

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
Lehrtort	Apprenticeship location
Betriebliche Bildung (Lehrtort + 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, n° 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 Connect (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

Available authorizations



Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes. API requires the following scopes. Select which ones you want to grant to Swagger UI.

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:

password:

type:

Request body

client_id:

client_secret:

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

Authorize

Keycloak

GET /api/v{version}/Keycloak/Config Gets the configuration.

GET /api/v{version}/Keycloak/User Get the authenticated User.

POST /api/v{version}/Keycloak/Bearer Gets the bearer token for the current service, as per SDBB Common BusinessLayer.Configurations.KeycloakConfiguration, asynchronously.

SearchService

GET /api/v{version}/ExportApprenticeshipPlaces Export the apprenticeship places with paging as per search criteria for export clients only when number of open places is greater than zero

GET /api/v{version}/ExportApprenticeships Export the apprenticeships with paging as per search criteria for export client

Models

KeycloakConfiguration >

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

SearchService

GET /api/v{version}/ExportApprenticeshipPlaces Export the apprenticeship places with paging as per search criteria for export clients only when number of open places is greater than zero

GET /api/v{version}/ExportApprenticeships Export the apprenticeships with paging as per search criteria for export client

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

GET /api/v{version}/ExportApprenticeshipPlaces Export the apprenticeship places with paging as per search criteria for export clients only when number of open places is greater than zero

ExportApprenticeshipPlaces can be called asynchronous

Parameters Try it out

Name	Description
countyCodes string (query)	List of County Codes as input which are separated by comma
sbfNumbers string (query)	List of SBF1 numbers for professions as input which are separated by comma
swissDocNumbers string (query)	List of Swissdoc numbers for professions as input which are 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: text/plain

Code	Description
200	<p>Successfully completed</p> <p>Example Value Model</p> <pre>{ "pageIndex": 0, "pageSize": 0, "pagesCount": 0, "count": 0 }</pre>

GET /api/v{version}/ExportApprenticeships Export the apprenticeships with paging as per search criteria for export client

ExportApprenticeships can be called asynchronous

Parameters Try it out

Name	Description
countyCodes string (query)	List of County Codes as input which are separated by comma
sbfNumbers string (query)	List of SBF1 numbers for professions as input which are separated by comma
swissDocNumbers string (query)	List of Swissdoc numbers for professions as input which are 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: text/plain

Code	Description
200	<p>Successfully completed</p> <p>Example Value Model</p> <pre>{ "pageIndex": 0, "pageSize": 0, "pagesCount": 0, "count": 0 }</pre>

⚠ 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

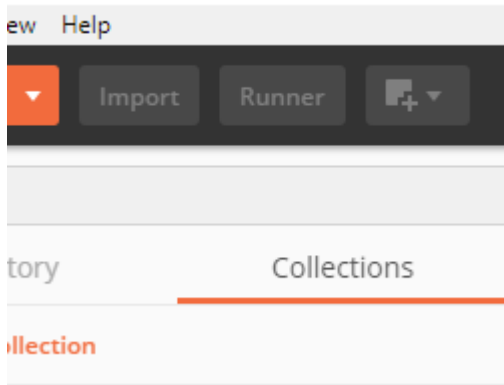
Postman is a free tool which offers graphical interface to send API requests.

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.

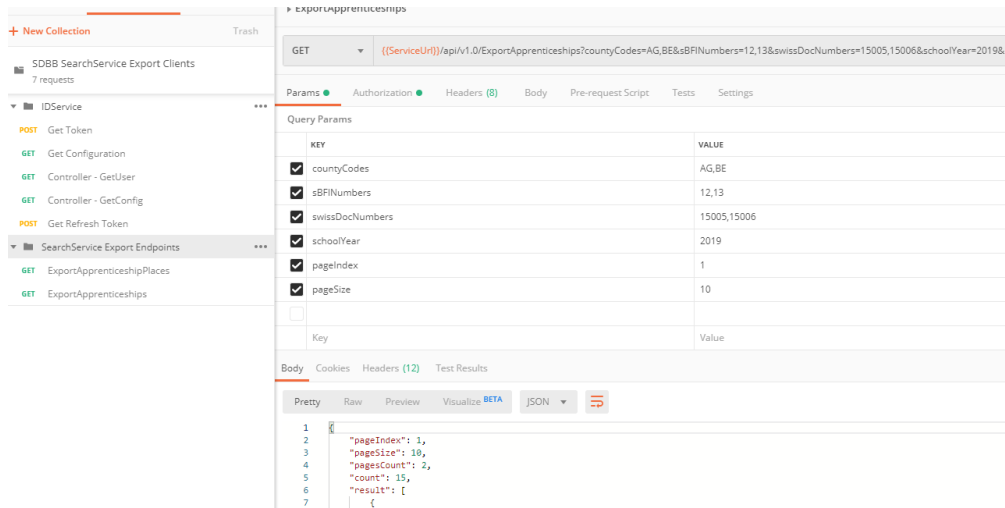
There are two json files named SDBB SearchServiceExportClients.postman_collection.json and SDBB SDBB SearchServiceExportClients.postman_environment.json which include the environment variables and pre-build in request for SearchService to get mock data of apprenticeships and apprenticeship places for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

Example for importing SearchService Collections and environment as shown below

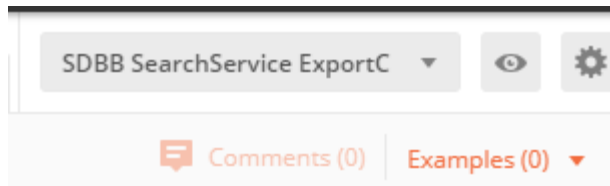
1. Download and install Postman from the link <https://www.getpostman.com/downloads/>
2. After that open Postman App
3. Follow the steps below to open the above two json files SDBB SearchServiceExportClients.postman_collection.json and SDBB SDBB SearchServiceExportClients.postman_environment.json provided 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.

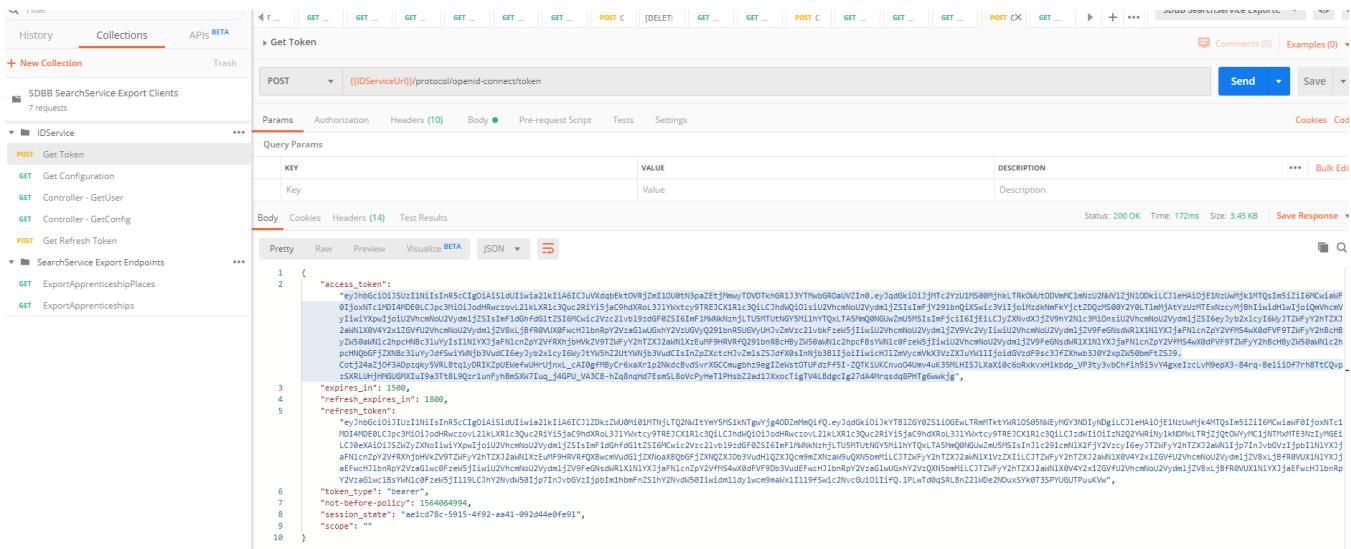
2. IDServiceUrl - URL of the ID Service provider for authentication and authorization
3. ClientId - Client Id to get the token
4. ClientSecret - Client secret to get the token
5. UserName - user name
6. Password - password
7. crtToken - The bearer token which will be used for API request for authorization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

SDBB SearchService ExportClients			Edit
VARIABLE	INITIAL VALUE	CURRENT VALUE	
ServiceUrl	https://lsr-search-test.sdbbervices.ch/	https://lsr-search-test.sdbbervices.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	eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUiwiwia2IklIA6ICjuVXdqbEktOVRjZm11OU0tN3paZEtmMmwyTDVDTkhGR1J3YTMwbGROaUVZln0.eyjQdGkiOijhYzQwYzk4OS1kN...	eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUiwiwia2IklIA6ICjuVXdqbEktOVRjZm11OU0tN3paZEtmMmwyTDVDTkhGR1J3YTMwbGROaUVZln0.eyjQdGkiOijjMTc2YzU1MS00M...	
UserName	test_lsr_exportclientname	test_lsr_exportclientname	
Password	123456	123456	
RefreshToken	eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUiwiwia2IklIA6ICjuVXdqbEktOVRjZm11OU0tN3paZEtmMmwyTDVDTkhGR1J3YTMwbGROaUVZln0.eyjQdGkiOijjINS04Y...	eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUiwiwia2IklIA6ICjuVXdqbEktOVRjZm11OU0tN3paZEtmMmwyTDVDTkhGR1J3YTMwbGROaUVZln0.eyjQdGkiOijjINS04Y...	
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



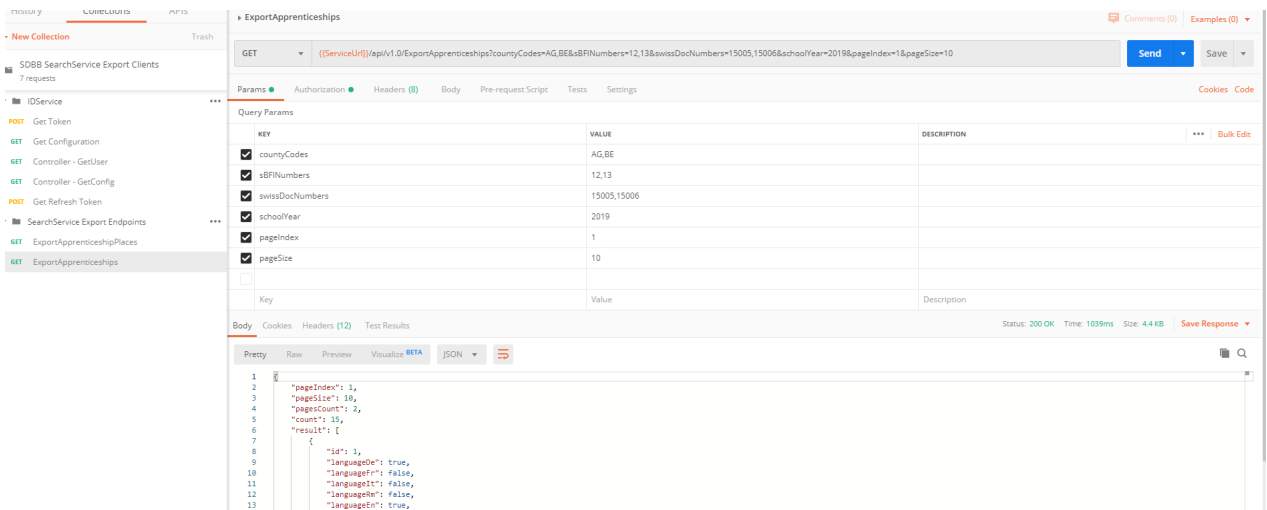
Copy the access_token from above and paste it in crtToken variable in SearchService Export Client Environment as shown below

SDBB SearchService ExportClients		Edit
VARIABLE	INITIAL VALUE	CURRENT VALUE
ServiceUrl	https://lsr-search-test.sdbbtest.com/	https://lsr-search-test.sdbbtest.com/
IDServiceUrl	https://id-test.sdbbtest.com/auth/realms/SDBB_Test	https://id-test.sdbbtest.com/auth/realms/SDBB_Test
ClientId	SearchService	SearchService
ClientSecret	0a05b3cb-a600-45a1-af2f-8640f99ccb8b	0a05b3cb-a600-45a1-af2f-8640f99ccb8b
crtToken	eyJhbGciOiJIUzI1NiIsInR5cCI6IWRX	eyJhbGciOiJIUzI1NiIsInR5cCI6IWRX
UserName	test_lsr_exportclientname	test_lsr_exportclientname
Password	123456	123456
RefreshToken	eyJhbGciOiJIUzI1NiIsInR5cCI6IWRX	eyJhbGciOiJIUzI1NiIsInR5cCI6IWRX

Example to call Refresh Token

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

Get Refresh Token



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. **⚠ 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:

2019

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Mo	Tu	We	Th	Fr	Sa	Su
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28			

Mo	Tu	We	Th	Fr	Sa	Su
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Mo	Tu	We	Th	Fr	Sa	Su
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Mo	Tu	We	Th	Fr	Sa	Su
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Mo	Tu	We	Th	Fr	Sa	Su
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Mo	Tu	We	Th	Fr	Sa	Su
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Mo	Tu	We	Th	Fr	Sa	Su
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

17.01 final version of the documentation and delivery of a MOCK

2020

Mo	Tu	We	Th	Fr	Sa	Su
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Mo	Tu	We	Th	Fr	Sa	Su
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	

Mo	Tu	We	Th	Fr	Sa	Su
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Mo	Tu	We	Th	Fr	Sa	Su
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Mo	Tu	We	Th	Fr	Sa	Su
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Mo	Tu	We	Th	Fr	Sa	Su
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Mo	Tu	We	Th	Fr	Sa	Su
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Mo	Tu	We	Th	Fr	Sa	Su
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Mo	Tu	We	Th	Fr	Sa	Su
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

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

29.11 delivery of API's documentation by SDBB

20.12 deadline for feedback from the clients