

API Dokumentation

Die REST-API des Geolocation-Portals basiert auf [OData](#) (Open Data Protocol). Über diese API soll es anderen Programmen und Einrichtungen, der Stadt Mosbach, einen Zugriff auf die Daten ermöglicht werden. Dadurch können vollautomatische Datensätze im Geolocation-Portal angelegt werden.

Funktionen der API

GET

Aufruf von Datensätze über die ID. Wenn keine ID angegeben wurde, werden alle Datensätze zurückgegeben.

Die ID wird dabei in Klammern an die URL angehängt.

Bsp.:

[/api/records\(7\)](#)

POST

Um per POST Datensätze hinzuzufügen, müssen benötigte Parameter im JSON Format über den Request-Body an die API übergeben werden. Wichtig ist, dass der Content-Type der POST Anfrage dementsprechend auf „application/json“ eingestellt wurde.

Die ID darf hierbei nicht übergeben werden, da diese automatisch von der Datenbank erzeugt wird.

Bsp. (HTTP):

```
POST /api/records HTTP/1.1
Host: geolocation-portal.ddnss.de
Content-Type: application/json
{
  "dataset_upload": "2018-06-23T07:30:20",
  "dataset_modified_date": "2018-06-23T07:30:20",
  "title": "Kindergartenübersicht",
  "description": "Die Gemeinde Mosbach besitzt in vielen Ortschaften einen Kindergarten. Dieser Geolocation Datensatz zeigt die Position jedes Kindergartens innerhalb der Gemeinde Mosbach.",
  "category_id": 3,
  "licence_id": 1,
  "publisher_id": 1,
  "rating": 0,
  "dia_data": false,
  "geo_data": true,
  "role_id": 4,
  "record_active": true,
  "location_id": 1
}
```

PATCH

Bestehende Datensätze können über PATCH geändert werden. Die Auswahl des Datensatzes geschieht über die Angabe der ID in Klammern in der URL.

PATCH überschreibt den vorhandenen Datensatz, daher müssen alle notwendigen Parameter, wie bei POST, im Request-Body im JSON-Format angegeben werden.

DELETE

Per DELETE lassen sich Datensätze löschen. Die ID muss in Klammern an die URL angehängt werden.

Vor dem Löschen müssen eventuelle Verweise auf andere Objekte gelöscht werden.

Verkettung

Mit OData ist eine einfache Verkettung von Datentypen der Abfragen möglich. So lassen sich bestimmte Informationen direkt Abrufen. Der verkettete Datentyp ist hierbei immer im Singular.

Bsp.:

[/api/records\(7\)/comment](#)

Gibt von dem record, mit der ID 7, alle Kommentare zurück

Liste der Ressourcen

Die API bietet Zugriff auf folgende Datensätze:

Name	URL
record	/api/records
category	/api/categories
file	/api/files
location	/api/locations
publisher	/api/publishers
licence	/api/licences
comment	/api/comments
user	/api/users

Welche Attribute und Parameter für die einzelnen Daten benötigt werden, wird in den jeweiligen Tabellen oder unter [/api/\\$metadata](/api/$metadata) angezeigt.

Records

name	type	nullable
Id	Edm.Int32	false
dataset_upload	Edm.DateTime	false
dataset_modified_date	Edm.DateTime	false
title	Edm.String	false
description	Edm.String	
category_id	Edm.Int32	
licence_id	Edm.Int32	
publisher_id	Edm.Int32	
rating	Edm.Int32	false
dia_data	Edm.Boolean	false
geo_data	Edm.Boolean	false

Categories

name	type	nullable
Id	Edm.Int32	false
parent_id	Edm.Int32	false
name	Edm.String	
description	Edm.String	
icon	Edm.String	

Files

name	type	nullable
Id	Edm.Int32	false
record_id	Edm.Int32	false
file_upload_date	Edm.DateTime	false
download_count	Edm.Int32	
file_icon	Edm.String	
file_size	Edm.Double	
name	Edm.String	

Locations

name	type	nullable
Id	Edm.Int32	false
name	Edm.String	

Publishers

name	type	nullable
Id	Edm.Int32	false
name	Edm.String	
description	Edm.String	

Licences

name	type	nullable
Id	Edm.Int32	false
name	Edm.String	
description	Edm.String	

Comments

name	type	nullable
Id	Edm.Int32	false
title	Edm.String	
text	Edm.String	
person_name	Edm.String	
bewertung	Edm.Int32	false
record_id	Edm.Int32	false

Users

name	type	nullable
Id	Edm.Int32	false
role_id	Edm.Int32	false
department_id	Edm.Int32	false
first_name	Edm.String	
last_name	Edm.String	
username	Edm.String	
password	Edm.String	
last_password_change	Edm.DateTime	
create_date	Edm.DateTime	false
account_active	Edm.Boolean	false
login_attempts	Edm.Int32	
last_login	Edm.DateTime	