12.09.2019

# LCM-Webservice-API

## Aktuelle Webservice-Adressen

* PROD: [http://storage.internal-local.ch](https://slack-redir.net/link?url=http%3A%2F%2Fstorage.internal-local.ch&v=3)
* Staging (TER, TEP): <http://storage-stg.preprod-local.ch>
* DEV: [http://storage-dev.preprod-local.ch](https://slack-redir.net/link?url=http%3A%2F%2Fstorage-dev.preprod-local.ch&v=3)

## Authentifizierung

Die Authentifizierung erfolgt über folgenden Link:

<https://auth.staging.local.ch/oauth/check_token>

ClientId (Username) ist in jedem Fall – ob DEV, TER, TEP oder PROD – immer lcm

In lcm-example-deployment.properties ist ein verschlüsseltes Passwort (oauth.clientSecret) angegeben > Klartext?

## 12.09.2019: URL Aenderung in places.json und doc.html

alt:

<http://datastore-stg.lb-service/v2/places/GMv21tkh3U4pTYa3u71WXw>

neu:

<http://storage.internal-local.ch/v2/places/GMv21tkh3U4pTYa3u71WXw>

places.json

{

"href": "http://127.0.0.1:8080/bsicrmserver/rest/v1/customers/16773896/places",

"offset": 0,

"limit": 50,

"items": [

{

"href": "http://storage.internal-local.ch/v2/places/GMv21tkh3U4pTYa3u71WXw"

}

]

}

## Dokumentation der Webservices

Die Webservices sind unter dem Link

<http://cl01s560.ltvintra.ltv.ch:7004/bsicrmserver/rest/v1/doc/>

aufgelistet. Zu jedem Webservice ist zu den verfügbaren Methoden (GET, POST, DELETE) eine Dokumentation verlinkt.

Die Dokumentation enthält folgende Punkte:

* Description: Allgemeine Beschreibung darüber, wozu der Webservice dient.
* URI Parameters: Feldname, Datentyp zu notwendigen oder optionalen Parametern, welche im URI übergeben werden
* Query Parameters: Feldname, Datentyp und mögliche Werte (bei Listen, Enums)
* Security Schemes: Authorisierungs-Protokoll (hier immer OAuth .20), Angabe von Accesstoken- und Authorisierungs-URI = https://auth.local.ch
* Response: Beispiel eines Abfrage-Ergebnisses

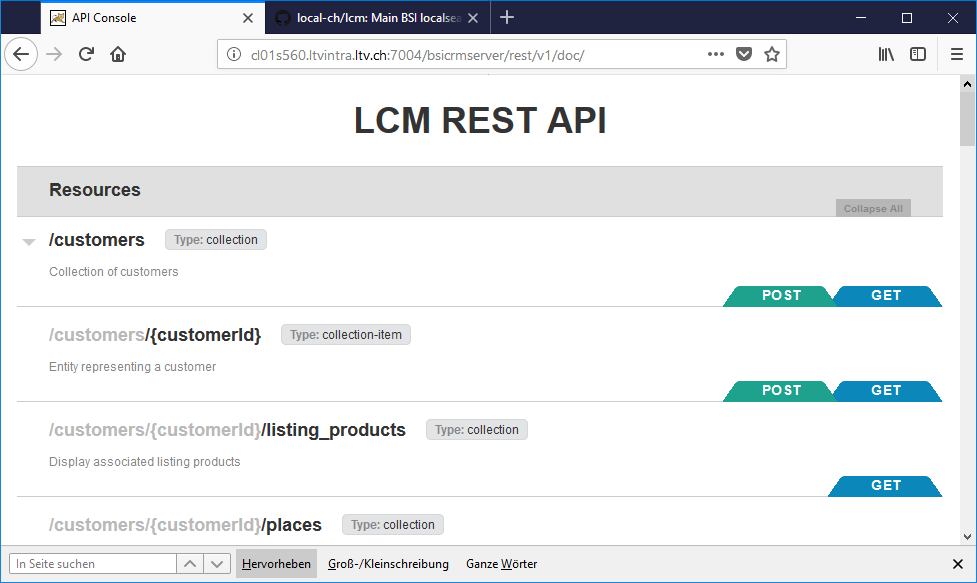


Abb. Dokumentation der verfügbaren Webservices

**Beispiel:** Unten ist beispielhaft die Dokumentation eines Webservice widergegeben.

### /customers/{customerId}/places

**Type:** collection **Traits:** hasInternalServerError, hasUnauthorizedError

Display associated places

GET

### Request

### Description

Get a list of associated places

### URI Parameters

#### customerIdrequired, string

### Query Parameters

#### langone of (de, fr, it, en)

translate multilanguage attributes in given language

#### limitinteger, default: 20

Limit the number of elements on the response

**Example:** 80

#### offsetinteger

Skip over a number of elements by specifying an offset value for the query

**Example:** 20

### Security Schemes

1. OAuth 2.0

#### Headers

#### Authorizationstring

Bearer token is needed

#### Settings

#### accessTokenUri

https://auth.local.ch

#### authorizationUri

https://auth.local.ch

### Response

###    Status 200

#### Body application/json

Example:

{

"href": "http://127.0.0.1:8080/bsicrmserver/rest/v1/customers/16773896/places",

"offset": 0,

"limit": 50,

"items": [

{

"href": "http://storage.internal-local.ch/v2/places/GMv21tkh3U4pTYa3u71WXw"

}

]

}

# Authorisierung bei Webservices mit oAuth2

Soll ein Webservice aus einer (Web-) Anwendung aufgerufen werden, muss die Anmeldung an den Authorisierungs-Server vom Programm gesteuert werden. So können z.B. mit Programmstart alle Berechtigungen des angemeldeten Users bereits in einer Session gespeichert werden.

Zum Test während der Entwicklung ist eine manuelle Authorisierung praktikabler; diese soll hier vorgestellt werden.

## Tokenizer

Zur Authentifizierung für einen Webservice ist ein Token erforderlich. Unter dem Link

<https://intranet.localsearch.ch/pages/viewpage.action?spaceKey=PP&title=EG+10+Core+System+Environments>

sind 5 Tokenizer URLs für 5 verschiedene Environments verfügbar. Für das Development sollte Environment = STG verwendet werden.

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

Abb. Die untere Tabelle enthält URL’s zu den Tokenizern für die verschiedenen Environments

Für Tests und Development sollte die Tokenizer URL unter STG verwendet werden:

[https://tokenizer.preprod-local.ch](https://tokenizer.preprod-local.ch/)

Damit öffnet sich das unten abgebildete Fenster. Unter «Environment» ist für Testst «Staging» auszuwählen.

**WICHTIG**

«Grant type» ist aktuell immer «client\_credentials» und NICHT «authorization\_code».

Mit «client\_credentials» werden Client-ID und Client-Secret übergeben – das Client-Secret wird von Tokenizer selbstständig eingefügt.

Mit «authorization\_code» werden zusätzlich Benutzername und Passwort verlangt – Eingabefelder dazu gibt es aber nicht. Wird «OK» gedrückt, erhält man die unten abgebildete Fehlermeldung. Diese erscheint erneut, wenn man den Tokenizer anschliessend wieder aufruft. In diesem Fall muss der Browsercache einmal gelöscht werden.

Einige Client ID’s benötigen immer Grant type = «authorization\_code», z.B. «salesforce». Diese sind hier nicht verwendbar.

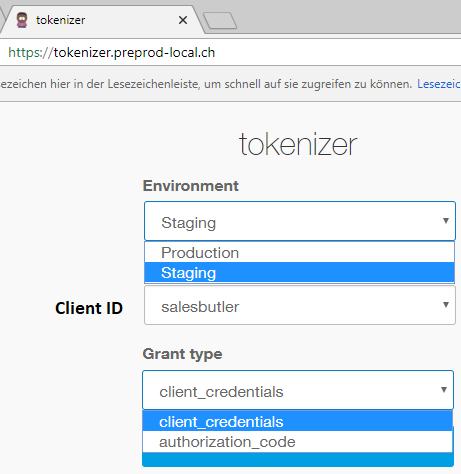


Abb. Tokenizer: Unter «Client ID» ist eine Liste von Client-IDs zugänglich.



Abb. Fehler nach Aufruf eines Token mit Grant type = «authorization\_code»

## Aufruf eines Webservice mit Authorisierung

Mit der Endpoint-URL <http://storage-stg.preprod-local.ch> soll die Resource /v2/places abgefragt werden.

Zunächst wird ein Token wie oben beschrieben besorgt; unten ist es eines für Client ID = «salesbutler», welche umfassende Berechtigungen besitzt und deshalb als eine Art Generalschlüssel verwendet werden kann.

Das Access Token wird nach Klick auf «OK» ausgegeben und kann in den Zwischenspeicher kopiert werden.

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

Abb. Ausgabe eines Access token für Client ID = «salesbutler»

Im Beispiel wird der Webservice mit URL <http://storage-stg.preprod-local.ch/v2/places> aufgerufen

### Mit SoapUI

Der blaue Button «Endpoint Explorer» öffnet den unten abgebildeten Dialog. Die Authorisierung erfolgt im Register «Authentication & Headers».

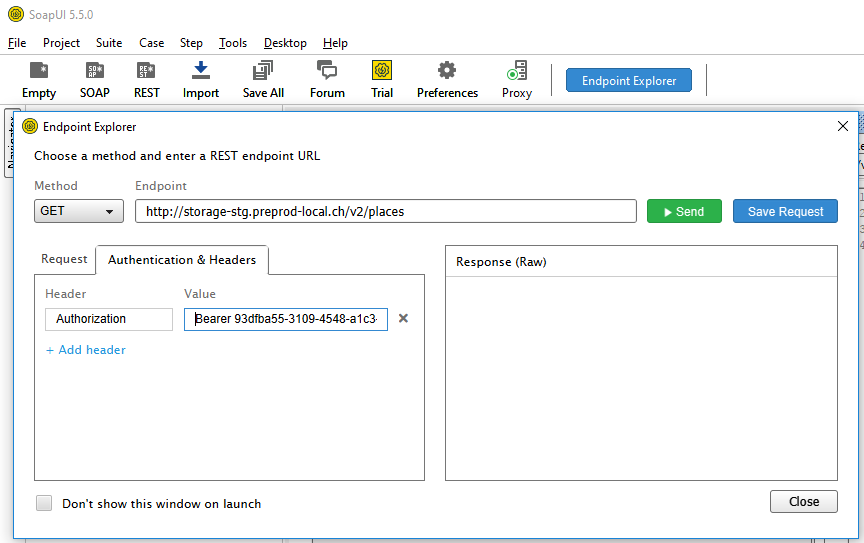


Abb. SoapUI: Aufruf eines Webservice mit Authorisierung

Als «Header» wird dort «Authorization» eingegeben. «Value» enthält das Token und zusätzlich den Token-Type; in diesem Fall «Bearer»

Unter «Method» kann eine der in der Dokumentation zu diesem Webservice angegebenen Methoden ausgewählt werden – in diesem Fall ist es GET.

Mit Klick auf «Send» wird das Ergebnis in das Respose Fenster ausgegeben.

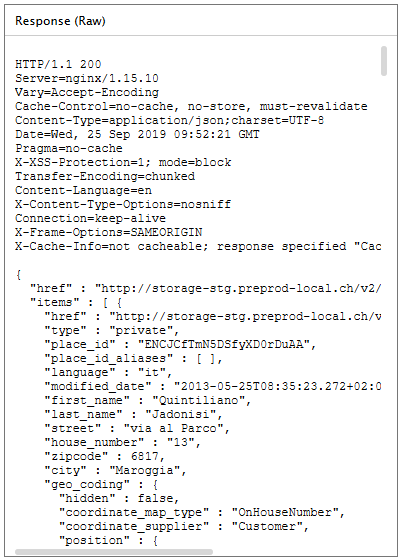


Abb. Rückgabe des Webservice <http://storage-stg.preprod-local.ch/v2/places> in SoapUI (Ausschnitt)

Rest API:

\* Property "rest.provider.credentials" entfernen

\* Neues Property "oauth.clientId" hat auf allen Umgebungen der Wert "lcm"

\* Neues Property "oauth.clientSecret"

- Wert für TEP "TJFVJUjzc92uoOLUEejKRtGI"

- Wert für Prod "9q5KrHgSC4yo5XZdrAsYi155"

\* Neues Property "oauth.authEndpoint"

- Wert für TEP "https://auth.staging.local.ch/oauth/check\_token"

- Wert für Prod "https://auth.local.ch/oauth/check\_token"

example properties file:

\* nicht verwendete imap properties wurden entfernt