

AKTENNOTIZ

Hackathon 2025: FAQ Data

1. Rules

- Limit service requests, api access (e.g. no buildup of a "shadow" database)

2. Metadata of Geodata in geoportal.lu.ch (json)

Geopard is the metadata platform of geoportal.lu.ch. It contains all metadata of all geodata available. We provide you with a json export of geopard with the metadata of all publicly available data of the geoportal.

2.1. Geopard Object model

Here you see the structure of the json. You can open the products_ktl.json file in e.g.  Firefox and you see the structure of the json as well.

products_ktl.json (The Root Geodata Catalog)

Collection 	← "Kollektion" (Thematic grouping of datasets, e.g., "Amtliche Vermessung")
metauid, uuid, title, purpose	←  Core Identifiers. `metauid` is the key for lookup. `purpose` = contains description
timestamp, revision	← Metadata creation and revision dates.
keywords[]	←  Essential for Search. Thematic terms.
resource_maintenances[]	←  KEY: Contains `dateoflastupdate` for the Collection.
contact_gis, contact_spec, contact_owner	← Defines responsible parties.
resourceconstraints[]	← Defines usage rights and access restrictions (e.g., "A: öffentlich").
urls	←  KEY Human Access Points.
openly: { "url": "..." }	←  Metadata URL (Human-readable factsheet).
webapp: { "url": "..." }	←  Geodatashop URL (Product page).
datasets[]	← Contains individual Dataset objects/IDs.
services[]	← Contains Service objects/IDs for the collection.

— Dataset 📄	← A single data product (vector, raster, etc., e.g., "Wasserpflanzen")
— metauid, uuid, title, abstract	← 🔍 Core Identifiers. `abstract` = detailed description.
— purpose	← The intended use/goal of the dataset.
— keywords[]	← 🔑 More specific search terms for the dataset.
— resource_maintenances[]	← 📅 KEY: Contains `dateoflastupdate` for this dataset.
— data_type, feature_type	← Defines data type (e.g., raster, vector, point, polygon).
— extents[]	← Geographic bounding box information.
— contact_gis, contact_spec, contact_owner	← Defines dataset-specific contacts.
— resourceconstraints[]	← Defines dataset-specific usage rights.
— distributor	← Describes how to obtain raw data files.
— formats[]: "shp", "gpkg"...	← 📁 Available formats (plus media like "zip").
— citations[]	← External references or legal basis for the data.
— urls	← 🌐 KEY Human Access Points.
— openly: { "url": "..." }	← 📄 Metadata URL (Human-readable factsheet).
— webapp: { "url": "..." }	← 🛒 Geodatashop URL (Product page).
— services[]	← Service objects/IDs providing access to this data.
— Service 🌐	← An OGC/ESRI API endpoint exposing geodata to applications
— metauid, uuid, title	← 🔍 Core Identifiers.
— service_type	← Critical: "WMS", "WFS", "MapServer", "Feature-Service", etc.
— data_type (Dienst)	← General service type (e.g., "service").
— contact_gis	← Technical contact for the service.
— resourceconstraints[]	← Service-level usage constraints.
— urls (openly)	← Metadata link specific to the service (synced with dataset/ collection).
— elements[] > resources[]	← 🌐 CRITICAL API ENDPOINTS array.
— { "format": "WMSServer", "path": "..." }	← 🖼️ WMS URL (map images/tiles).
— { "format": "WFSServer", "path": "..." }	← 📊 WFS URL (raw vector data queries).
— { "format": "MapServer", "path": "..." }	← ArcGIS ESRI REST API URL.

2.2. Attributes needed to find a dataset or collection 🔍

To build an effective search, your chatbot must focus on these key fields within each dataset object. They provide the context needed to match a user's query to the right data.

metauid

A unique identifier for the metadata entry (e.g., WRZONXXX_COL).

It's the primary key for any dataset or service. When your chatbot finds a relevant result, it should use the metauid to reliably retrieve that exact object again for more details. It prevents ambiguity.

title

The human-readable name of the dataset or service (e.g., "Strassenlärmbelastungskastaster").

This is the most important field for keyword matching. Your chatbot should search this field first when a user asks for a specific dataset.

purpose / abstract

A short description explaining what the data represents and its intended use. This provides context that goes beyond the title. A user might ask, "I need data that shows where it is loud." The purpose or abstract will contain keywords like "noise" or "emissions" that help your chatbot confirm it has found the correct dataset.

keywords

An array of search terms associated with the dataset (e.g., ["Lärm", "Verkehr", "Strasse"]).

3. Webmaps

1.1. What webmaps are there?

Overview of webmaps via: <https://map.geo.lu.ch/feed.xml>

1.2. Zoom to coordinates

In webmaps you can zoom to webmaps via url parameters: <https://map.geo.lu.ch/oberflaechengewaesser/netz?FOCUS=2649576:1211796:4515&marker>

"& marker" adds a marker pin at the given location

4. Translate location to coordinates:

4.1. Location Service API

- Requests can be made for: EGRID (ID of parcels), Parcel number, address (street name + number + PLZ), name of municipality, Ortsname, Flurname, Gebäudeversicherungsnummer (insurance number), EGID (ID of buildings)
- Examples of a search for "mur" to get results for Murbacherstrasse...
<https://svc.geo.lu.ch/locationfinder/api/v1/lookup?query=mur&limit=100&filter=type%3AAdresse>
<https://svc.geo.lu.ch/locationfinder/api/v1/lookup?query=mur&limit=100&filter=type%3AGrundst%C3%BCck>
<https://svc.geo.lu.ch/locationfinder/api/v1/lookup?query=mur&limit=100&filter=type%3AGemeinde%2COrt%2CFlurname>
<https://svc.geo.lu.ch/locationfinder/api/v1/lookup?query=mur&limit=100&filter=type%3Agebaudeversicherungsnummer>
<https://svc.geo.lu.ch/locationfinder/api/v1/lookup?query=mur&limit=100&filter=type%3AEGRID>
<https://svc.geo.lu.ch/locationfinder/api/v1/lookup?query=mur&limit=100&filter=type%3AEGID>

- Example answer from the service:

```
{
  "id": 602788,
  "type": "Adresse",
  "name": "Mürgi 1.1, 6025 Neudorf",
  "cx": 2658045, <----- center coordinates
  "cy": 1225376,
  "xmin": 2657995, <----- extent coordinates
  "ymin": 1225326,
  "xmax": 2658095,
  "ymax": 1225426,
  "fields": {          <----- other location info
    "plz": "6025",
    "str": "Mürgi",
    "hnr": "1.1",
    "ort": "Neudorf",
    "info": "AVEINVXX_V2_PT",
    "id": "AVEINVXX_V2_PT",
    "code": "ADR_502290823-0",
    "egid": "502290823",
    "edid": "0",
    "gem": "Beromünster",
    "bfsnr": "1081"
  }
}
```

- Full docu (no access): [Wiki - LocationFinder - GEO Projektverwaltung](#)
- Open Street Map (OSM) location finder: [Nominatim](#)

4.2. Datasets for location

POI Datasets for searches like Bahnhof Luzern...

- Öffentlicher Verkehr: Haltestellen (öffentlich) - [Öffentlicher Verkehr: Haltestellen \(öffentlich\) \(OEVHSTXX_DS\) - Metadaten Kanton Luzern](#)

5. How to access geodata via webservice? □

Services are the **APIs** (Application Programming Interfaces) for geodata. They allow you to fetch map tiles or raw data directly over the web. You identify the service type you need using the **service_type** field and the **format** field inside the resources[] array:

Web Map Service (WMS)

With a Web Map Service (WMS), a combination of map layers is requested for a specific area. The WMS server generates the requested map and delivers it as a **raster image**. The generated raster image can be directly integrated into a desktop GIS (e.g., ArcGIS, QGIS, AutoCAD) or a suitable web application.

Web Map Services are useful when constantly up-to-date geodata in a predefined visualization is required.

Example for WMS GetFeatureInfo-Request (dataset: Erdwärmenutzung: Zulässigkeit):

https://public.geo.lu.ch/ogd/services/managed/EWNUTZXX_COL_V3_MP/MapServer/WMSServer?service=wms&version=1.3.0&request=getfeatureinfo&format=image%2fpng&transparent=true&query_layers=3&layers=3&feature_count=10&info_format=text%2fplain&lang=de&i=50&j=50&crs=epsg%3a2056&styles=&width=101&height=101&bbox=2667573.7499999995%2c1210072.5%2c2667826.2499999995%2c1210325

Docu Canton Lucerne: [Geodienste einbinden - Kanton Luzern](#) => Mehr Informationen zu Web Map Services (WMS) einblenden (click on arrow).

Web Feature Service (WFS)

With a Web Feature Service (WFS), access to **vector data** from a predefined set of map layers is possible.

Data obtained via WFS is delivered without a predefined visualization.

The transmitted vector data can be directly integrated into suitable GIS applications (e.g., ArcGIS, QGIS, AutoCAD) and analyzed.

Docu Canton Lucerne: [Geodienste einbinden - Kanton Luzern](#) => Mehr Informationen zu Web Feature Services (WFS) einblenden (click on arrow).

ESRI Rest API

The ESRI REST API is a specific type of web service built by ESRI (the company behind ArcGIS) that uses standard web requests (HTTP) to deliver geographic information, making it easy for web apps to request maps and data. You can access data using structured URLs that return JSON.

Example on Wildruhezonen:

In Collection find the Service: [managed/WRZONXXX_COL_V1_MP \(MapServer\)](#)

Find out the embedded layers in the response. We search here for Wildruhezonen which is a dataset and here layer 2.

This results in full query at specific coordinates.

public.geo.lu.ch/ogd/rest/services/managed/WRZONXXX_COL_V1_MP/MapServer/2/query?geometry=2649672.80,1207836.67&geometryType=esriGeometry-Point&inSR=2056&spatialRel=esriSpatialRelIntersects&f=json

Docu by Esri: [Build powerful apps with ArcGIS services | Esri Developer REST APIs documentation | Esri Developer](#)