the application,
- comments
- and language of learning

without notifying the cantons of the modifications, but directly in LSR.
LSR thus becomes the only application containing the **up-to-date data of large companies**.
Technically an API is made available to the cantons so that they can retrieve this data managed by large companies to update their own apprenticeship management system.

Rücklieferung vs Volllieferung

An end point is therefore made available to the cantons to specifically get the **data of these large companies**. The provision of data to the cantons has often been called "Rücklieferung". But **no regular and planned file exchange is planned**, cantons can request and get data whenever they wish using the LSR API.

A second end point is made available to the cantons which makes it possible to retrieve data from apprenticeship places **for all enterprises** and also **other cantons than the user's one**. In a future version of LSR, the cantons will be able to announce via an API some updates to apprenticeship places at any time, without having to wait for the next export of their data and its import into LSR. The cantons will be able to use this API in addition to the data file transfer. The file must continue to contain all the data of the canton. In order to be able to check the consistency between the cantonal system and the data present in LSR, this second end point allows access to all the data. The cantons will therefore be able to ensure that, regardless of the mode of transmission, the data in their system and in LSR are consistent. This end point was called "Volllieferung".

The end points provided by LSR are:

- **CentrallyManagedApprenticeships** in order to get the data of large companies (Rücklieferung)
- **Apprenticeships** in order to get all the apprenticeships (Volllieferung)

Modifications relating to apprenticeship locations and authorized professions remain in the responsibility of the cantons, which must provide them to LSR in data files in accordance with Dat-Rili (http://www.sbbk.ch/dyn/23086.php).

The cantons providing their data according to the format of these guidelines, the LSR API provides the data to the cantons in a very similar structure.

The structure of the data provided after a request is **identical for the 2 end points**. This structure is extremely close to the structure defined in Dat-Rili 3.07 for transactions 3010.

Business rules

The business rules applied when providing data to the cantons are described in the document *LSR_ Business Rules Import KDL-fr* or *LSR_ Business Rules Import KDL-de* sent to the counties (published on http://www.sdbb.ch/dyn/211420.asp or http://www.sdbb.ch/dyn/211555.asp).

Only data that complies with these rules is available via the API.

In particular the end point CentrallyManagedApprenticeships provides **valid data and deleted data**. The deleted data correspond to data provided by the cantons at a given time and later deleted by the canton (either because the data is no longer present in the data file, or later by announcing a deletion). When the canton deletes a data, it is not physically deleted in LSR because this data can be shared with large companies. The data is invalidated in LSR but can be made valid again by the canton, the modifications made by the large company are then not lost.

The end point Apprenticeships **only provides valid data** (so in the results the flag Deleted is always False).

The end point CentrallyManagedApprenticeships allows either to recover all data or only those modified since a specific date.

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

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

The provided end points allow to get the data of:

- the enterprise
- the apprenticeship location
- the application contact
- the online application
- the apprenticeship places

in the form of data whose level of detail is apprenticeship (as in the files provided by the cantons). 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

⚠️As the API is not highly available, so you must manage the case where it is temporarily unavailable.

Obtaining and using a token

To be able to use the LSR 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 canton 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_<2-character canton code>

In order to be able to create this user, an email address must be provided by each canton. 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 Rücklieferung vs Volllieferung.

From the 4 information above, the user obtains an access_token that is valid for the life time specified in authentication under exprires_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.

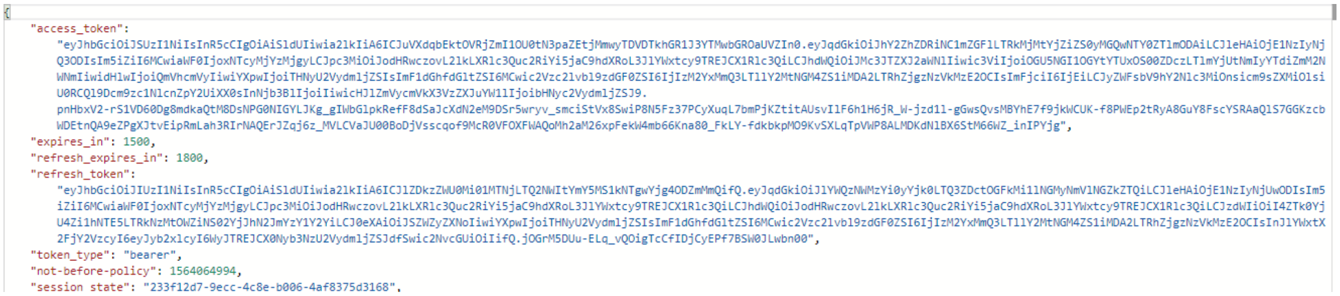
⚠ **For the test environment, at the beginning only one single generic user is created for all cantons (see section "Tool Swagger"). Each county must send to SDBB e-mail addresses in order to create users for TEST environment to Datenmanagement@sdbb.ch. The generic user will be deleted once cantonal 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":
        "eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJuVXdqbEktOVRjZmI1OU0tN3paZEtjMmwyTDVDTkhGR1J3YTMwbGROaUVZIn0.eyJqdGkiOiJhY2ZhZDRiNC1mZGF1LTRkMjMtYjZiZS0yMGQwNTY0ZTlmODAiLCJleHAiOjE1NzIyNj
        Q3ODIsIm5iZiI6MCwiaWF0IjoxNTcyMjYzMjgyLCJpc3MiOiJodHRwczovL2lkLXRlc3Quc2RiYi5jaC9hdXRoL3JlYWxtcy9TREJCX1Rlc3QiLCJhdWQiOiJMc3JTZXJ2aWNlIiwic3ViIjoiOGU5NGI1OGYtYTUxOS00ZDczLTlmYjUtNmIyYTdiZmM2N
        WNmIiwidHlwIjoiQmVhcmVyIiwiYXpwIjoiTHNyU2VydmljZSIsImF1dGhfdGltZSI6MCwic2Vzc2lvbl9zdGF0ZSI6IjIzM2YxMmQ3LTllY2MtNGM4ZS1iMDA2LTRhZjgzNzVkMzE2OCIsImFjciI6IjEiLCJyZWFsbV9hY2Nlc3MiOnsicm9sZXMiOlsi
        U0RCQl9Dcm9zc1NlcnZpY2UiXX0sInNjb3BlIjoiIiwicHJlZmVycmVkX3VzZXJuYW1lIjoibHNyc2VydmljZSJ9.
        pnHbxV2-rS1VD60Dg8mdkaQtM8DsNPG0NIGYLJKg_gIWbGlpkRefF8dSaJcXdN2eM9DSr5wryv_smciStVx8SwiP8N5Fz37PCyXuqL7bmPjKZtitAUsvIlF6h1H6jR_W-jzd1l-gGwsQvsMBYhE7f9jkWCUK-f8PWEp2tRyA8GuY8FscYSRAaQlS7GGKzcb
        WDEtnQA9eZPgXJtvEipRmLah3RIrNAQErJZqj6z_MVLCVaJU00BoDjVsscqof9McR0VFOXFWAQoMh2aM26xpFekW4mb66Kna80_FkLY-fdkbkpMO9KvSXLqTpVWP8ALMDKdNlBX6StM66WZ_inIPYjg",
    "expires_in": 1500,
    "refresh_expires_in": 1800,
    "refresh_token":
        "eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJlZDkzZWU0Mi01MTNjLTQ2NWItYmY5MS1kNTgwYjg4ODZmMmQifQ.eyJqdGkiOiJlYWQzNWMzYi0yYjk0LTQ3ZDctOGFkMi1lNGMyNmVlNGZkZTQiLCJleHAiOjE1NzIyNjUwODIsIm5
        iZiI6MCwiaWF0IjoxNTcyMjYzMjgyLCJpc3MiOiJodHRwczovL2lkLXRlc3Quc2RiYi5jaC9hdXRoL3JlYWxtcy9TREJCX1Rlc3QiLCJhdWQiOiJodHRwczovL2lkLXRlc3Quc2RiYi5jaC9hdXRoL3JlYWxtcy9TREJCX1Rlc3QiLCJzdWIiOiI4ZTk0Yj
        U4Zi1hNTE5LTRkNzMtOWZiNS02YjJhN2JmYzY1Y2YiLCJ0eXAiOiJSZWZyZXNoIiwiYXpwIjoiTHNyU2VydmljZSIsImF1dGhfdGltZSI6MCwic2Vzc2lvbl9zdGF0ZSI6IjIzM2YxMmQ3LTllY2MtNGM4ZS1iMDA2LTRhZjgzNzVkMzE2OCIsInJlYWxtX
        2FjY2VzcyI6eyJyb2xlcyI6WyJTREJCX0Nyb3NzU2VydmljZSJdfSwic2NvcGUiOiIifQ.jOGrM5DUu-ELq_vQOigTcCfIDjCyEPf7BSW0JLwbn00",
    "token_type": "bearer",
    "not-before-policy": 1564064994,
    "session state": "233f12d7-9ecc-4c8e-b006-4af8375d3168",
```

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"

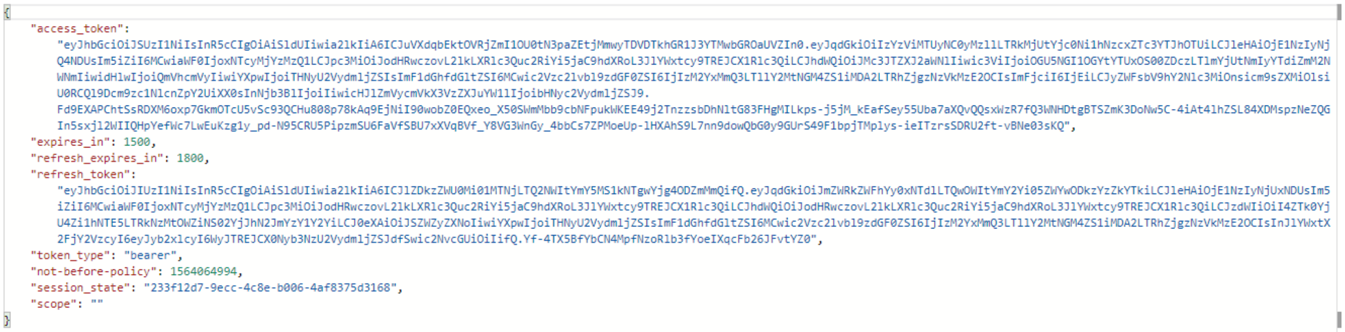
A detailed example to get the token will be covered below in section about using the Tool Postman

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":
        "eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJuVXdqbEktOVRjZmI1OU0tN3paZEtjMmwyTDVDTkhGR1J3YTMwbGROaUVZIn0.eyJqdGkiOiIzYzViMTUyNC0yMzllLTRkMjUtYjc0Ni1hNzcxZTc3YTJhOTUiLCJleHAiOjE1NzIyNj
        Q4NDUsIm5iZiI6MCwiaWF0IjoxNTcyMjYzMzQ1LCJpc3MiOiJodHRwczovL2lkLXRlc3Quc2RiYi5jaC9hdXRoL3JlYWxtcy9TREJCX1Rlc3QiLCJhdWQiOiJMc3JTZXJ2aWNlIiwic3ViIjoiOGU5NGI1OGYtYTUxOS00ZDczLTlmYjUtNmIyYTdiZmM2N
        WNmIiwidHlwIjoiQmVhcmVyIiwiYXpwIjoiTHNyU2VydmljZSIsImF1dGhfdGltZSI6MCwic2Vzc2lvbl9zdGF0ZSI6IjIzM2YxMmQ3LTllY2MtNGM4ZS1iMDA2LTRhZjgzNzVkMzE2OCIsImFjciI6IjEiLCJyZWFsbV9hY2Nlc3MiOnsicm9sZXMiOlsi
        U0RCQl9Dcm9zc1NlcnZpY2UiXX0sInNjb3BlIjoiIiwicHJlZmVycmVkX3VzZXJuYW1lIjoibHNyc2VydmljZSJ9.
        Fd9EXAPChtSsRDXM6oxp7GkmOTcU5vSc93QCHu808p78kAq9EjNiI90wobZ0EQxeo_X50SWmMbb9cbNFpukWKEE49j2TnzzsbDhNltG83FHgMILkps-j5jM_kEafSey55Uba7aXQvQQsxWzR7fQ3WNHDtgBTSZmK3DoNw5C-4iAt4lhZSL84XDMspzNeZQG
        In5sxjl2WIIQHpYefWc7LwEuKzg1y_pd-N95CRU5PipzmSU6FaVfSBU7xXVqBVf_Y8VG3WnGy_4bbCs7ZPMoeUp-lHXAhS9L7nn9dowQbG0y9GUrS49F1bpjTMplys-ieITzrsSDRU2ft-vBNe03sKQ",
    "expires_in": 1500,
    "refresh_expires_in": 1800,
    "refresh_token":
        "eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJlZDkzZWU0Mi01MTNjLTQ2NWItYmY5MS1kNTgwYjg4ODZmMmQifQ.eyJqdGkiOiJmZWRkZWFhYy0xNTdlLTQwOWItYmY2Yi05ZWYwODkzYzZkYTkiLCJleHAiOjE1NzIyNjUxNDUsIm5
        iZiI6MCwiaWF0IjoxNTcyMjYzMzQ1LCJpc3MiOiJodHRwczovL2lkLXRlc3Quc2RiYi5jaC9hdXRoL3JlYWxtcy9TREJCX1Rlc3QiLCJhdWQiOiJodHRwczovL2lkLXRlc3Quc2RiYi5jaC9hdXRoL3JlYWxtcy9TREJCX1Rlc3QiLCJzdWIiOiI4ZTk0Yj
        U4Zi1hNTE5LTRkNzMtOWZiNS02YjJhN2JmYzY1Y2YiLCJ0eXAiOiJSZWZyZXNoIiwiYXpwIjoiTHNyU2VydmljZSIsImF1dGhfdGltZSI6MCwic2Vzc2lvbl9zdGF0ZSI6IjIzM2YxMmQ3LTllY2MtNGM4ZS1iMDA2LTRhZjgzNzVkMzE2OCIsInJlYWxtX
        2FjY2VzcyI6eyJyb2xlcyI6WyJTREJCX0Nyb3NzU2VydmljZSJdfSwic2NvcGUiOiIifQ.Yf-4TX5BfYbCN4MpfNzoRlb3fYoeIXqcFb26JFvtYZ0",
    "token_type": "bearer",
    "not-before-policy": 1564064994,
    "session_state": "233f12d7-9ecc-4c8e-b006-4af8375d3168",
    "scope": ""
}
```

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-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

In the description of each response field, the link to the corresponding field in Dat-Rili 3.07 is given.

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 cantons.

username : <environment TEST, QUAL or nothing for PROD>_LSR_{CantonCode}

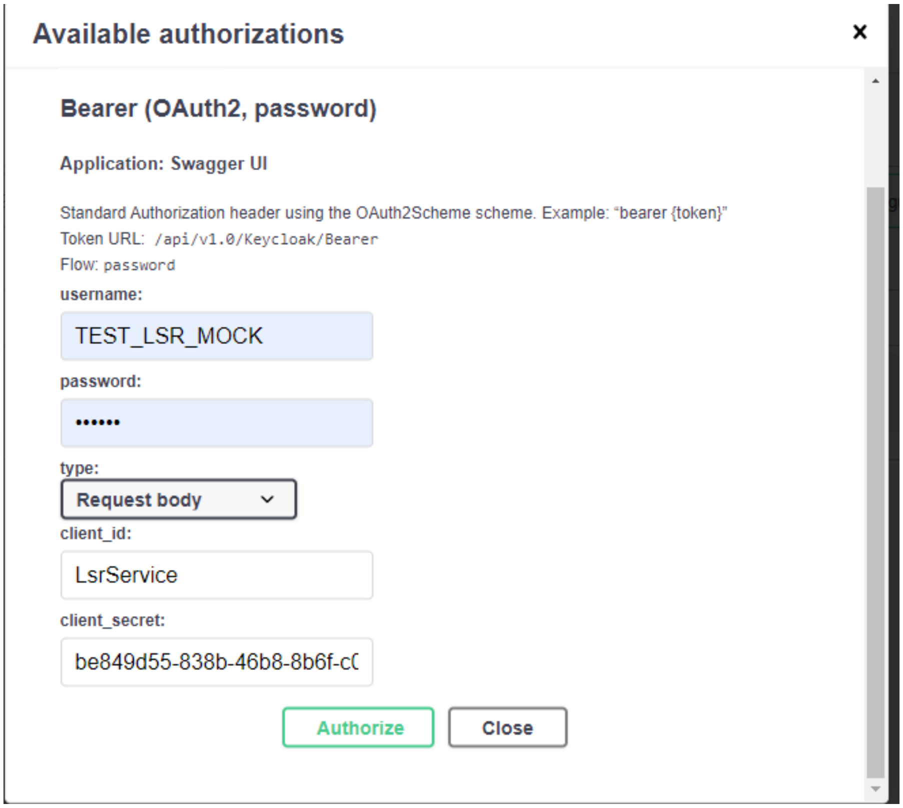
password: *******

⚠️Temporary User Name and Password for mock data:

username : TEST_LSR_MOCK

password : 123456

Click on Authorize



Available authorizations

Bearer (OAuth2, password)

Application: Swagger UI

Standard Authorization header using the OAuth2Scheme scheme. Example: "bearer {token}"
Token URL: `/api/v1.0/Keycloak/Bearer`
Flow: `password`

username: TEST_LSR_MOCK

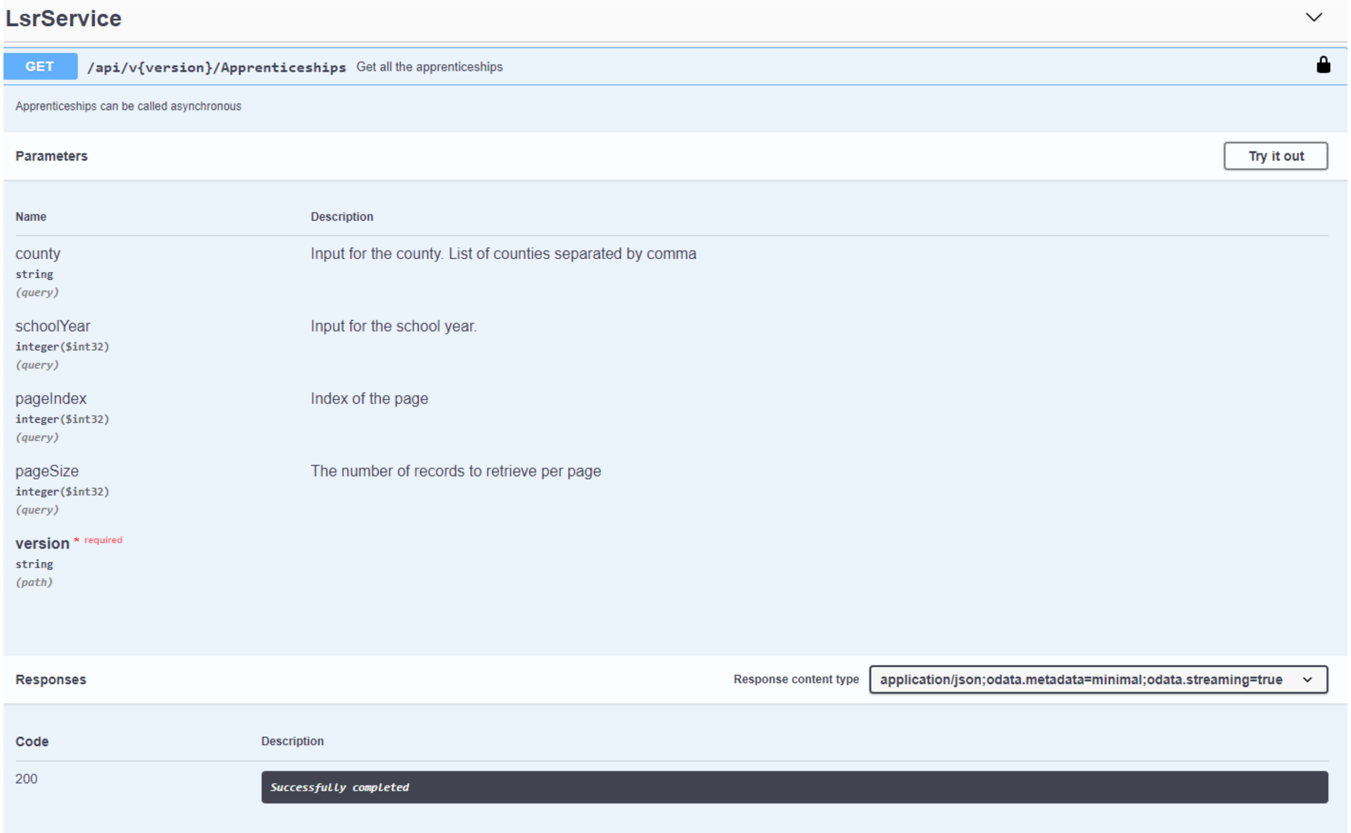
password: ••••••

type: Request body

client_id: LsrService

client_secret: be849d55-838b-46b8-8b6f-c(

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



LsrService

GET `/api/v{version}/Apprenticeships` Get all the apprenticeships

Apprenticeships can be called asynchronous

Parameters

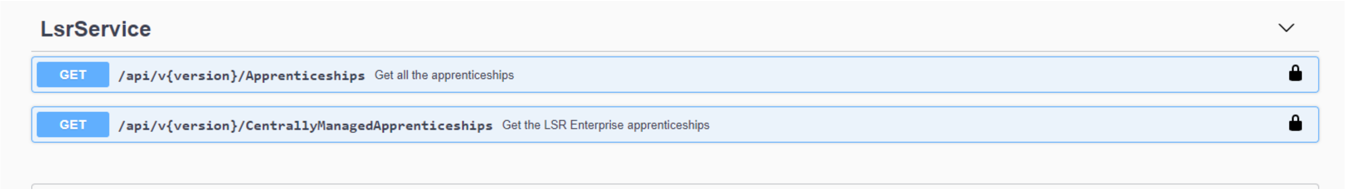
Name	Description
county string *(query)*	Input for the county. List of counties separated by comma
schoolYear integer($int32) *(query)*	Input for the school year.
pageIndex integer($int32) *(query)*	Index of the page
pageSize integer($int32) *(query)*	The number of records to retrieve per page
version * required string *(path)*	

Responses

Response content type: application/json;odata.metadata=minimal;odata.streaming=true

Code	Description
200	*Successfully completed*

Once authorised, swagger will display the two endpoints under LsrService as shown below:

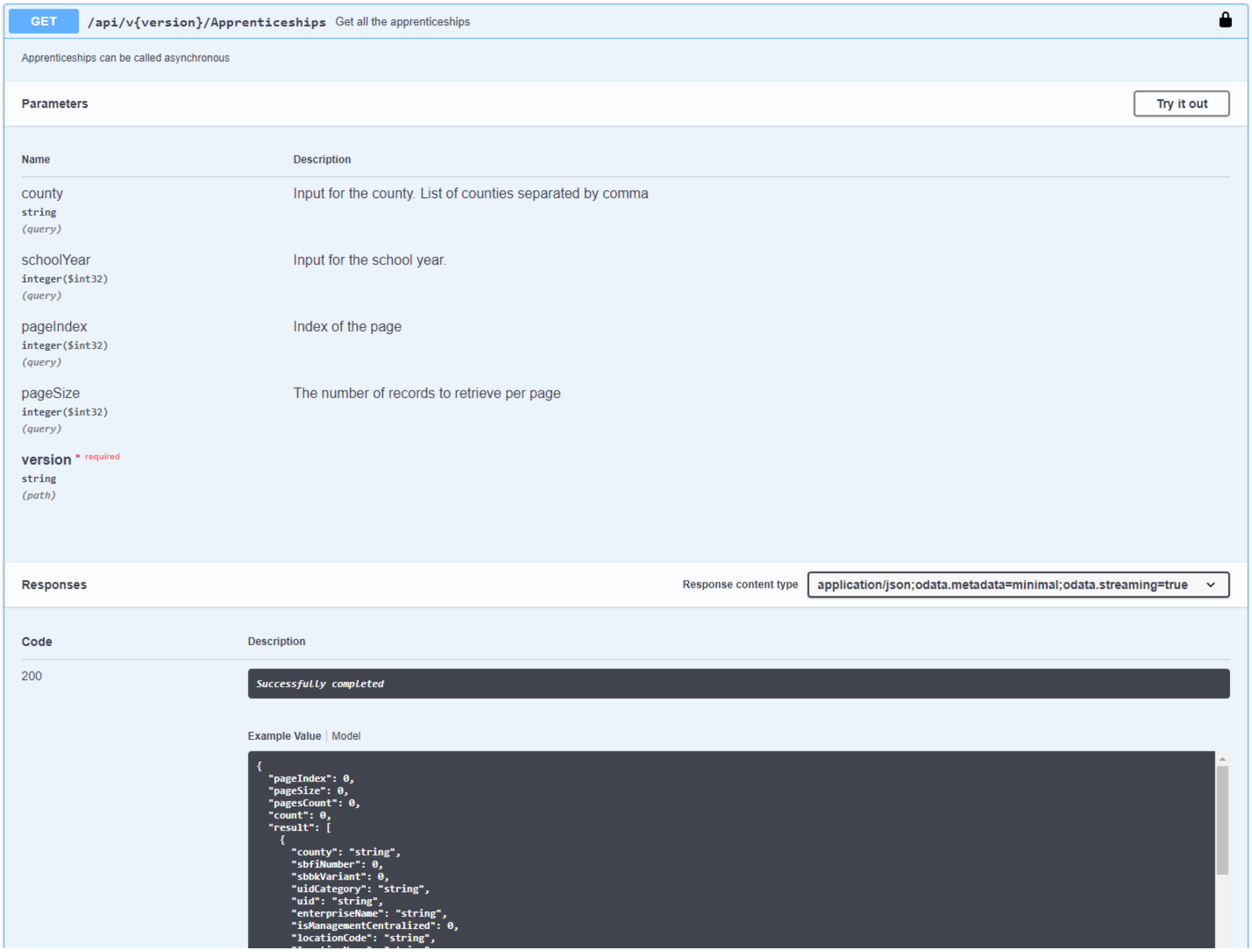


LsrService

GET `/api/v{version}/Apprenticeships` Get all the apprenticeships

GET `/api/v{version}/CentrallyManagedApprenticeships` Get the LSR Enterprise apprenticeships

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



GET `/api/v{version}/Apprenticeships` Get all the apprenticeships

Apprenticeships can be called asynchronous

Parameters

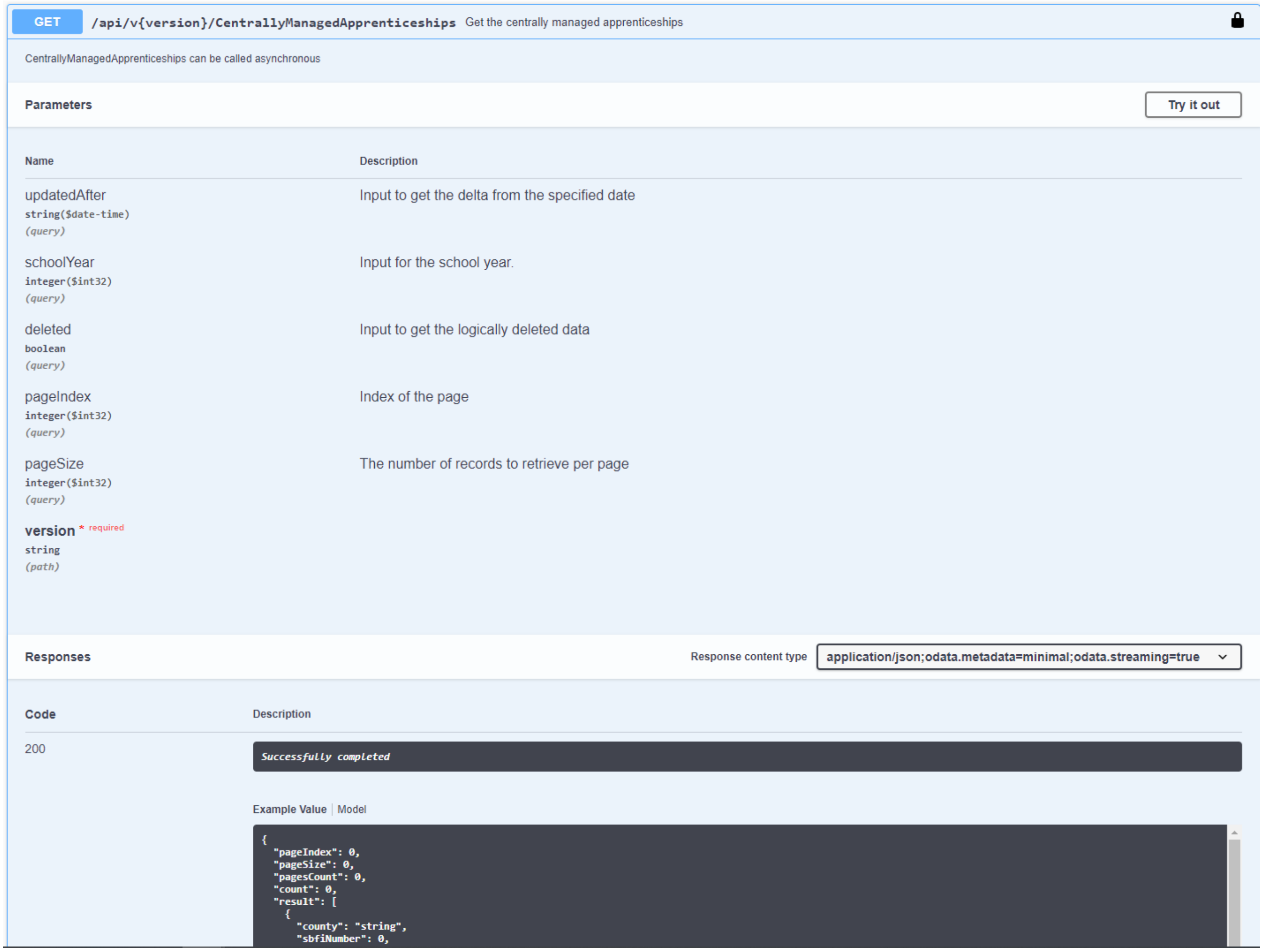
Name	Description
county `string` *(query)*	Input for the county. List of counties separated by comma
schoolYear `integer($int32)` *(query)*	Input for the school year.
pageIndex `integer($int32)` *(query)*	Index of the page
pageSize `integer($int32)` *(query)*	The number of records to retrieve per page
version * required `string` *(path)*	

Responses

Response content type: application/json;odata.metadata=minimal;odata.streaming=true

Code	Description
200	*Successfully completed*

```
{
  "pageIndex": 0,
  "pageSize": 0,
  "pagesCount": 0,
  "count": 0,
  "result": [
    {
      "county": "string",
      "sbfiNumber": 0,
      "sbbkVariant": 0,
      "uidCategory": "string",
      "uid": "string",
      "enterpriseName": "string",
      "isManagementCentralized": 0,
      "locationCode": "string",
```



`/api/v{version}/CentrallyManagedApprenticeships` Get the centrally managed apprenticeships

CentrallyManagedApprenticeships can be called asynchronous

Parameters

Name	Description
updatedAfter string($date-time) *(query)*	Input to get the delta from the specified date
schoolYear integer($int32) *(query)*	Input for the school year.
deleted boolean *(query)*	Input to get the logically deleted data
pageIndex integer($int32) *(query)*	Index of the page
pageSize integer($int32) *(query)*	The number of records to retrieve per page
version * required string *(path)*	

Responses

Response content type: application/json;odata.metadata=minimal;odata.streaming=true

Code	Description
200	*Successfully completed*

Example Value | Model

```
{
  "pageIndex": 0,
  "pageSize": 0,
  "pagesCount": 0,
  "count": 0,
  "result": [
    {
      "county": "string",
      "sbfiNumber": 0,
```

⚠**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

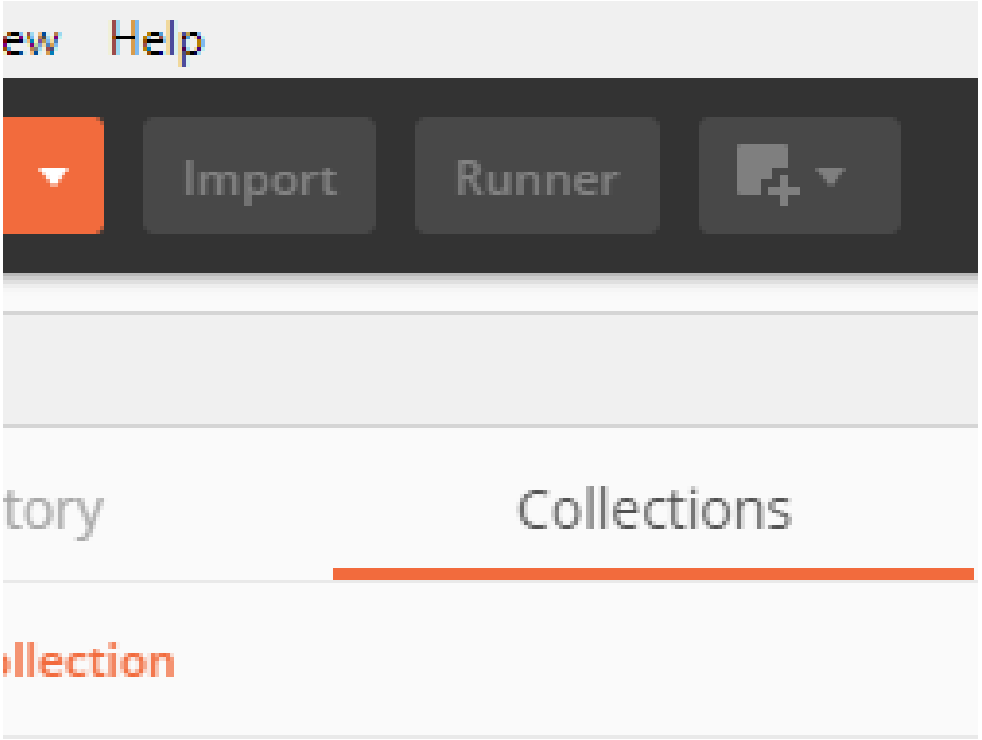
Postman is a free tool which offers graphical interface to send API requests.

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.

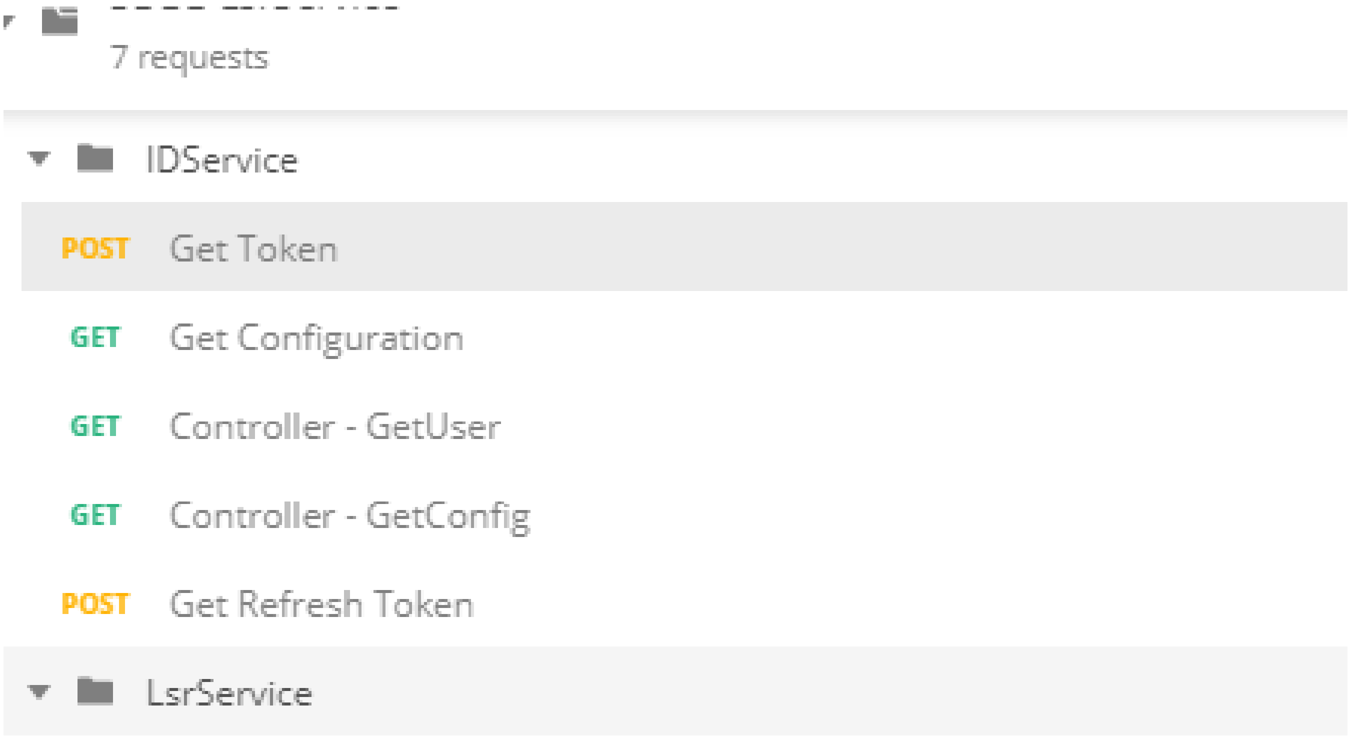
There are two json files named SDBB LsrService.postman_collection.json and SDBB LsrService.postman_environment.json which include the environment variables and pre-build in request for LsrService to get mock data for apprenticeships.

Example for importing LsrService 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
4. Click on Import and choose the two json files



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


7 requests

IDService

POST Get Token

GET Get Configuration

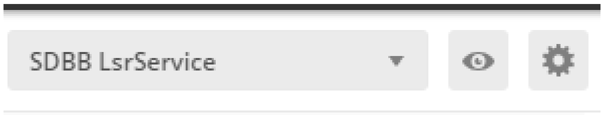
GET Controller - GetUser

GET Controller - GetConfig

POST Get Refresh Token

LsrService

Environment selected should be as shown below



6. SDBBLsrService contains pre-defined request for endpoints related to IDService and Lsr Service to get mock data.

7. SDBB LsrService 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

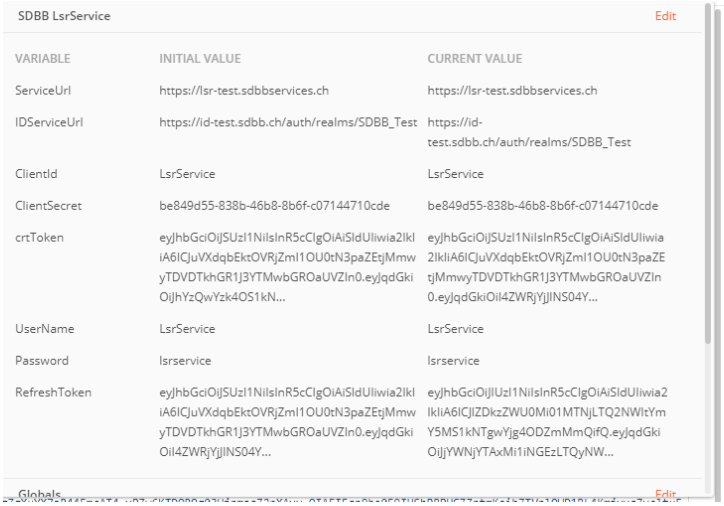
    4. ClientSecret - Client secret to get the token

    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

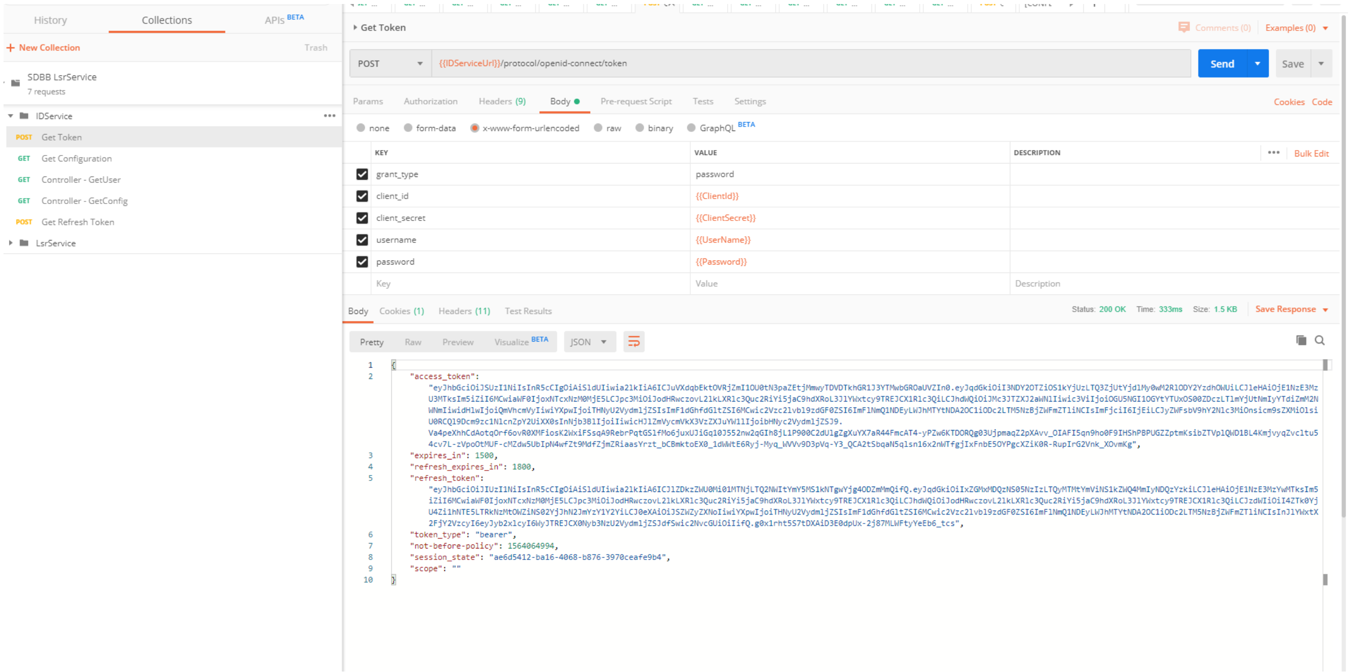

SDBB LsrService

VARIABLE	INITIAL VALUE	CURRENT VALUE
ServiceUrl	https://lsr-test.sdbbservices.ch	https://lsr-test.sdbbservices.ch
IDServiceUrl	https://id-test.sdbb.ch/auth/realms/SDBB_Test	https://id-test.sdbb.ch/auth/realms/SDBB_Test
ClientId	LsrService	LsrService
ClientSecret	be849d55-838b-46b8-8b6f-c07144710cde	be849d55-838b-46b8-8b6f-c07144710cde
crtToken	eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJuVXdqbEktOVRjZmI1OU0tN3paZEtjMmwyTDVDTkhGR1J3YTMwbGROaUVZIn0.eyJqdGkiOiJhYzQwYzk4OS1kN...	eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJuVXdqbEktOVRjZmI1OU0tN3paZEtjMmwyTDVDTkhGR1J3YTMwbGROaUVZIn0.eyJqdGkiOiI4ZWRjYjJlNS04Y...
UserName	LsrService	LsrService
Password	lsrservice	lsrservice
RefreshToken	eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJuVXdqbEktOVRjZmI1OU0tN3paZEtjMmwyTDVDTkhGR1J3YTMwbGROaUVZIn0.eyJqdGkiOiI4ZWRjYjJlNS04Y...	eyJhbGciOiJlUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJlZDkzZWU0Mi01MTNjLTQ2NWItYmY5MS1kNTgwYjg4ODZmMmQifQ.eyJqdGkiOiJjYWNjYTAxMi1iNGEzLTQyNW...

which will be used by IDService and LsrService.

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



POST {{IDServiceUrl}}/protocol/openid-connect/token

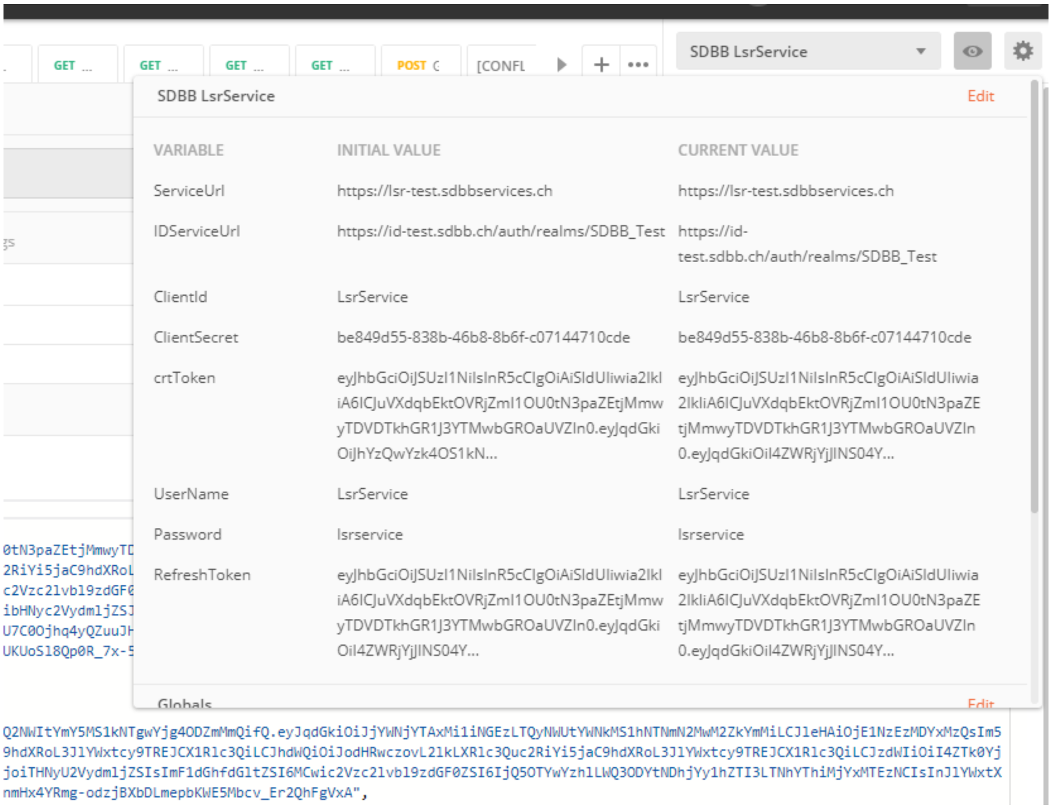
○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary ○ GraphQL BETA

	KEY	VALUE	DESCRIPTION
☑	grant_type	password	
☑	client_id	{{ClientId}}	
☑	client_secret	{{ClientSecret}}	
☑	username	{{UserName}}	
☑	password	{{Password}}	
	Key	Value	Description

Status: 200 OK Time: 333ms Size: 1.5 KB

```
{
    "access_token":
        "eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJuVXdqbEktOVRjZmI1OU0tN3paZEtjMmwyTDVDTkhGR1J3YTMwbGROaUVZIn0.eyJqdGkiOiI3NDY2OTZiOS1kYjUzLTQ3ZjUtYjdlMy0wM2RlODY2YzdhOWUiLCJleHAiOjE1NzE3Mz
        U3MTksIm5iZiI6MCwiaWF0IjoxNTcxNzM0MjE5LCJpc3MiOiJodHRwczovL2lkLXRlc3Quc2RiYi5jaC9hdXRoL3JlYWxtcy9TREJCX1Rlc3QiLCJhdWQiOiJMc3JTZXJ2aWNlIiwic3ViIjoiOGU5NGI1OGYtYTUxOS00ZDczLTlmYjUtNmIyYTdiZmM2N
        WNmIiwidHlwIjoiQmVhcmVyIiwiYXpwIjoiTHNyU2VydmljZSIsImF1dGhfdGltZSI6MCwic2Vzc2lvbl9zdGF0ZSI6ImF1NmQ1NDEyLWJhMTYtNDA2OC1iODc2LTM5NzBjZWFmZTliNCIsImFjciI6IjEiLCJyZWFsbV9hY2Nlc3MiOnsicm9sZXMiOlsi
        U0RCQl9Dcm9zc1NlcnZpY2UiXX0sInNjb3BlIjoiIiwicHJlZmVycmVkX3VzZXJuYW1lIjoibHNyc2VydmljZSJ9.
        Va4peXhhCdAotqOrf6ovR0XMFiosK2WxiFSsqA9RebrPqtGSlfMo6juxUJiGq10J552nw2qGIh8jL1P900C2dUlgZgXuYX7aR44FmcAT4-yPZw6KTDORQg03UjpmaqZ2pXAvv_OIAFI5qn9ho0F9IHShPBPUGZZptmKsibZTVplQWD1BL4KmjvyqZvcltu5
        4cv7L-zVpoOtMUF-cMZdw5UbIpN4wfZt9MdfZjmZRiaasYrzt_bCBmktoEX0_1dWWtE6Ryj-Myq_WVVv9D3pVq-Y3_QCA2tSbqaN5qlsn16x2nWTfgjIxFnbE5OYPgcXZiK0R-RupIrG2Vnk_XOvmKg",
    "expires_in": 1500,
    "refresh_expires_in": 1800,
    "refresh_token":
        "eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJlZDkzZWU0Mi01MTNjLTQ2NWItYmY5MS1kNTgwYjg4ODZmMmQifQ.eyJqdGkiOiIxZGMxMDQzNS05NzIzLTQyMTMtYmViNS1kZWQ4MmIyNDQzYzkiLCJleHAiOjE1NzE3MzYwMTksIm5
        iZiI6MCwiaWF0IjoxNTcxNzM0MjE5LCJpc3MiOiJodHRwczovL2lkLXRlc3Quc2RiYi5jaC9hdXRoL3JlYWxtcy9TREJCX1Rlc3QiLCJhdWQiOiJodHRwczovL2lkLXRlc3Quc2RiYi5jaC9hdXRoL3JlYWxtcy9TREJCX1Rlc3QiLCJzdWIiOiI4ZTk0Yj
        U4Zi1hNTE5LTRkNzMtOWZiNS02YjJhN2JmYzY1Y2YiLCJ0eXAiOiJSZWZyZXNoIiwiYXpwIjoiTHNyU2VydmljZSIsImF1dGhfdGltZSI6MCwic2Vzc2lvbl9zdGF0ZSI6ImF1NmQ1NDEyLWJhMTYtNDA2OC1iODc2LTM5NzBjZWFmZTliNCIsInJlYWxtX
        2FjY2VzcyI6eyJyb2xlcyI6WyJTREJCX0Nyb3NzU2VydmljZSJdfSwic2NvcGUiOiIifQ.g0x1rht5S7tDXAiD3E0dpUx-2j87MLWFtyYeEb6_tcs",
    "token_type": "bearer",
    "not-before-policy": 1564064994,
    "session_state": "ae6d5412-ba16-4068-b876-3970ceafe9b4",
    "scope": ""
}
```

Copy the access_token from above and paste it in crtToken variable in LsrService Environment as shown below



SDBB LsrService

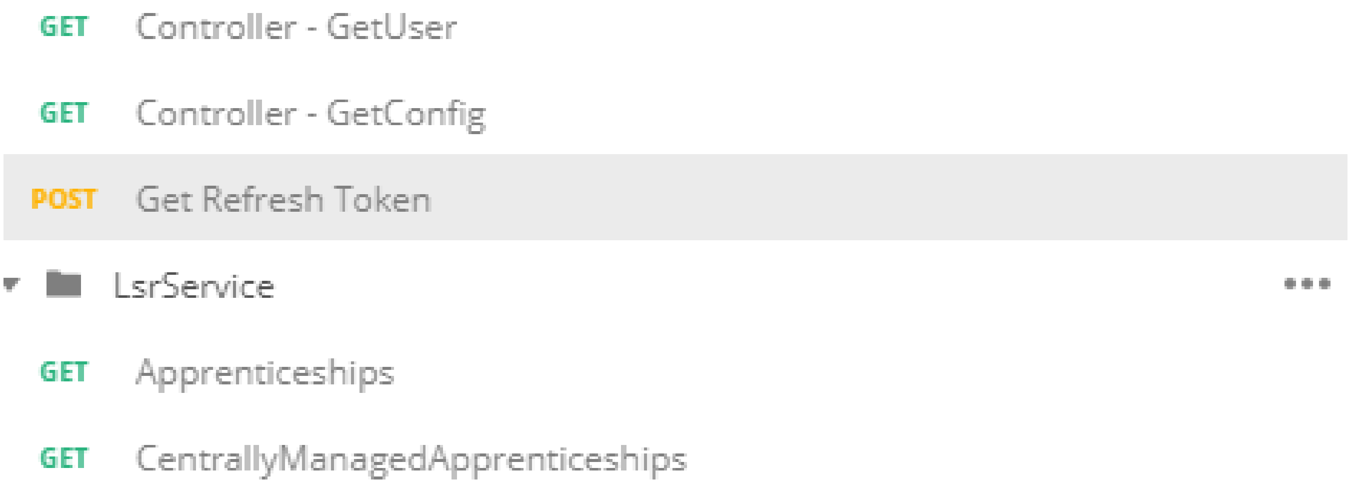
VARIABLE	INITIAL VALUE	CURRENT VALUE
ServiceUrl	https://lsr-test.sdbbservices.ch	https://lsr-test.sdbbservices.ch
IDServiceUrl	https://id-test.sdbb.ch/auth/realms/SDBB_Test	https://id-test.sdbb.ch/auth/realms/SDBB_Test
ClientId	LsrService	LsrService
ClientSecret	be849d55-838b-46b8-8b6f-c07144710cde	be849d55-838b-46b8-8b6f-c07144710cde
crtToken	eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJuVXdqbEktOVRjZmI1OU0tN3paZEtjMmwyTDVDTkhGR1J3YTMwbGROaUVZIn0.eyJqdGkiOiJhYzQwYzk4OS1kN...	eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJuVXdqbEktOVRjZmI1OU0tN3paZEtjMmwyTDVDTkhGR1J3YTMwbGROaUVZIn0.eyJqdGkiOiI4ZWRjYjJlNS04Y...
UserName	LsrService	LsrService
Password	lsrservice	lsrservice
RefreshToken	eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJuVXdqbEktOVRjZmI1OU0tN3paZEtjMmwyTDVDTkhGR1J3YTMwbGROaUVZIn0.eyJqdGkiOiI4ZWRjYjJlNS04Y...	eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJuVXdqbEktOVRjZmI1OU0tN3paZEtjMmwyTDVDTkhGR1J3YTMwbGROaUVZIn0.eyJqdGkiOiI4ZWRjYjJlNS04Y...

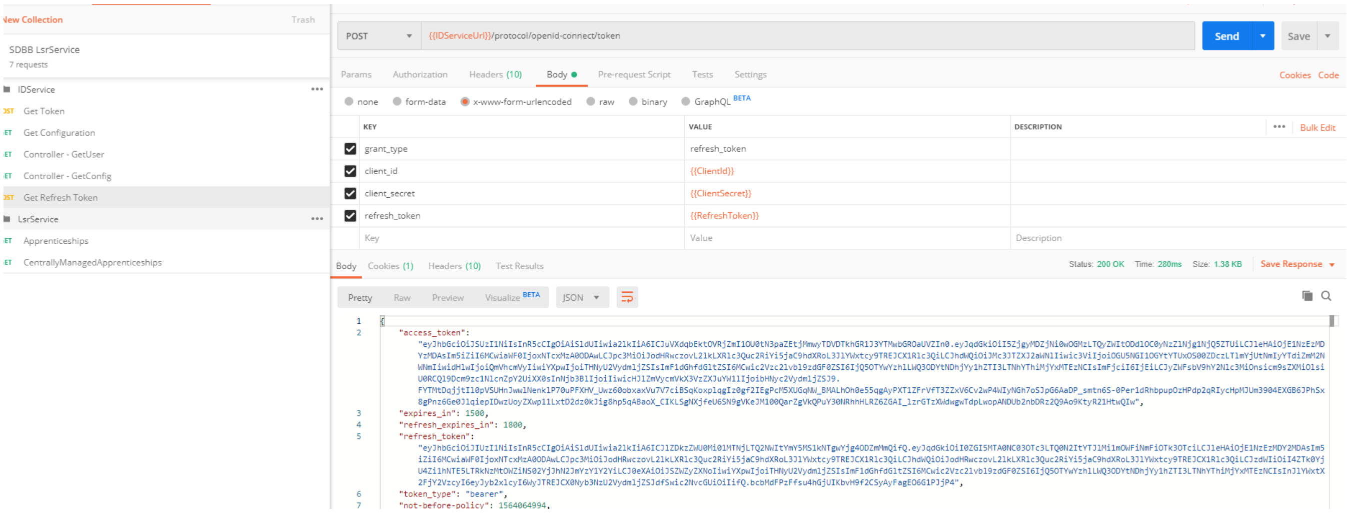
```
0tN3paZEtjMmwyTD
2RiYi5jaC9hdXRoL
c2Vzc2lvbl9zdGF0
ibHNyc2VydmljZSJ
U7C0Ojhq4yQZuuJH
UKUoSl8Qp0R_7x-5

Q2NWItYmY5MS1kNTgwYjg4ODZmMmQifQ.eyJqdGkiOiJjYWNjYTAxMi1iNGEzLTQyNWUtYWNkMS1hNTNmN2MwM2ZkYmMiLCJleHAiOjE1NzEzMDYxMzQsIm5
9hdXRoL3JlYWxtcy9TREJCX1Rlc3QiLCJhdWQiOiJodHRwczovL2lkLXRlc3Quc2RiYi5jaC9hdXRoL3JlYWxtcy9TREJCX1Rlc3QiLCJzdWIiOiI4ZTk0Yj
joiTHNyU2VydmljZSIsImF1dGhfdGltZSI6MCwic2Vzc2lvbl9zdGF0ZSI6IjQ5OTYwYzhlLWQ3ODYtNDhjYy1hZTI3LTNhYThiMjYxMTEzNCIsInJlYWxtX
nmHx4YRmg-odzjBXbDLmepbKWE5Mbcv_Er2QhFgVxA",
```

Example to call Refresh Token

Use the refresh_token in the above pics to get an access token without login page

Get Refresh Token





POST {{IDServiceUrl}}/protocol/openid-connect/token

○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary ○ GraphQL BETA

	KEY	VALUE	DESCRIPTION
☑	grant_type	refresh_token	
☑	client_id	{{ClientId}}	
☑	client_secret	{{ClientSecret}}	
☑	refresh_token	{{RefreshToken}}	
	Key	Value	Description

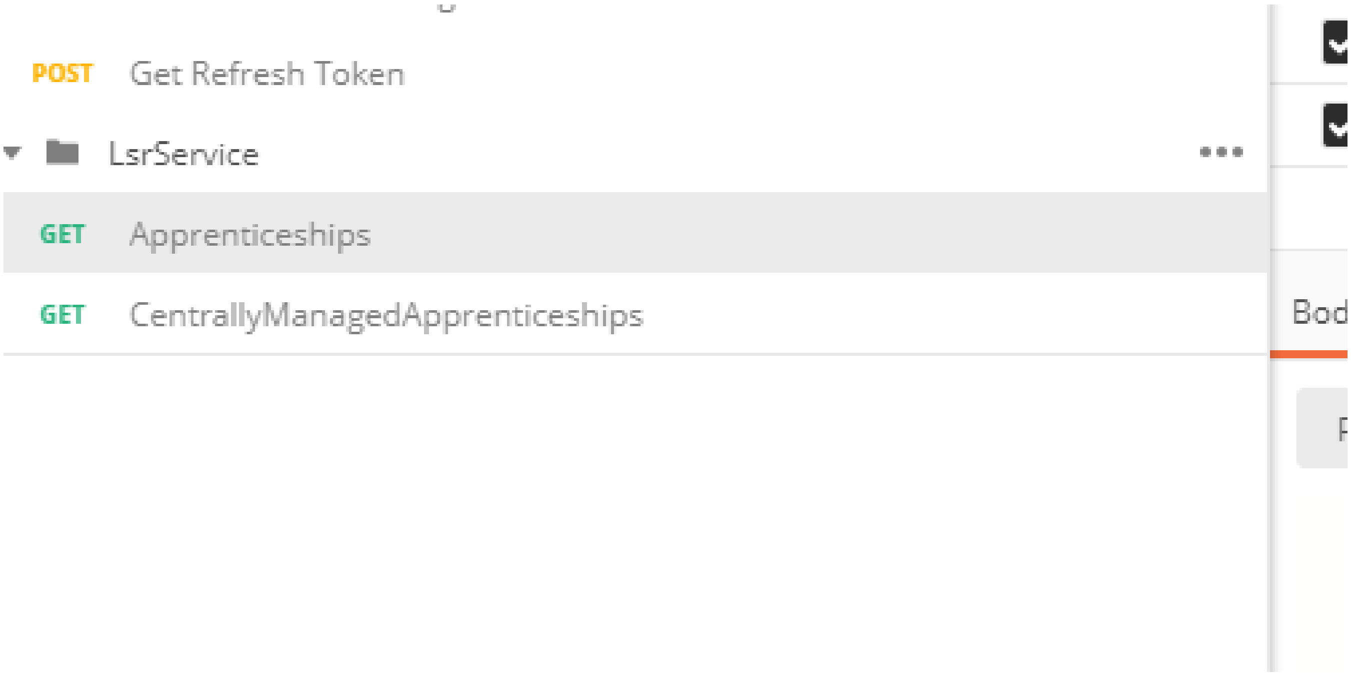
Body Cookies (1) Headers (10) Test Results

Status: 200 OK Time: 280ms Size: 1.38 KB Save Response

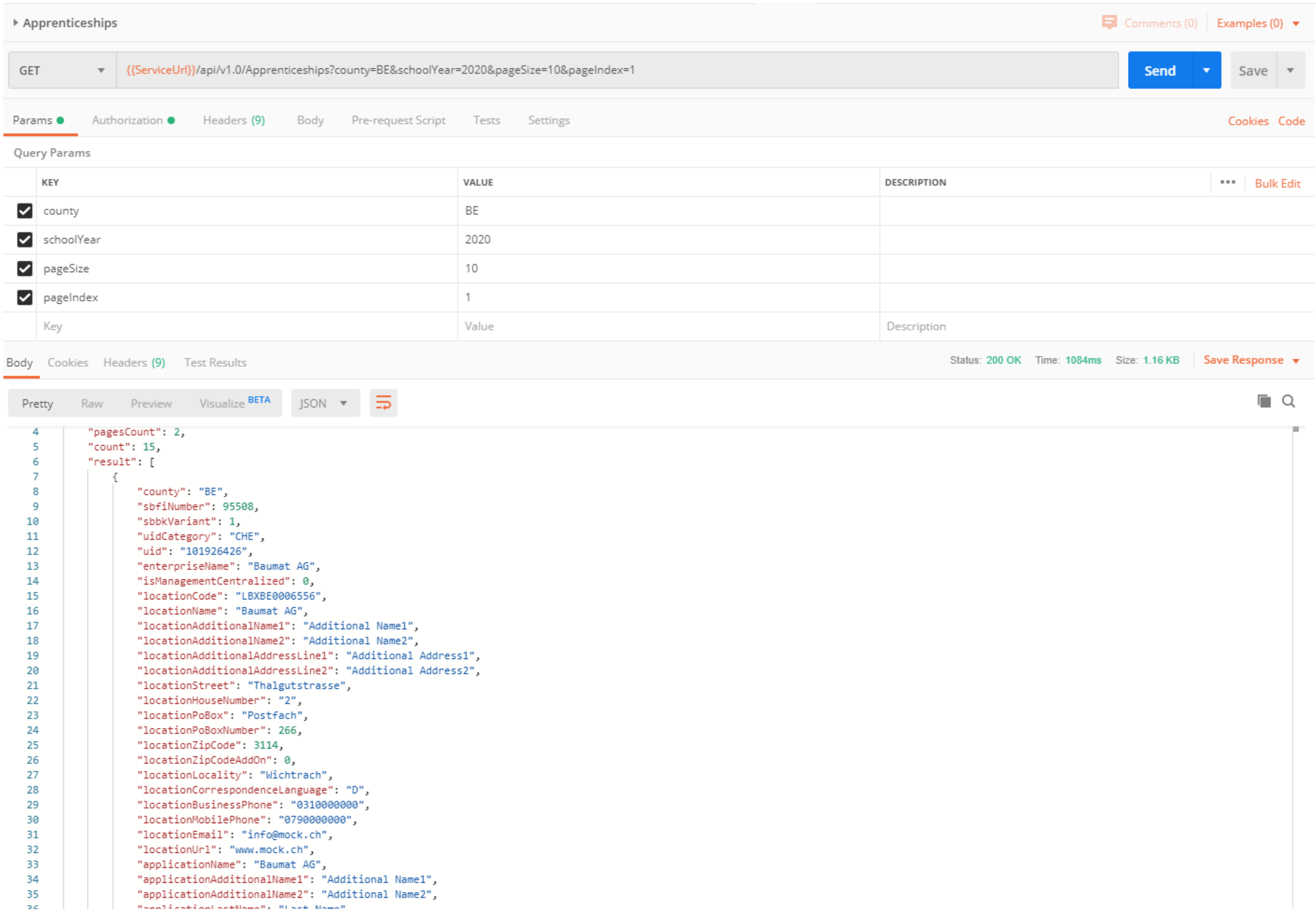
```
{
    "access_token":
        "eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJuVXdqbEktOVRjZmI1OU0tN3paZEtjMmwyTDVDTkhGR1J3YTMwbGROaUVZIn0.eyJqdGkiOiI5ZjgyMDZjNi0wOGMzLTQyZWItODdlOC0yNzZlNjg1NjQ5ZTUiLCJleHAiOjE1NzEzMD
        YzMDAsIm5iZiI6MCwiaWF0IjoxNTcxMzA0ODAwLCJpc3MiOiJodHRwczovL2lkLXRlc3Quc2RiYi5jaC9hdXRoL3JlYWxtcy9TREJCX1Rlc3QiLCJhdWQiOiJMc3JTZXJ2aWNlIiwic3ViIjoiOGU5NGI1OGYtYTUxOS00ZDczLTlmYjUtNmIyYTdiZmM2N
        WNmIiwidHlwIjoiQmVhcmVyIiwiYXpwIjoiTHNyU2VydmljZSIsImF1dGhfdGltZSI6MCwic2Vzc2lvbl9zdGF0ZSI6IjQ5OTYwYzhlLWQ3ODYtNDhjYy1hZTI3LTNhYThiMjYxMTEzNCIsImFjciI6IjEiLCJyZWFsbV9hY2Nlc3MiOnsicm9sZXMiOlsi
        U0RCQl9Dcm9zc1NlcnZpY2UiXX0sInNjb3BlIjoiIiwicHJlZmVycmVkX3VzZXJuYW1lIjoibHNyc2VydmljZSJ9.
        FYTMtDqjjtIl0pVSUHnJwwlNenklP70uPFXHV_Uwz60obxaxVu7V7ciBSqKoxplqgIz0gf2IEgPcM5XUGqNW_BMALhOh0e55qgAyPXT1ZFrVfT3ZZxV6Cv2wP4WIyNGh7oSJpG6AaDP_smtn6S-0Per1dRhbpupOzHPdp2qRIycHpMJUm3904EXGB6JPhSx
        8gPnz6Ge0JlqiepIDwzUoyZXwp11LxtD2dz0kJig8hp5qABaoX_CIKLSgNXjfeU6SN9gVKeJM100QarZgVkQPuY30NRhhHLRZ6ZGAI_lzrGTzXWdwgwTdpLwopANDUb2nbDRz2Q9Ao9KtyR21HtwQIw",
    "expires_in": 1500,
    "refresh_expires_in": 1800,
    "refresh_token":
        "eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJlZDkzZWU0Mi01MTNjLTQ2NWItYmY5MS1kNTgwYjg4ODZmMmQifQ.eyJqdGkiOiI0ZGI5MTA0NC03OTc3LTQ0N2ItYTJlMi1mOWFiNmFiOTk3OTciLCJleHAiOjE1NzEzMDY2MDAsIm5
        iZiI6MCwiaWF0IjoxNTcxMzA0ODAwLCJpc3MiOiJodHRwczovL2lkLXRlc3Quc2RiYi5jaC9hdXRoL3JlYWxtcy9TREJCX1Rlc3QiLCJhdWQiOiJodHRwczovL2lkLXRlc3Quc2RiYi5jaC9hdXRoL3JlYWxtcy9TREJCX1Rlc3QiLCJzdWIiOiI4ZTk0Yj
        U4Zi1hNTE5LTRkNzMtOWZiNS02YjJhN2JmYzY1Y2YiLCJ0eXAiOiJSZWZyZXNoIiwiYXpwIjoiTHNyU2VydmljZSIsImF1dGhfdGltZSI6MCwic2Vzc2lvbl9zdGF0ZSI6IjQ5OTYwYzhlLWQ3ODYtNDhjYy1hZTI3LTNhYThiMjYxMTEzNCIsInJlYWxtX
        2FjY2VzcyI6eyJyb2xlcyI6WyJTREJCX0Nyb3NzU2VydmljZSJdfSwic2NvcGUiOiIifQ.bcbMdFPzFfsu4hGjUIKbvH9f2CSyAyFagEO6G1PJjP4",
    "token_type": "bearer",
    "not-before-policy": 1564064994,
```

Example to call API endpoint for mock data

1. Get All Apprenticeships :



Click on Send



GET {{ServiceUrl}}/api/v1.0/Apprenticeships?county=BE&schoolYear=2020&pageSize=10&pageIndex=1

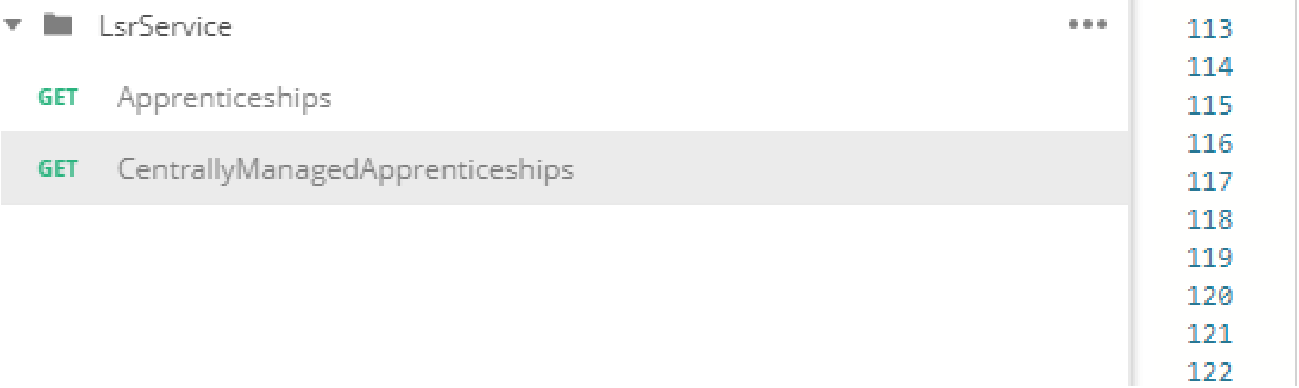
Query Params

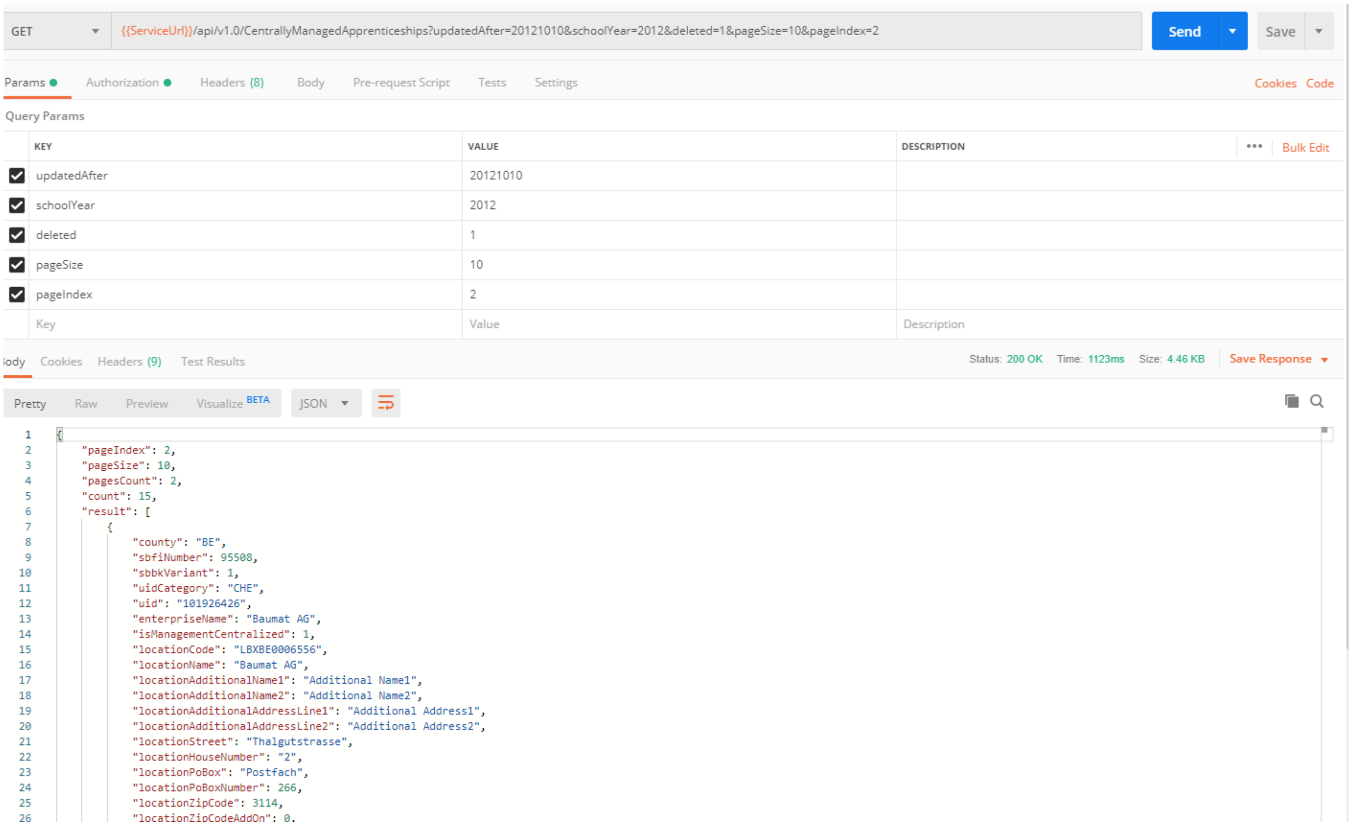
	KEY	VALUE	DESCRIPTION
☑	county	BE	
☑	schoolYear	2020	
☑	pageSize	10	
☑	pageIndex	1	
	Key	Value	Description

Status: 200 OK Time: 1084ms Size: 1.16 KB

```
    "pagesCount": 2,
    "count": 15,
    "result": [
        {
            "county": "BE",
            "sbfiNumber": 95508,
            "sbbkVariant": 1,
            "uidCategory": "CHE",
            "uid": "101926426",
            "enterpriseName": "Baumat AG",
            "isManagementCentralized": 0,
            "locationCode": "LBXBE0006556",
            "locationName": "Baumat AG",
            "locationAdditionalName1": "Additional Name1",
            "locationAdditionalName2": "Additional Name2",
            "locationAdditionalAddressLine1": "Additional Address1",
            "locationAdditionalAddressLine2": "Additional Address2",
            "locationStreet": "Thalgutstrasse",
            "locationHouseNumber": "2",
            "locationPoBox": "Postfach",
            "locationPoBoxNumber": 266,
            "locationZipCode": 3114,
            "locationZipCodeAddOn": 0,
            "locationLocality": "Wichtrach",
            "locationCorrespondenceLanguage": "D",
            "locationBusinessPhone": "0310000000",
            "locationMobilePhone": "0790000000",
            "locationEmail": "info@mock.ch",
            "locationUrl": "www.mock.ch",
            "applicationName": "Baumat AG",
            "applicationAdditionalName1": "Additional Name1",
            "applicationAdditionalName2": "Additional Name2",
```

2. Centrally Managed Apprenticeships





GET {{ServiceUrl}}/api/v1.0/CentrallyManagedApprenticeships?updatedAfter=20121010&schoolYear=2012&deleted=1&pageSize=10&pageIndex=2

Query Params

	KEY	VALUE	DESCRIPTION
☑	updatedAfter	20121010	
☑	schoolYear	2012	
☑	deleted	1	
☑	pageSize	10	
☑	pageIndex	2	
	Key	Value	Description

Status: 200 OK Time: 1123ms Size: 4.46 KB

```
{
    "pageIndex": 2,
    "pageSize": 10,
    "pagesCount": 2,
    "count": 15,
    "result": [
        {
            "county": "BE",
            "sbfiNumber": 95508,
            "sbbkVariant": 1,
            "uidCategory": "CHE",
            "uid": "101926426",
            "enterpriseName": "Baumat AG",
            "isManagementCentralized": 1,
            "locationCode": "LBXBE0006556",
            "locationName": "Baumat AG",
            "locationAdditionalName1": "Additional Name1",
            "locationAdditionalName2": "Additional Name2",
            "locationAdditionalAddressLine1": "Additional Address1",
            "locationAdditionalAddressLine2": "Additional Address2",
            "locationStreet": "Thalgutstrasse",
            "locationHouseNumber": "2",
            "locationPoBox": "Postfach",
            "locationPoBoxNumber": 266,
            "locationZipCode": 3114,
            "locationZipCodeAddOn": 0,
```

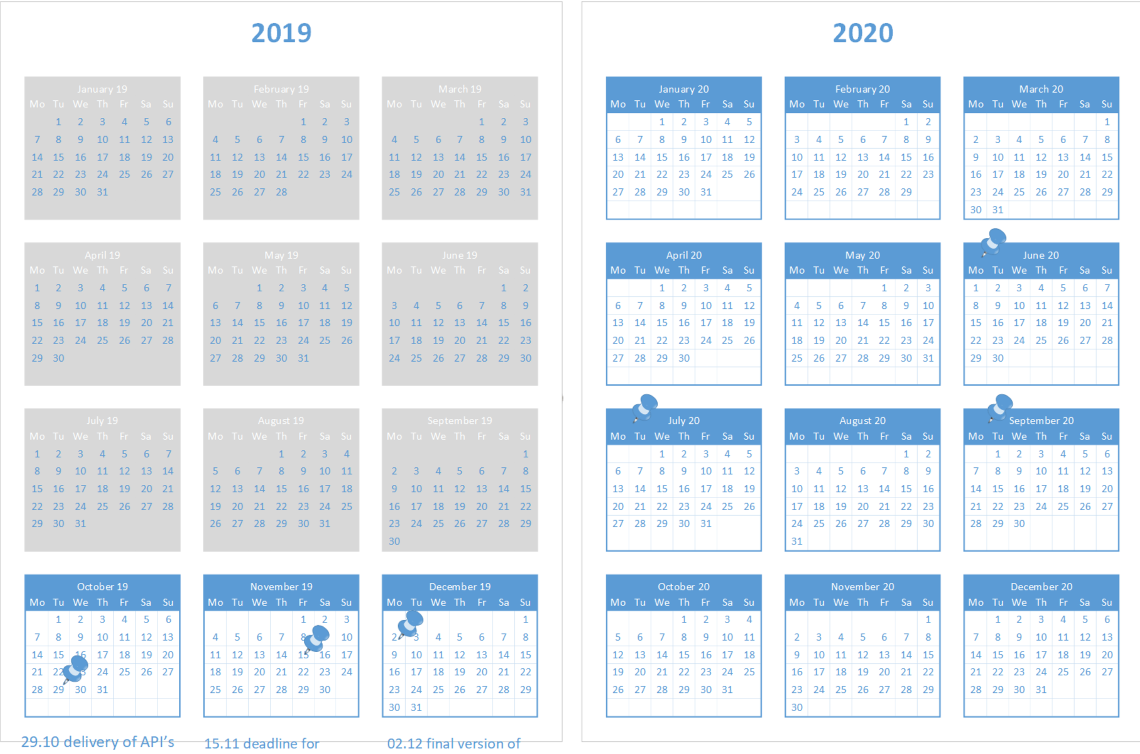
API planning

Steps to consider:

- Documentation
  - Delivery of documentation to the cantons by SDBB
  - Provision of feedback by the cantons
  - Provision of a final version of the documentation by SDBB
- **MOCK**
  - Provision of a MOCK to the cantons in order to carry out tests and obtain dummy data in the response. ⚠️ **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 cantons.**
  - The two endpoints which return mock data are included in the collection and environment json files for lsrservice 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. The user's canton will also not be taken into account to filter the data
- 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 large companies

The main milestones are represented below:



June 2020 Launching acceptance testing on QUAL

July 2020 End of user tests

September 2020 Go Live API

29.10 delivery of API's documentation by SDBB

15.11 deadline for feedback from the cantons

02.12 final version of the documentation and delivery of a MOCK