

Web-APIs

für geographische Anwendungen

Jakob Miksch

meggsimum - Büro für Geoinformatik

FOSSGIS 2022, 09.03.2022

Jakob Miksch



- Geoinformatiker
- Anwendungsentwickler
- OSGeo Foundation Charter Member

✉ jakob@meggsum.de

⌚ [@jakobmiksch](https://www.twitter.com/jakobmiksch)

🐦 [@jakobmiksch](https://www.twitter.com/jakobmiksch)

meggsimum



✉ info@meggsimum.de
⌚ [@meggsimum](https://www.twitter.com/meggsimum)
🐦 [@meggsimum](https://www.twitter.com/meggsimum)

- meggsimum.de
- Dienstleistungen im Bereich GIS, Webmapping & GDI
- Maßgeschneiderte WebGIS-Lösungen
- Softwarekonzepte und Softwareentwicklung
- Geodaten
- Beratung und Schulung

(Web) APIs

- API = Application Programming Interface
- allgegenwärtig im Internet
- Datenaustausch
- Softwareentwicklung

Beispiel

OpenWeatherMap

The screenshot shows the OpenWeatherMap API documentation homepage. At the top, there is a navigation bar with links for Guide, API, Pricing, Maps, Our Initiatives, Partners, Blog, Marketplace, Sign in, and Support. The main content area is divided into several sections, each with a title, a brief description, and two buttons: 'API doc' and 'Subscribe'.

- Current Weather Data**
 - Access current weather data for any location including over 200,000 cities
 - We collect and process weather data from different sources such as global and local weather models, satellites, radars and a vast network of weather stations
 - JSON, XML, and HTML formats
 - Included in both free and paid subscriptions
- Hourly Forecast 4 days**
 - Hourly forecast is available for 4 days
 - Forecast weather data for 96 timestamps
 - JSON and XML formats
 - Included in the Developer, Professional and Enterprise subscription plans
- One Call API**
 - Make one API call and get current, forecast and historical weather data
 - Minute forecast** for 1 hour
 - Hourly forecast** for 48 hours
 - Daily forecast** for 7 days
 - Historical data** for 5 previous days
 - National weather alerts**
 - JSON format
 - Included in both free and paid subscriptions
- Daily Forecast 16 days**
 - 16 days forecast is available for any location or city
 - 1-day step for 16 days
 - JSON and XML formats
 - Included in all paid subscription plans
- Climatic Forecast 30 days**
 - Forecast weather data for 30 days
 - JSON format
 - Included in the Developer, Professional and Enterprise subscription plans
- Bulk Downloading**
 - You may request current weather and forecasts in bulk with a variable data granulation
 - Current weather bulk is available for 209,000+ cities
 - Variety of hourly and daily forecast bulks depends on the frequency of data updating
 - Additionally, this product allows to get archived current and forecasts weather data for 7 previous days
 - Included in the Professional and Enterprise subscription plans

<https://openweathermap.org/api>

Beispiel

Google Street View

Google Maps Platform

Overview Products Pricing Documentation ▾ Blog Community ▾

Search English Sign in

Introduction

The Street View Static API lets you embed a static (non-interactive) Street View panorama or thumbnail into your web page, without the use of JavaScript. The viewport is defined with URL parameters sent through a standard HTTP request, and is returned as a static image.



https://maps.googleapis.com/maps/api/streetview?size=400x400&location=47.5763831,-122.4211769&fov=80&heading=70&pitch=0&key=YOUR_API_KEY&signature=YOUR_SIGNATURE

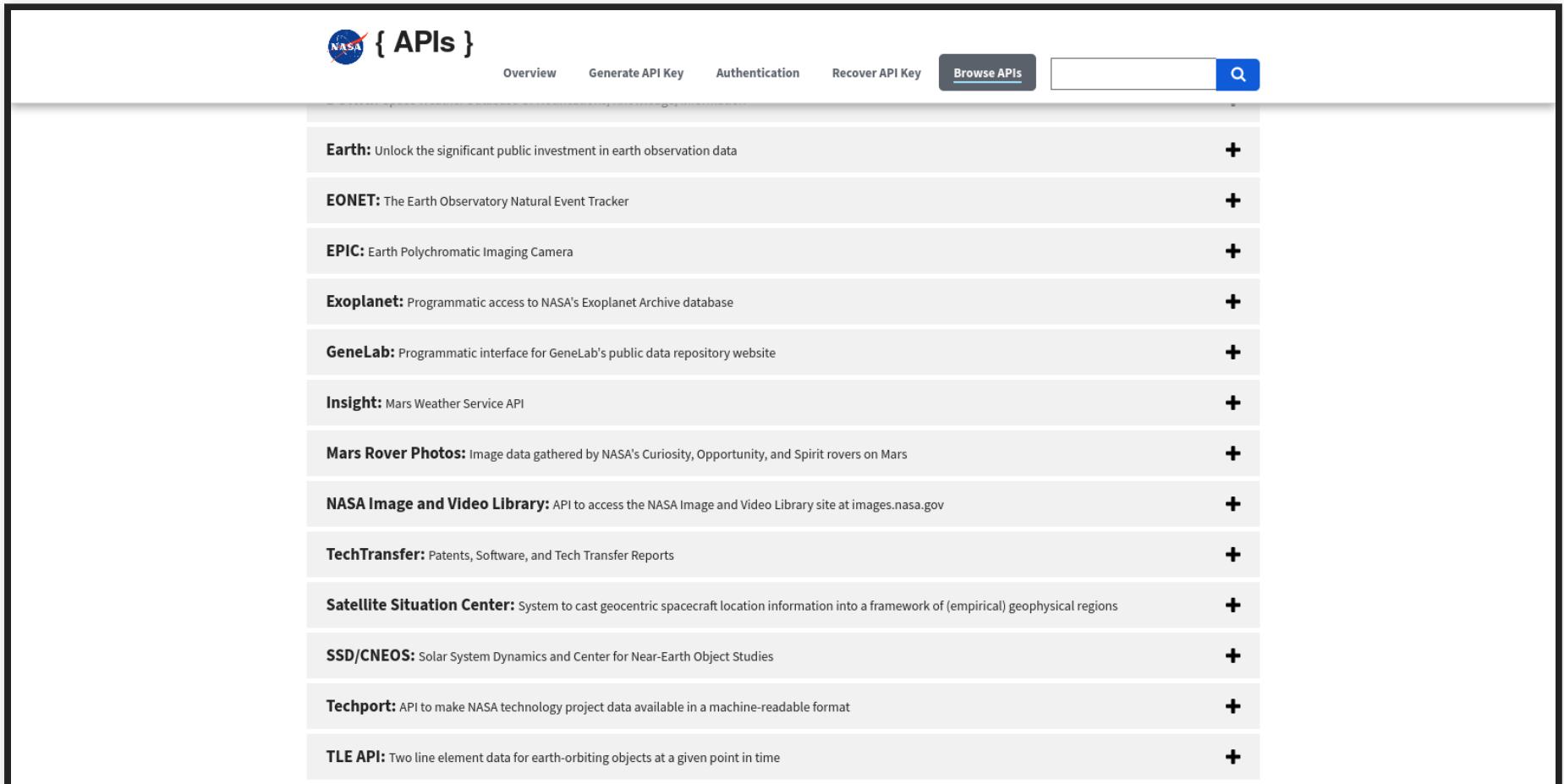
On this page

- Introduction
- Before you begin
- URL parameters
- Required parameters
- Optional parameters
- No imagery available
- Show only valid images
- More information

<https://developers.google.com/maps/documentation/streetview/overview>

Beispiel

NASA



The screenshot shows the NASA APIs website interface. At the top, there is a navigation bar with the NASA logo, a search bar, and several menu items: Overview, Generate API Key, Authentication, Recover API Key, and a highlighted 'Browse APIs' button. Below the navigation bar is a list of API entries, each with a plus sign icon on the right side.

API Name	Description	Action
Earth:	Unlock the significant public investment in earth observation data	+
EONET:	The Earth Observatory Natural Event Tracker	+
EPIC:	Earth Polychromatic Imaging Camera	+
Exoplanet:	Programmatic access to NASA's Exoplanet Archive database	+
GeneLab:	Programmatic interface for GeneLab's public data repository website	+
Insight:	Mars Weather Service API	+
Mars Rover Photos:	Image data gathered by NASA's Curiosity, Opportunity, and Spirit rovers on Mars	+
NASA Image and Video Library:	API to access the NASA Image and Video Library site at images.nasa.gov	+
TechTransfer:	Patents, Software, and Tech Transfer Reports	+
Satellite Situation Center:	System to cast geocentric spacecraft location information into a framework of (empirical) geophysical regions	+
SSD/CNEOS:	Solar System Dynamics and Center for Near-Earth Object Studies	+
Techport:	API to make NASA technology project data available in a machine-readable format	+
TLE API:	Two line element data for earth-orbiting objects at a given point in time	+

<https://api.nasa.gov/index.html>

Austausch Formate

- XML
- JSON
- ...

XML

```
<breakfast_menu>
  <food>

    <name>Belgian Waffles</name>

    <price>$5.95</price>

    <description>
      Two of our famous Belgian Waffles with plenty of real
      maple syrup
    </description>

    <calories>650</calories>
  </food>
```

<https://www.w3schools.com/xml/simple.xml>

JSON

```
{  
  "squadName": "Super hero squad",  
  "homeTown": "Metro City",  
  "formed": 2016,  
  "secretBase": "Super tower",  
  "active": true,  
  "members": [  
    {  
  
    "name": "Molecule Man",  
  
    "age": 29,  
  
    "secretIdentity": "Dan Jukes",  
  ]}
```

<https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>

GeoJSON

```
{  
  "type": "Feature",  
  "geometry": {  
    "type": "Point",  
    "coordinates": [125.6, 10.1]  
  },  
  "properties": {  
    "name": "Dinagat Islands"  
  }  
}
```

<https://geojson.org/>

HTTP Methods

GET

POST

PUT

PATCH

DELETE

...

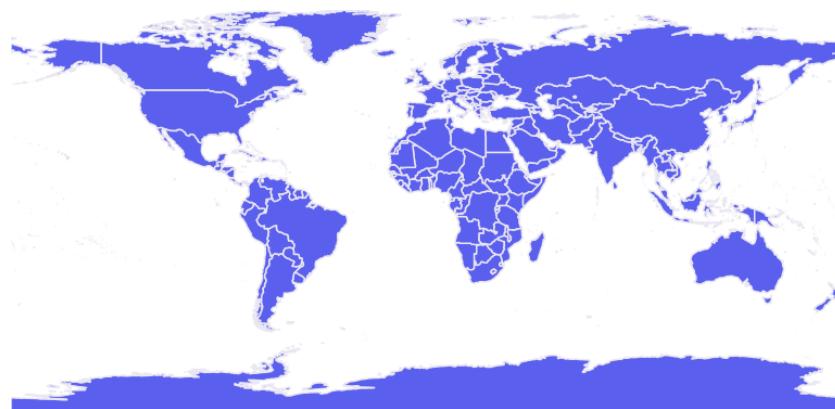
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

URL - Query Parameter

```
https://www.example.com?  
color=blue&  
type=dark
```

Beispiel WMS

```
https://services.meggsimum.de/geoserver/mgsm-world/wms?  
service=WMS&  
version=1.1.0&  
request=GetMap&  
layers=mgsm-world:ne_10m_admin_0_countries&  
bbox=-180.0, -90.0, 180.0, 83.6341&  
width=768&  
height=370&  
srs=EPSG:4326&  
format=image/png
```



API Typen

- SOAP
- REST
- GraphQL
- ...

OGC Standards

- WMS, WFS, WCS, ...
- Beispiel: **GeoServer (meggsimum)**
- Weiterentwicklung: RESTful OGC API

Authentifizierung

typischerweise mit API-key

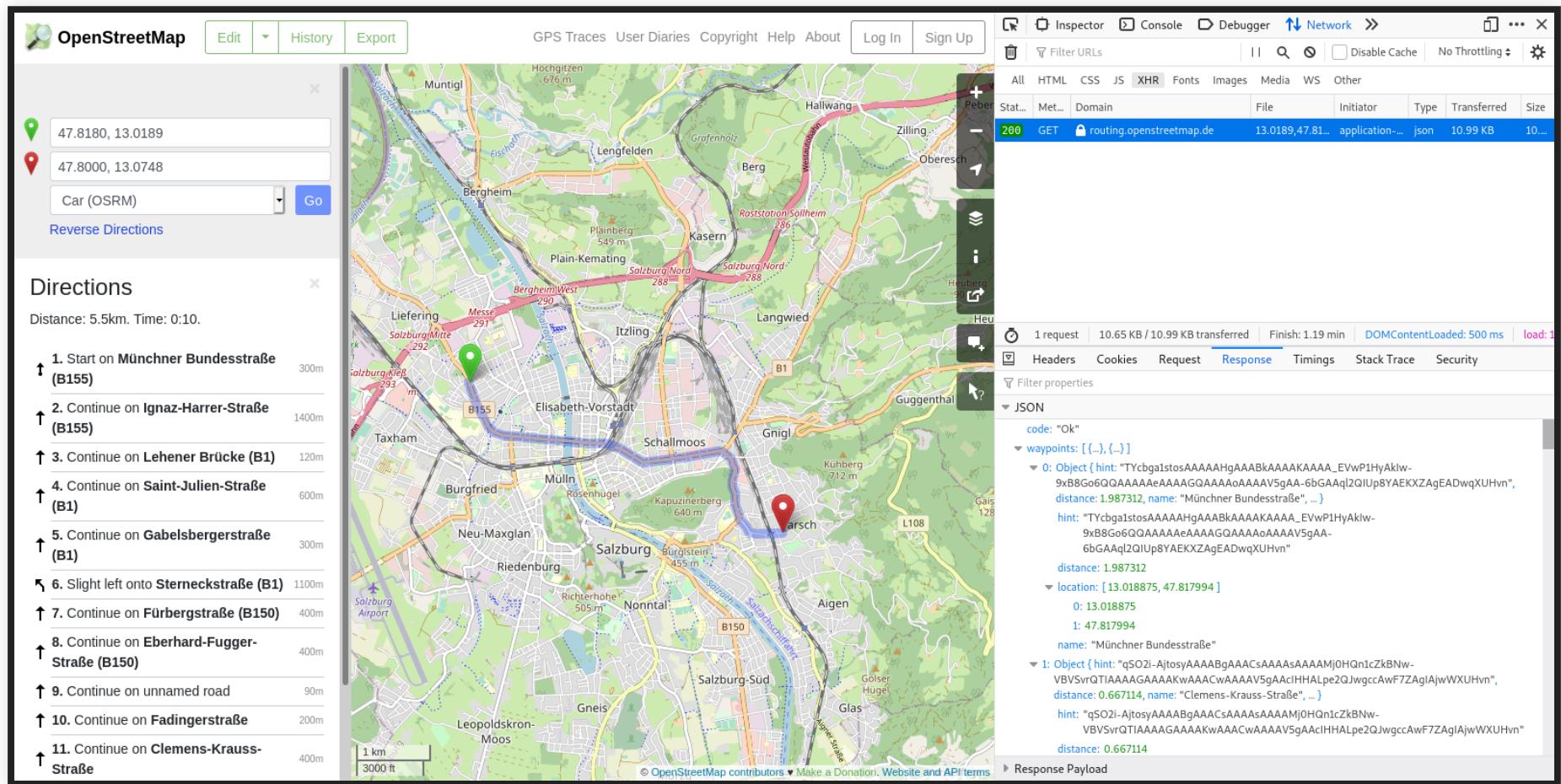
```
https://api.openrouteservice.org/v2/directions/driving-car?  
api_key=<--ADD_YOUR_API_KEY-->&  
start=8.681495,49.41461&  
end=8.687872,49.420318
```

Werkzeuge zum Zugriff

- Webbrowser
- Kommandozeile
- Programmiersprache
- Desktop Programme

Webbrowser

Entwicklerkonsole oder mit JavaScript



Kommandozeile

```
curl -X GET \  
'https://photon.komoot.io/api/?q=salzburg&limit=2'
```

ergibt:

```
{
  "features": [
    {
      "geometry": {
        "coordinates": [
          13.0464806,
          47.7981346
        ],
        "type": "Point"
      },
      "type": "Feature",
      "properties": {
        "osm_id": 86538.
```

Python

```
import requests

url = 'https://photon.komoot.io/api/?q=salzburg&limit=2'
response = requests.request('GET', url )

print(response)
```

R

```
library(httr)
library(jsonlite)

url = "https://photon.komoot.io/api/?q=salzburg"
res = GET(url)
data = fromJSON(rawToChar(res$content))

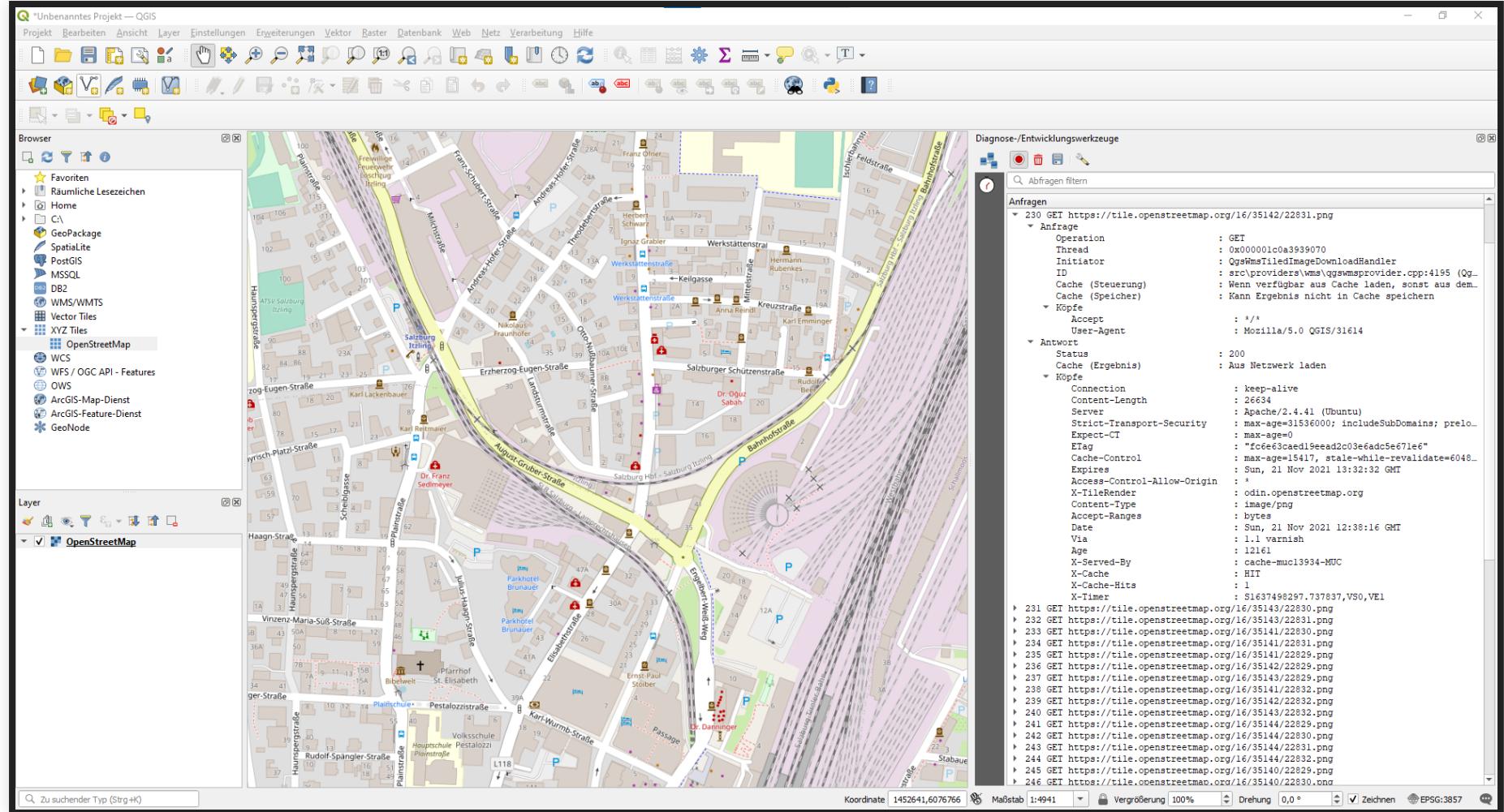
print(data)
```

JavaScript

```
const requestOptions = {  
    method: 'GET'  
};  
  
const url = "https://photon.komoot.io/api/?q=salzburg";  
fetch(url, requestOptions)  
    .then(response => response.text())  
    .then(result => console.log(result))  
    .catch(error => console.log('error', error));
```

QGIS

mit Python oder dem Debugging Panel



Desktop Programme

Hoppscotch oder Postman

The screenshot shows the Hoppscotch API client interface. On the left, there's a sidebar with icons for REST, GraphQL, Realtime, Docs, and Settings. The main area has tabs for REST, GraphQL, and Realtime. A search bar at the top right contains the URL `https://photon.komoot.io/api/`. Below it, a "Parameters" tab is selected, showing two entries: `q` with value `Schillerstraße 30, Salzburg` and `limit` with value `5`. There are also other rows for "Parameter 3" and "Value 3". To the right of the parameters are "Send" and "Save" buttons. The bottom section shows the response body in JSON format, listing coordinates, properties (osm_id, extent), and location details (country, city, countrycode, postcode) for a point in Salzburg, Austria.

```
30 "geometry": {  
31   "coordinates": [  
32     13.03942320885388,  
33     47.8232754  
34   ],  
35   "type": "Point"  
36 },  
37   "type": "Feature",  
38   "properties": {  
39     "osm_id": 11259047,  
40     "extent": [  
41       13.0389547,  
42       47.8238426,  
43       13.0401179,  
44       47.8227124  
45     ],  
46     "country": "Austria",  
47     "city": "Salzburg",  
48     "countrycode": "AT",  
49     "postcode": "5020",  
50   }  
51 }
```

GeoCoding APIs

- OpenCage
- openrouteservice
- Photon
- Nominatim
- ...

Routing APIs

- GraphHopper
- openrouteservice
- OSRM (FOSSGIS Instanz)
- Valhalla (FOSSGIS Instanz)
- ...

Overpass API

Datenabfrage von OpenStreetMap (Beispiel)

```
1 /*
2 This has been generated by the overpass-turbo wizard.
3 The original search was:
4 "museum"
5 */
6 [out:json][timeout:25];
7 // gather results
8 (
9   // query part for: "museum"
10  node["tourism"="museum"]({{bbox}});
11  way["tourism"="museum"]({{bbox}});
12  relation["tourism"="museum"]({{bbox}});
13 );
14 // print results
15 out body;
16 >;
17 out skel qt;
```

Map Data

Loaded – nodes: 210, ways: 13, relations: 1
Displayed – polis: 13, lines: 0, polygons: 12

Actinia API

Prozessierung von Geodaten mit GRASS GIS

OPERATIONS

- Authentication Management
- API Log
- Cache Management
- Satellite Image Algorithms
- Vegetation index comput...
- Query the Google Lands...
- Download and import La...
- Download and import Se...
- NDVI computation of an ...
- NDVI computation of an ...**
- Query the Google Sentin...
- Generate the download u...
- Location Management
- Mapset Management
- Processing
- Raster Management
- Raster Statistics
- STRDS Management
- STRDS Sampling
- STRDS Statistics
- Vector Management
- Resource Management

SCHEMA DEFINITIONS

- LocationListResponseModel
- SimpleResponseModel
- MapsetInfoResponseModel
- ProcessLogModel
- GrassModule

Satellite Image Algorithms

NDVI computation of an arbitrary Sentinel 2A scene.
The results are stored in the Google Cloud Storage.

PATH

POST /sentinel2_process_gcs/ndvi/{product_id}

DESCRIPTION

NDVI computation of an arbitrary Sentinel 2A scene. The processing is as follows: A user specific Sentinel 2A scene (Bands 04 and 08) will be download and imported into an ephemeral database.. The NDVI will be computed via r.mapcalc. The result of the computation is available as gzipped geotiff file. In addition, the univariate statistic will be computed as well as a preview image including a legend and scale. Minimum required user role: user.

REQUEST PARAMETERS

product_id	The product id of a sentinel scene
	<small>required in path string default:</small>
	S2A_MSIL1C_20170212T04141_N020 4_R008_T31TGJ_20170212T04138

RESPONSES

200 OK	This response includes all created resources as URL as well as the processing log and other metadata.
400 Bad Request	The error message and a detailed log why

Response Example (200 OK)

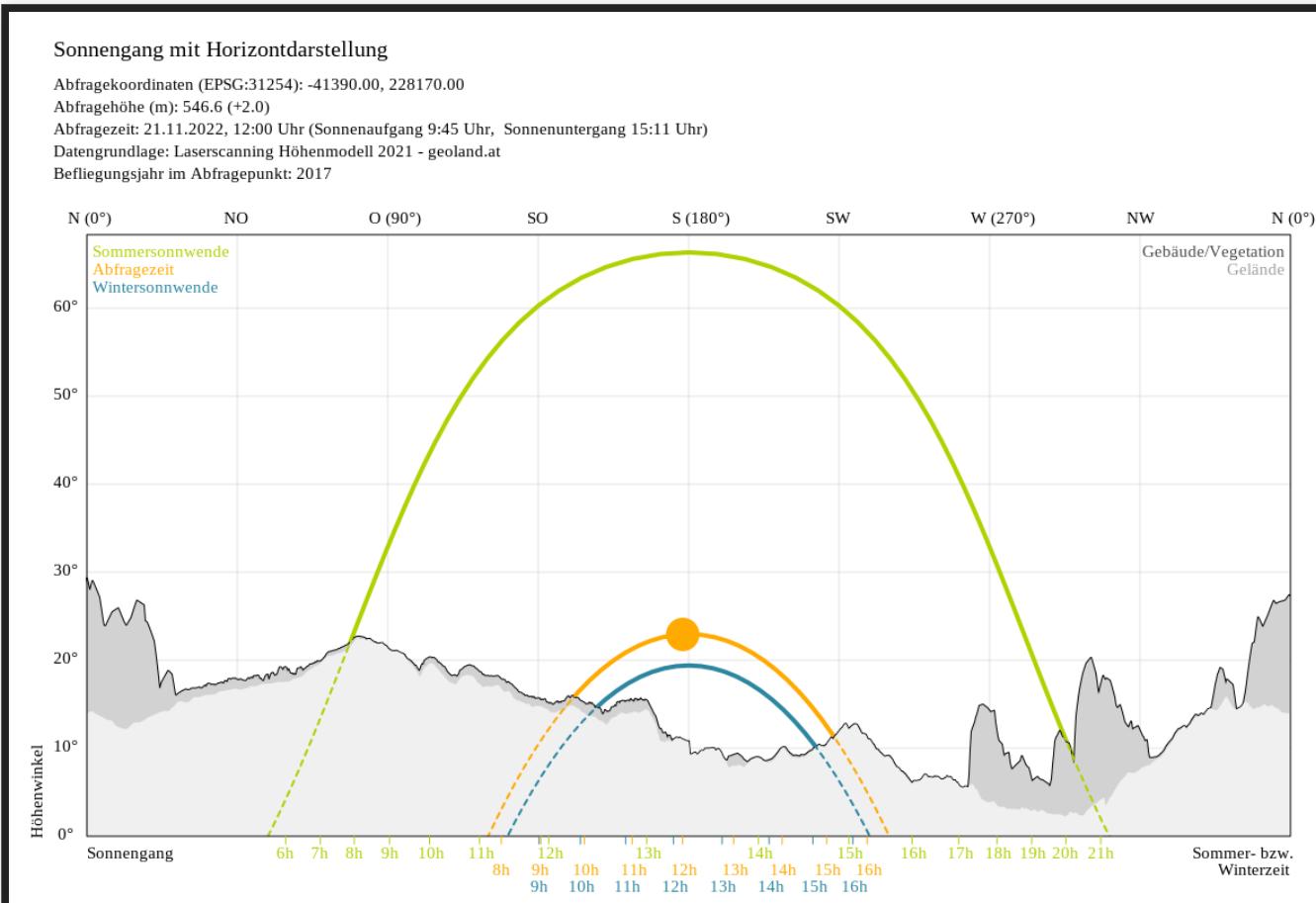
```
{  
  "accept_datetime": "2018-05-30 12:25:43.987713",  
  "accept_timestamp": 1527683143.9877105,  
  "api_info": {  
    "endpoint": "asyncephemeralsentinel2processingresource",  
    "method": "POST",  
    "path": "/api/v1/sentinel2_process_gcs/  
  }  
}
```

Sonnenstand

```
http://voibos.rechenraum.com/voibos/voibos?  
name=sonnengang&  
Koordinate=-41390, 228170&  
CRS=31254&  
Datum=11-21:12:34&  
H=2&  
Output=Horizont, Sonnenzeit
```

[Dokumentation](#)

Sonnenstand



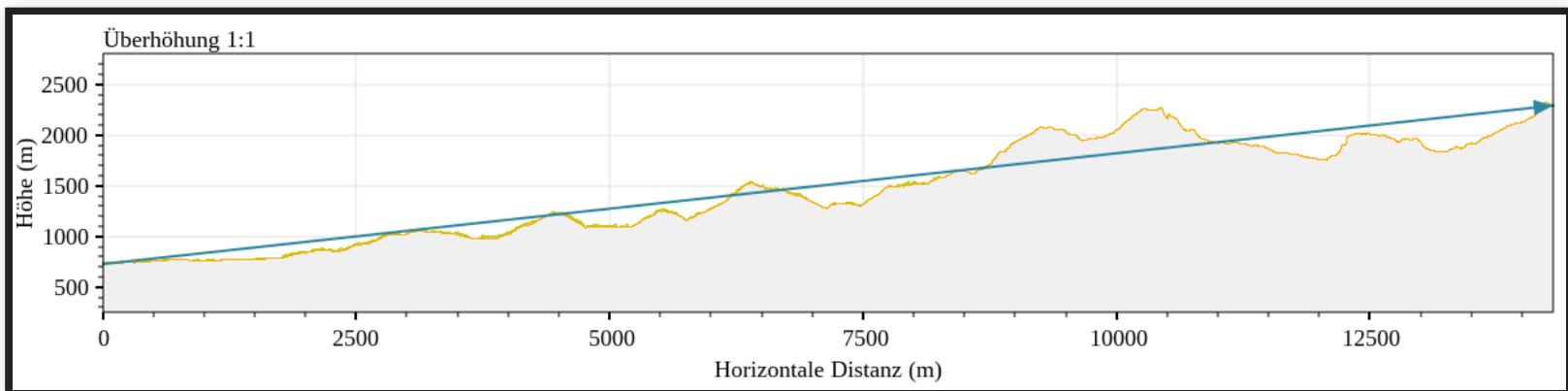
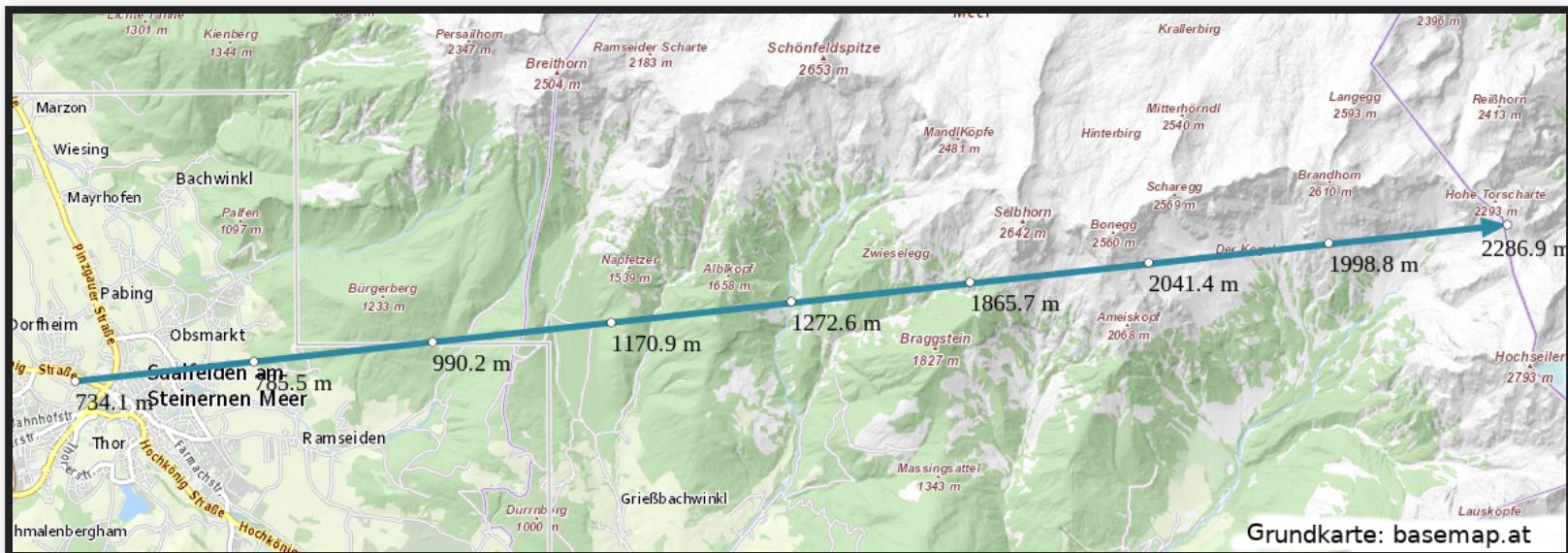
Dokumentation

Höhenprofil

```
https://voibos.rechenraum.com/voibos/voibos?  
name=profilservice&  
Beschriftung=Beispiel&  
Startkoordinate=12.83935,47.42960&  
Zielkoordinate=13.027871,47.443578&  
CRS=4326&  
Ueberhoehung=1&  
Stuetzpunktabstand=2&  
Hintergrund=geolandbasemap&  
Output=Profil,Lage
```

[Dokumentation](#)

Höhenprofil



Links

- Folien
- meggsumum GeoServer Instanz
- meggsumum Webseite
- Vortrag in FOSSGIS Pretalx
- Kontakt: jakob@meggsumum.de

Impressum

Autor

Jakob Miksch

meggsimum - Büro für Geoinformatik (Christian Mayer)

Schillerstraße 2a

67122 Mutterstadt

info@meggsimum.de



Lizenz

Diese Folien sind unter **CC BY-SA** veröffentlicht.

Vortragsfolien, PDF-Version, git repository