

geocat.ch technical training

-

machin-to-machine, API, automation

Benoit Regamey

<mailto:benoit.regamey@swisstopo.ch>

15.05.2023

What is geocat.ch

A modified instance of the open source projet geonetwork

<https://geonetwork-opensource.org/>

Currently running on version 4 (4.2.3)

Documentation : <https://geonetwork-opensource.org/manuals/4.0.x/en/index.html>



3 instances running :

Prod : <https://www.geocat.ch>

Int : <https://geocat-int.dev.bgdi.ch>

Dev : <https://geocat-dev.dev.bgdi.ch>

Metadata are stored in XML following the iso19139.che schema. This is an extended ISO19139 schema with swiss specific add-ons

What is geocat.ch – the home page

The screenshot shows the geocat.ch home page. At the top, a blue navigation bar contains the logo 'geocat.ch' (highlighted with a red box), a search icon, 'Map', 'Documentation', 'Sign in', and 'English'. The main header features the 'geocat.ch geographic catalogue' logo and a cartoon cat on a globe. Below this is a search bar (highlighted with a red box) containing the text 'Search ...' and a search icon. Underneath the search bar, it says 'Search 14559 data sets, services and maps, ...'. The main content area is titled 'Browse by' (highlighted with a red box) and includes three tabs: 'Topic' (selected), 'Catalogue', and 'Type of resources'. Below the tabs is a grid of 24 blue boxes, each representing a topic with a folder icon and a count. The topics are: E Spatial Planning, Cadastre (2008); L Environmental and Nature Protection (1511); A Base Maps, Land Cover, Aerial and Satellite Imagery (1447); Q Utilities, Supply, Disposal, Communication (1071); P Transportation (962); G Forest, Flora, Fauna (859); O Buildings, Infrastructure, Facilities (742); F Geology, Soils, Natural Hazards (717); I Inland Waters (659); D Political and Administrative Boundaries (501); M Population, Society, Culture (491); S Agriculture (449); B Location, Reference Systems (371); C Elevation (348); K Atmosphere, Climatology (231); T Economical Activities (158); N Health (142); R Military, Security (137); and H Oceans.

Topic	Count
E Spatial Planning, Cadastre	2008
L Environmental and Nature Protection	1511
A Base Maps, Land Cover, Aerial and Satellite Imagery	1447
Q Utilities, Supply, Disposal, Communication	1071
P Transportation	962
G Forest, Flora, Fauna	859
O Buildings, Infrastructure, Facilities	742
F Geology, Soils, Natural Hazards	717
I Inland Waters	659
D Political and Administrative Boundaries	501
M Population, Society, Culture	491
S Agriculture	449
B Location, Reference Systems	371
C Elevation	348
K Atmosphere, Climatology	231
T Economical Activities	158
N Health	142
R Military, Security	137
H Oceans	

Powered by GeoNetwork 4.2.3 SNAPSHOT About GitHub API Share and comment

What is geocat.ch – the search application

The screenshot displays the geocat.ch search application interface. At the top, a blue navigation bar contains the logo 'geocat.ch', a search icon, and links for 'Map' and 'Documentation'. A search bar is located in the center of the top bar. Below the search bar, a map on the left shows a geographical area with several orange boxes highlighting specific regions. To the right of the map is a 'Filter' panel with various options like 'Catalogue', 'Topic', 'Keywords', 'Spatial representation type', 'Formats', 'Scales', 'Resolution', and 'Update frequencies'. The main content area is a grid of search results, each with a title, a small image, a description, and the provider's name. The results include 'Cartographie des sols agricoles de la plaine du Rhône', 'Anciennes cartes nationales', 'Zivilschutzkataster', 'Wii (SG) - Strategie Verkehr', 'Réseau hydrographique', and 'Forêts protectrices'. The footer of the application contains the text 'Powered by GeoNetwork 4.2.3 SNAPSHOT' and various social media icons.


What is geocat.ch – the default view

geocat.ch Search Map Documentation Sign in English


Back to search Download Display mode

swissALTI3D multidirectional Hillshade

The hillshade is derived from the digital elevation model swissALTI3D. It illustrates the surface relief of Switzerland and the Principality of Liechtenstein. A hillshade is derived from the elevation model by calculating the exposure of the surface to sunlight at a certain or several angles. The swissALTI3D hillshade uses a multidirectional method that combines light from 6 different sources (sun's positions). The average sun's position is located to the northwest.







Spatial extent



- Representation type**
Grid
- Resolution**
0.5 m
- Coordinate reference system**
EPSG:21781, EPSG:21781, EPSG:5728

API

 WMS	ch.swisstopo.swissalti3d-reliefschattierung	Add to map	
 WMFS	ch.swisstopo.swissalti3d-reliefschattierung	Add to map	

WMFS: WMFS-BGDJ Dienst, Layer "swissALTI3D multidirektionales Relief"

WMS: WMS-BGDJ Dienst, Layer "swissALTI3D multidirektionales Relief"

Format: ASCII XYZ

Powered by GeoNetwork 4.2.3 SNAPSHOT About GitHub API

What is geocat.ch – the map viewer

The screenshot displays the geocat.ch web application interface. At the top, a blue navigation bar contains the logo, search, map, and documentation links, along with a sign-in button and a language dropdown set to English. Below the navigation bar is a search input field with the placeholder text "Search for a place".

The main content area features a 3D topographic relief map of a mountainous region in Switzerland, with a red boundary line indicating a specific area of interest. The map shows detailed terrain features, including rivers and mountain peaks. The label "néve" is visible on the map.

On the right side, a "Manage layers" panel is open, showing a list of layers. The first layer, "swissALTI3D multidirektionales Relief", is checked. Below this, a "Background map:" section lists several options: "ch swisstopo.pixelkarte-farbe (map.geo.admin.ch)", "OpenStreetMap", "Stamen Watercolor", "Stamen Toner", and "Stamen Terrain". A "Default map" button is located at the bottom of this panel.

At the bottom of the interface, a blue footer bar contains the text "Powered by GeoNetwork 4.2.3.SNAPSHOT" and a series of social media and utility icons (About, Github, API, Share and comment, Twitter, Facebook, LinkedIn, Email, and RSS).


What is geocat.ch – the default view

geocat.ch Search Map Edit Admin console Documentation Benoit Regamey ADMINISTRATOR English


Back to search Edit Delete Manage record Download Display mode

swissALTI3D multidirectional Hillshade

The hillshade is derived from the digital elevation model swissALTI3D. It illustrates the surface relief of Switzerland and the Principality of Liechtenstein. A hillshade is derived from the elevation model by calculating the exposure of the surface to sunlight at a certain or several angles. The swissALTI3D hillshade uses a multidirectional method that combines light from 6 different sources (sun's positions). The average sun's position is located to the northwest.







Spatial extent



- Representation type**
Grid
- Resolution**
0.5 m
- Coordinate reference system**
EPSG:21781, EPSG:21781, EPSG:5728

API

 WMS	ch.swisstopo.swissalti3d-reliefschattierung	Add to map	
WMS-BGDI Dienst, Layer "swissALTI3D multidirektionales Relief"			
 WMFS	ch.swisstopo.swissalti3d-reliefschattierung	Add to map	
WMFS-BGDI Dienst, Layer "swissALTI3D multidirektionales Relief"			

Format
ASCII XYZ

Powered by GeotNetwork 4.2.3.SNAPSHOT About GitHub API

What is geocat.ch – the editor (GUI)

swissALTI3D Reliefschattierung multidirektional

Categories Group Validate Cancel Save & close Save metadata

Identification info

Title * swissALTI3D Reliefschattierung multidirektional
AI Deutsch Français Italiano English Rumantsch

Alternate title swissALTI3D multidirektionales Relief
AI Deutsch Français Italiano English Rumantsch

Date * Publication 03 / 28 / 2011

Date * Revision 03 / 15 / 2017

Citation identifier

Code * ch.swisstopo.swissalti3d-reliefschattierung

Citation identifier

Collective title Höhendaten (Landesvermessung)
AI Deutsch Français Italiano English Rumantsch

Abstract * Die Reliefschattierung ist ein Derivat aus dem digitalen Höhenmodell swissALTI3D. Es erlaubt die Visualisierung des Reliefs der Schweiz und des Fürstentums Liechtenstein. Die Reliefschattierung ist ein digitales Schummerungsbild bei einem oder mehreren berechneten Lichteinfällen. Für die Darstellung der vorliegenden Reliefschattierung aus swissALTI3D wurden sechs verschiedene Sonnenstände kombiniert. Der mittlere Sonnenstand befindet sich im Nordwesten.
AI Deutsch Français Italiano English Rumantsch

Point of contact

Responsible party

Organisation name Bundesamt für Landestopografie swisstopo

Overview

Choose or drop an image here

Associated resources

+ Add

Online resources

+ Add

Sort by Protocol Uri Title

- swissALTI3D
<https://www.swisstopo.admin.ch/de/geodata/height/alti3d.html>
- Permalink opendata.swiss
Permalink opendata swiss
<https://opendata.swiss/en/perma/1964cc81-5298-460f-9228-41120315be-a8@bundesamtl-fur-landestopografie-swisstopo>
- WMTS
ch.swisstopo.swissalti3d-reliefschattierung
WMTS-FSDI service, layer "swissALTI3D multidirectional hillshade"
<https://wmts.geo.admin.ch/EPSG:3857/1.0.0/WMTSCapabilities.xml?lang=en>
- WMS
ch.swisstopo.swissalti3d-reliefschattierung
WMS-FSDI service, layer "swissALTI3D multidirectional hillshade"
<https://wms.geo.admin.ch/?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities&lang=en>

What is geocat.ch – the editor (XML)

The screenshot displays the geocat.ch XML editor interface. The main window shows an XML document for 'swissALT3D Reliefschattierung multidirektional'. The XML content includes metadata such as file identifiers, language (German), and contact information for the Swiss Federal Office of Topography. A sidebar on the right, titled 'Associated resources', lists various links and services related to the data, including a WMS service and a map preview. A red box highlights the 'XML' option in the sidebar's menu.

```
1 kche:CHE_MD_Metadata xmlns:gco="http://www.isotc211.org/2005/gco"
2   xmlns:che="http://www.geocat.ch/2008/che"
3   xmlns:gmx="http://www.isotc211.org/2005/gmx"
4   xmlns:gts="http://www.isotc211.org/2005/gts"
5   xmlns:srv="http://www.isotc211.org/2005/srv"
6   xmlns:gmd="http://www.isotc211.org/2005/gmd"
7   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8   xmlns:xlink="http://www.w3.org/1999/xlink"
9   xmlns:gml="http://www.opengis.net/gml/3.1"
10  xmlns:gss="http://www.isotc211.org/2005/gss"
11  xmlns:gmi="http://www.isotc211.org/2005/gmi"
12  gco:isoType="gmd:MD_Metadata"
13
14  <gmd:fileIdentifier>
15    <gco:CharacterString>1964cc81-5298-460f-9228-41120315bea8</gco:CharacterString>
16  </gmd:fileIdentifier>
17  <gmd:language>
18    <gmd:LanguageCode codeList="http://www.ioc.gov/standards/iso639-2/" codeListValue="ger"/>
19  </gmd:language>
20  <gmd:characterSet>
21    <gmd:MD_CharacterSetCode codeListValue="utf8"
22      codeList="http://standards.iso.org/ittf/PubliClyAvailableStandards/ISO_19139_Schemas/resources/codelist/ML_gmxCodelists
23      .xml#MD_CharacterSetCode"/>
24  </gmd:characterSet>
25  <gmd:hierarchyLevel>
26    <gmd:MD_ScopeCode codeListValue="dataset"
27      codeList="http://standards.iso.org/ittf/PubliClyAvailableStandards/ISO_19139_Schemas/resources/codelist/ML_gmxCodelists
28      .xml#MD_ScopeCode"/>
29  </gmd:hierarchyLevel>
30  <gmd:contact xlink:href="local://srv/api/registries/entries/d9bb4f52-e97c-42a4-91e2-a7894d8df1cf?lang=ger,fr,ita,eng,roh&amp;process=1"//gmd:CI_RoleCode
31    />
32    <gmd:CI_Contact>
33      <gmd:CI_Individual>
34        <gmd:organisationName xsi:type="gmd:PT_FreeText_PropertyType">
35          <gco:CharacterString>Bundesamt für Landestopografie swisstopo</gco:CharacterString>
36        </gmd:organisationName>
37        <gmd:PT_FreeText>
38          <gmd:LocalisedCharacterString locale="#DE">Bundesamt für Landestopografie swisstopo</gmd:LocalisedCharacterString>
39          <gmd:LocalisedCharacterString locale="#FR">Office fédéral de topographie swisstopo</gmd:LocalisedCharacterString>
40          <gmd:LocalisedCharacterString locale="#IT">Ufficio federale di topografia swisstopo</gmd:LocalisedCharacterString>
41          <gmd:LocalisedCharacterString locale="#EN">Federal Office of Topography swisstopo</gmd:LocalisedCharacterString>
42          <gmd:LocalisedCharacterString locale="#RH">Uffizi federal da topografia swisstopo</gmd:LocalisedCharacterString>
43        </gmd:PT_FreeText>
44      </gmd:CI_Individual>
45      <gmd:organisationName>
46        <gco:CharacterString>Metadaten</gco:CharacterString>
47      </gmd:organisationName>
48      <gmd:PT_FreeText>
49        <gmd:LocalisedCharacterString locale="#DE">Metadaten</gmd:LocalisedCharacterString>
50      </gmd:PT_FreeText>
51    </gmd:CI_Contact>
52  </gmd:contact>
53  </gmd:MD_Metadata>
54
```

Associated resources:

- swissALT3D
<https://www.swisstopo.admin.ch/de/geodata/height/alt3d.html>
- Permalink opendata.swiss
Permalink opendata swiss
<https://opendata.swiss/en/perma/1964cc81-5298-460f-9228-41120315bea8@bundesamt-fur-landestopografie-swisstopo>
- WMTS
ch.swisstopo.swissalt3d-reliefschattierung
WMTS-FSDI service, layer "swissALT3D multidirectional hillshade"
<https://wmts.geo.admin.ch/EPSSG/3857/1.0.0/WMTSCapabilities.xml?lang=en>
- WMS
ch.swisstopo.swissalt3d-reliefschattierung
WMS-FSDI service, layer "swissALT3D multidirectional hillshade"
<https://wms.geo.admin.ch/?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities&lang=en>
- Preview map.geo.admin.ch
Preview map.geo.admin.ch
<https://map.geo.admin.ch/?layers=ch.swisstopo.swissalt3d-reliefschattierung>
- map.geo.admin.ch
<https://map.geo.admin.ch/?topic=swisstopo&lang=de&bgLayer=ch.swisstopo.pixeikarte-farbe&layers=ch.swisstopo.swissalt3d-reliefschattierung&catalogNodes=1459>

geocat.ch architecture



Elasticsearch Index

Multilingual field

The main language from the localized languages is always copied to the “default” item

```
"resourceTitleObject": {  
  "default": "Cartographie des sols agricoles de la plaine du Rhône",  
  "langger": "Thematische Karten der landwirtschaftlichen Böden im Rhonetal",  
  "langfre": "Cartographie des sols agricoles de la plaine du Rhône"  
},
```

Code list field

Labels in all languages and 1 key

```
"cl_topic": [  
  {  
    "default": "Carte de référence de la couverture terrestre",  
    "langger": "Oberflächenbeschreibung",  
    "langeng": "Imagery base maps earth cover",  
    "langroh": "imageryBaseMapsEarthCover",  
    "langita": "Mappe di base",  
    "key": "imageryBaseMapsEarthCover",  
    "langfre": "Carte de référence de la couverture terrestre"  
  },
```

How to retrieve the Index

With the browser web developers tool

Either in the search application or in the default view, look for the POST request ending with “_search”, the request’s response contains the index. Try it live !

geocal.ch Search Map Documentation Sign in English

Back to search Download Display mode

LIDAR Laserscanning-Punktwolke - 2014

Im Projekt "Luftaufnahmen4Z" wird durch die Abteilung Geoinformation des Amtes für Raumentwicklung ARE des Kantons Zürich alle vier Jahre ein hochauflösendes Laserscanning (LIDAR, Mittlere Punktedichte von 8 Pkt/m²) zur Erstellung des digitalen Oberflächenmodells (DOM ZH) sowie des digitalen Terrainmodells (DTM ZH) angefertigt.

Es zeigt die Erdoberfläche mit allen beständigen und sichtbaren Kunstbauten. Als Folge davon entsteht ein DTM ZH (GIS-ZH Nr. 296) und das DOM-ZH (GIS-ZH Nr. 299). Ganzes Kantonsgebiet inkl. 500 m Buffer

Spatial extent

Representation type: vector

Scale: 1/1000

Coordinate reference system: CH1903+_LV95

legal constraints

Format: LAZ (140)

Technical information

Revision: 31-12-2014 Language

Powered by GeoNetwork 4.2.3 SNAPSHOT About GitHub API

```
POST www.geocal.ch/geoportal/parent/obj/jts/... 4.6.12... value: 1
relation: "44"
max_score: 10.932318
...
{
  "id": "7b172ecc-0c2b-4678-8b6f-3b6820b0f65b",
  "type": "doc",
  "doc": "7b172ecc-0c2b-4678-8b6f-3b6820b0f65b",
  "score": 10.932318,
  "_source": {
    "docType": "metadata",
    "metadataIdentifier": "7b172ecc-0c2b-4678-8b6f-3b6820b0f65b",
    "indexingDate": "2023-05-02T11:35:17.221Z",
    "docType": "metadata",
    "document": {
      "metadataIdentifier": "7b172ecc-0c2b-4678-8b6f-3b6820b0f65b",
      "standardNameObject": {
        "default": "GAM03 2+",
        "langauge": "GAM03 2+"
      },
      "indexingDate": "2023-05-02T11:35:17.221Z",
      "dateStamp": "2023-01-16T16:24:49.000Z",
      "mainLanguage": "deu",
      "otherLanguage": [
        "ger"
      ],
      "otherLanguageId": "DE",
      "characterSet": [
      ],
      "resourceType": [
        "dataset"
      ],
      "originalObject": {
        "default": "Amt für Raumentwicklung in - Geoinformation",
        "langauge": "Amt für Raumentwicklung in - Geoinformation"
      },
      "pointOfContactOrgObject": {
        "default": "Amt für Raumentwicklung in - Geoinformation",
        "langauge": "Amt für Raumentwicklung in - Geoinformation"
      },
      "contact": [
      ],
      "hierarchyLevel": [
      ],
      "geometryObject": [
      ],
      "maintenanceAndDateFrequency": [
      ],
      "type": [
      ],
      "spatialRepresentationType": [
      ],
      "modeType": [
      ],
      "function": [
      ],
      "country": [
      ],
      "legislationType": [
      ],
      "resourceTypeObject": {
        "default": "LIDAR Laserscanning-Punktwolke - 2014",
        "langauge": "LIDAR Laserscanning-Punktwolke - 2014",
        "langauge": "LIDAR Laserscanning-Punktwolke - 2014"
      },
      "revisionDateForResource": [
        "2014-12-31T00:00:00.000Z"
      ],
      "revisionYearForResource": "2014",
      "resourceDateForResource": "2014-12",
      "resourceDate": [
      ],
      "resourceTemporalDateRange": [
      ],
      "resourceAbstractObject": {
        "default": "Im Projekt 'Luftaufnahmen4Z' wird durch die Abteilung Geoinformation des Amtes für Raumentwicklung ARE des Kantons Zürich alle vier Jahre ein hochauflösendes Laserscanning (LIDAR, Mittlere Punktedichte von 8 Pkt/m2) zur Erstellung des digitalen Oberflächenmodells (DOM ZH) sowie des digitalen Terrainmodells (DTM ZH) angefertigt. ImZ zeigt die Erdoberfläche mit allen beständigen und sichtbaren Kunstbauten. Als Folge davon entsteht ein DTM ZH (GIS-ZH Nr. 296) und das DOM-ZH (GIS-ZH Nr. 299). Ganzes Kantonsgebiet inkl. 500 m Buffer. Langauge: 'in-Projekt 'Luftaufnahmen4Z', wird durch die Abteilung Geoinformation des Amtes für Raumentwicklung ARE"
      }
    }
  }
}
```

How to search into specific index attributes

In the free text search bar, use the q() query with a lucene query syntax



Search for word “lidar” in the title of metadata in all languages

Lucene query syntax

q(resourceAbstractObject.langeng:(Protected areas))

Index field name

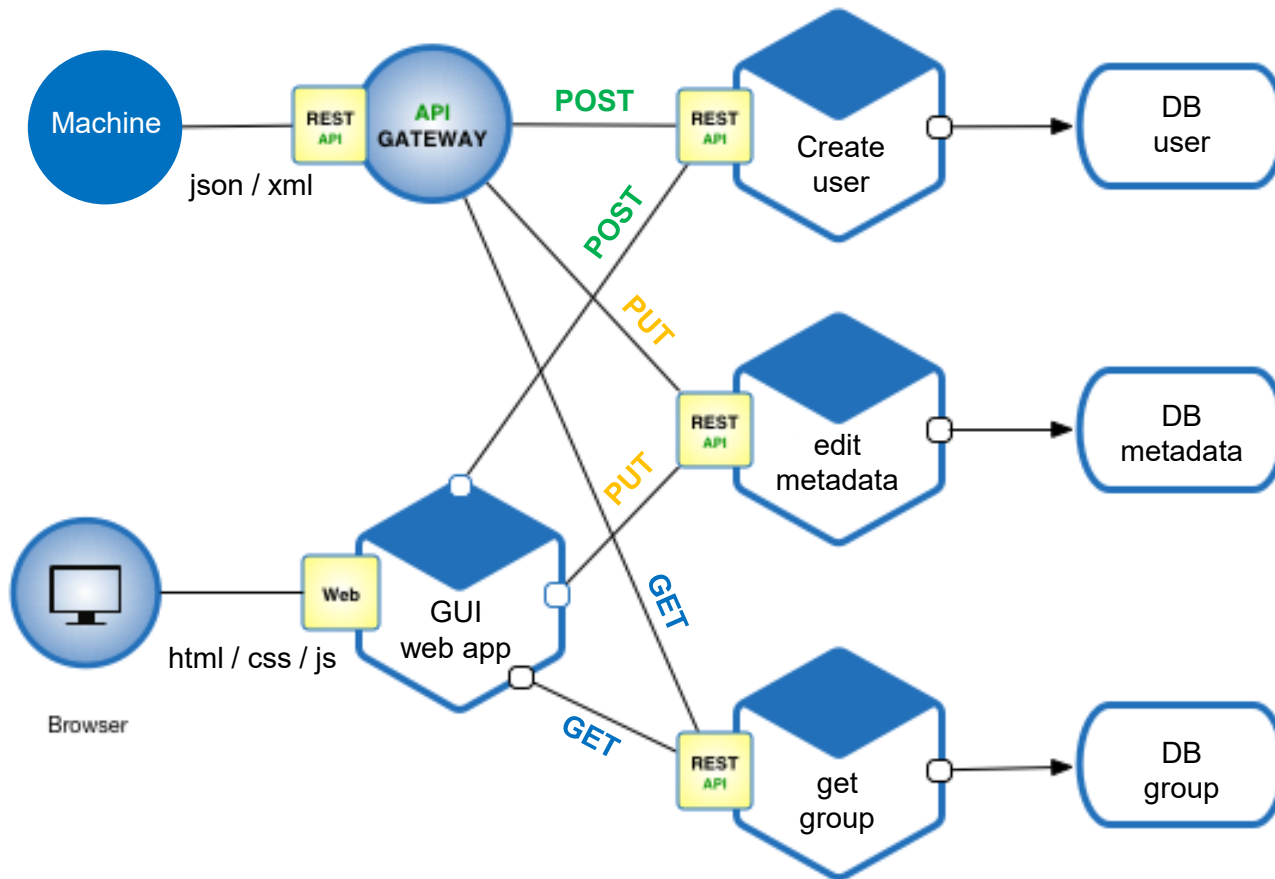
If object,
item name
Wildcard =
*

value

q(-groupOwner:68 OR -groupOwner:34923929)

Search for metadata not belonging to group ID 68 or 34923929

geocat.ch – microservice architecture



geonetwork (geocat) API

Powered by GeoNetwork 4.2.3.SNAPSHOT

About

 Github

API

Share and comment



API swagger on prod : <https://www.geocat.ch/geonetwork/doc/api/index.html>

API swagger on int : <https://geocat-int.dev.bgdi.ch/geonetwork/doc/api/index.html>

Example with GET request /groups :

In the swagger

<https://www.geocat.ch/geonetwork/doc/api/index.html#/groups/getGroups>

In the browser : <https://geocat-int.dev.bgdi.ch/geonetwork/srv/api/groups>

Elasticsearch API endpoint

Replace the former “q” search endpoint

Uses the elasticsearch “_search” endpoint

<https://www.elastic.co/guide/en/elasticsearch/reference/current/search-search.html>

Example, get all metadata from group id = 3 and that are not template. This is a POST request, the query goes into the request body as json :

```
{
  "from": 0,
  "query": {
    "bool": {
      "must": [
        {
          "query_string": {
            "query": "(groupOwner:\\\"3\\\")",
            "default_operator": "AND"
          }
        },
        {
          "terms": {
            "isTemplate": [
              "n"
            ]
          }
        }
      ]
    }
  },
  "track_total_hits": true,
  "sort": {"_id": "asc"}
}
```


Elasticsearch API endpoint

Only 10 results, increase the number of results with “size” parameter

```
{
  "from": 0,
  "query": {
    "bool": {
      "must": [
        {
          "query_string": {
            "query": "(groupOwner:\\\"3\\\")",
            "default_operator": "AND"
          }
        },
        {
          "terms": {
            "isTemplate": [
              "n"
            ]
          }
        }
      ]
    }
  },
  "track_total_hits": true,
  "sort": {"_id": "asc"},
  "size": 100
}
```

Elasticsearch API endpoint

Deep paginated search -> use a loop and the “search_after” parameter
Because we sort the results by “_id”, in each iteration, we can use the last id as search_after parameter to search the next 100 results.

E.g. iteration 2

```
{
  "from": 0,
  "query": {
    "bool": {
      "must": [
        {
          "query_string": {
            "query": "(groupOwner:'3\\')",
            "default_operator": "AND"
          }
        },
        {
          "terms": {
            "isTemplate": [
              "n"
            ]
          }
        }
      ]
    }
  },
  "track_total_hits": true,
  "sort": { "_id": "asc" },
  "size": 100,
  "search_after": ["2ce3e737-7c21-46a8-8825-f5ece731be58"]
}
```

Elasticsearch API endpoint

How to know the elasticsearch `_search` request's body structure ?

Once again, use the web developers tools to see what request your browser sends to the server ! Live demo

The image shows a browser window with the geocat.ch website. The search bar contains the text 'lidar'. Below the search bar, there are several search results cards for 'Normalisiertes Digitales Oberflächenmodell' (Digital Surface Model) for buildings and vegetation in the Canton of Zurich. The browser's developer tools are open to the Network tab, showing a list of requests. The selected request is a POST to 'www.geocat.ch/_search' with a body of JSON. The JSON body is expanded to show the following structure:

```
{
  "_source": [...],
  "includes": [...],
  "aggregations": {...},
  "from": 0,
  "query": {...},
  "function_score": {...},
  "boost": "5",
  "functions": [...],
  "query": {...},
  "bool": {...},
  "must": [...],
  "query_string": {...},
  "default_operator": "AND",
  "query": "any\\:\\\\(lidar) OR any.common:(lidar) OR resourceTitleObject:\\\\(lidar)*2 OR resourceTitleObject:\\\\(lidar)*6 AND (tag.default:\\\\opendata.swiss*) AND (format:\\\\(ASCII|XYZ|xyz))",
  "terms": {...},
  "is_template": [...],
  "score_mode": "multiply",
  "size": 30,
  "sort": [...],
  "score": [...],
  "track_total_hits": true
}
```

geonetwork (geocat) API authentication

Many API requests require an authentication and the appropriate permission.

The API uses a basic authentication (username and password) and a XSRF Token.

To retrieve the XSRF token, make first a request with a basic authentication to :

<https://www.geocat.ch/geonetwork/srv/api/me>

The token is then in the cookies under the key “XSRF-TOKEN”

In all authenticated requests, the basic authentication must be given and the token must be given in the cookies (like in the response from the “/me” request) and in the headers as well, under the key “X-XSRF-TOKEN”.

If you are using a programming language that has a library to call API requests with the concept of session, it is much more convenient to store the basic authentication, the cookies with the XSRF token and the headers with the XSRF token into a session object.

geonetwork (geocat) API authentication

Help with the API and example how to retrieve XSRF token and using it in later requests (usin cURL in UNIX environment)

<https://geonetwork-opensource.org/manuals/4.0.x/en/api/the-geonetwork-api.html#>

```
SERVER=http://localhost:8080/geonetwork
CATALOGUSER=admin
CATALOGPASS=admin
```

```
type=series
from=0
size=1000
```

```
rm results.json
rm -f /tmp/cookie;
```

```
curl -s -c /tmp/cookie -o /dev/null \
-X GET \
--user $CATALOGUSER:$CATALOGPASS \
-H "Accept: application/json" \
"$SERVER/srv/api/me";
```

```
TOKEN=`grep XSRF-TOKEN /tmp/cookie | cut -f 7`;
JSESSIONID=`grep JSESSIONID /tmp/cookie | cut -f 7`;
```

```
curl "$SERVER/srv/api/search/records/_search" \
-X 'POST' \
-H 'Accept: application/json, text/plain, */*' \
-H 'Content-Type: application/json;charset=UTF-8' \
--data-raw "{\"query\":{\"query_string\":{\"query\": \"+isHarvested:false +resourceType: $type\"},\"from\":$from,\"size\":$size,\"_source\": {\"include\": [\"resourceTitleObject.default\"],\"sort\": [{\"resourceTitleObject.default.keyword\": \"asc\"}]}}\" \
-H "X-XSRF-TOKEN: $TOKEN" -H "Cookie: XSRF-TOKEN=$TOKEN; JSESSIONID=$JSESSIONID" \
--compressed \
-o results.json
```

geonetwork (geocat) API authentication

E.g. using python

```
import requests

session = requests.Session()
session.cookies.clear()

# Store basic authentication into the session object
session.auth = ("username", "password")

# Make request to get cookies with XSRF token into the session object
session.post(url='https://www.geocat.ch/geonetwork/srv/eng/info?type=me')

# copy XSRF token from cookies and store it in the headers
cookies = session.cookies.get_dict()
token = cookies["XSRF-TOKEN"]
session.headers.update({"X-XSRF-TOKEN": token})

# Make any requests requiring authentication
session.delete(url="https://geocat.ch/geonetwork/srv/api/records?uuids=metadata-uuid")
```

The swagger uses the browser session. So if you are logged-in in geocat.ch and you use the same browser with the swagger, you are all set !

- XSRF token are valid only for short time (c.a. 10 min) if idle...
- If you are under a proxy, proxies info must be given for each requests sent !

Batch editing API request

<https://www.geocat.ch/geonetwork/doc/api/index.html#/records/batchEdit>

Geonetwork documentation : <https://geonetwork-opensource.org/manuals/4.0.x/en/user-guide/workflow/batchediting.html>

PUT request

Request's body

```
[
  {
    "xpath": "string",
    "value": "string"
  },
  {
    "xpath": "string",
    "value": "string"
  }
]
```

3 possible operations : insert, replace, delete

```
"value": "<gn_add> add something new at xpath location</gn_add>"
"value": "<gn_replace> replace tag content at xpath location </gn_replace>"
"value": "<gn_delete> delete content at xpath location</gn_delete>"
```

Batch editing API request

Can be processed on multiple metadata

With the parameter `uuids`, which is an array... or with the parameter `bucket=e101`, this takes selection of metadata in the current session (useful for the swagger).

Xpath syntax can be quite complex but powerful

https://www.w3schools.com/xml/xpath_intro.asp
<https://devhints.io/xpath>

E.g.

Remove status if status is equal to “obsolete”

```
[
  {
    "xpath": ".//gmd:identificationInfo//gmd:status/gmd:MD_ProgressCode[@codeListValue='obsolete']/parent::*",
    "value": "<gn_delete></gn_delete>"
  }
]
```


Batch editing API request

E.g. Add new identifier.

When you add something, the xpath can point down to the parent node (geocat knows where to put the new tag).

You can add and replace entire XML section

When you add new XML snippet, you have to provide all namespace used in the snippet before you use it (e.g. `xmlns:gmd` and `xmlns:gco` below)

Don't forget to replace XML special character :
< (<), & (&), > (>), " ("), and ' (')

```
[
  {
    "xpath": "./gmd:identificationInfo[1]/*/gmd:citation[1]/gmd:CI_Citation",
    "value": "<gn_add><gmd:identifier xmlns:gmd='http://www.isotc211.org/2005/gmd'
xmlns:gco='http://www.isotc211.org/2005/gco'>\" \
      <gmd:MD_Identifier>\" \
        <gmd:code>\" \
          <gco:CharacterString>layer_id</gco:CharacterString>\" \
        </gmd:code>\" \
      </gmd:MD_Identifier>\" \
    </gmd:identifier>\",
    </gn_add>\"
  }
]
```

Batch editing API request

E.g. Complex xpath

Select a `transferOption` node that has `only one resource` and the `protocol = "OGC:WMS"` and where the `URL` contains the phrase `"wms.geo.admin.ch"`.

```
xpath = "./gmd:distributionInfo//gmd:transferOptions[count(./*/*)=1\"\n    \" and ./gmd:protocol/gco:CharacterString[contains(text(), 'OGC:WMS')]]\"\n    \" and (./gmd:URL[contains(text(), 'wms.geo.admin.ch')]]\"\n    \" or ./che:LocalisedURL[contains(text(), 'wms.geo.admin.ch')])]]\"
```

You can explore and/or use the batch editing API request with a GUI implementation on the `geocat.ch` frontend application

<https://www.geocat.ch/geonetwork/srv/fre/catalog.edit#/batchedit>

You can preview the change in the XML !

API request response

Always good to check for an API request's response

Status code :

2xx : successful

3xx : redirect

4xx : client error (you made a mistake)

404 : resource not found

403 : not allowed

5xx : server error (the server made an error)

Response's content :

Usually **JSON**, or **XML**

You can have a successful status code (e.g. 200) in terms of http request but the request is still unsuccessful in terms of what you want to do (no processing, metadata not found, xpath not found, etc...)

E.g. batch editing request's response

```
{
  "errors": [],
  "infos": [],
  "uuid": "5da566b8-0e5a-41e4-8dc5-32ad4bfaa46d",
  "metadata": [],
  "metadataErrors": {},
  "metadataInfos": {
    "47380071": [
      {
        "message": "Metadata updated.",
        "uuid": "638f99cd-0532-4ba1-84e8-b572f17fec6a",
        "draft": false,
        "approved": false,
        "date": "2023-05-08T14:53:49.395Z"
      }
    ]
  },
  "numberOfNullRecords": 0,
  "numberOfRecordsProcessed": 1,
  "numberOfRecordsUnchanged": 0,
  "numberOfRecordsWithErrors": 0,
  "numberOfRecords": 1,
  "numberOfRecordNotFound": 0,
  "numberOfRecordsNotEditable": 0,
  "startIsoDateTime": "2023-05-08T14:53:48.6Z",
  "endIsoDateTime": "2023-05-08T14:53:49.556Z",
  "elapsedTimeInSeconds": 0,
  "totalTimeInSeconds": 0,
  "running": false,
  "type": "SimpleMetadataProcessingReport"
}
```

Search & replace API request

Apply a search and replace at the database level

<https://geocat-int.dev.bgdi.ch/geonetwork/doc/api/index.html#/processes/processSearchAndReplace>

Does not follow the logic of the XML structure, it is a text based search and replace (regular expressions are supported)

Like the batch editing request, you can use the search & replace API request with a GUI implementation on the geocat.ch frontend application

You can preview the change in the XML !

Dealing with subtemplates

Subtemplates or “reusable objects” are snippet of XML that are stored separately than the metadata and represent [contacts](#), [extents](#) and [formats](#).

The subtemplates are linked to the metadata by their UUID. Hence, it is possible to link subtemplates in many metadata (many-to-many relation).

The content comes only from the subtemplate, when a subtemplate is updated, the content of all metadata using it is also updated.

Thus, if we want to edit a metadata in a M2M way, it is useless to edit the content of contact, extent or format if they are subtemplates. It will be overridden by the subtemplate itself when saving the metadata.

Dealing with subtemplates

E.g. Contact for the metadata

```
<gmd:contact xlink:href="local://srv/api/registries/entries/aa0ddc95-736b-4440-a81c-2b704ee8b4b1?lang=ger,fre,ita,eng,roh&process=gmd:role/gmd:CI_RoleCode/@codeListValue~pointOfContact&schema=iso19139.che">  
</gmd:contact/>
```

aa0ddc95-736b-4440-a81c-2b704ee8b4b1 : the subtemplate UUID

pointOfContact : the code list key for contact role = "point of contact"

E.g. Extent

```
<gmd:extent xlink:href="local://srv/api/registries/entries/geocatch-subtpl-extent-hoheitsgebiet-5601?lang=ger,fre,ita,eng,roh&schema=iso19139.che">  
</gmd:extent/>
```

geocatch-subtpl-extent-hoheitsgebiet-5601 : the subtemplate UUID

E.g. Distribution format

```
<gmd:distributionFormat xlink:href="local://srv/api/registries/entries/22f3a6ef-9d98-41f6-9f86-51ef17e5e878?lang=ger,fre,ita,eng,roh&schema=iso19139.che">  
</gmd:distributionFormat/>
```

22f3a6ef-9d98-41f6-9f86-51ef17e5e878 : the subtemplate UUID

Dealing with subtemplates

How can we get the subtemplates UUID ?

Easiest way is to use the GUI editor, select manually a contact, extent or format and then explore the metadata in the editor XML view to fetch the UUID and see how geocat makes the link to the subtemplate.

For extents that are administrative boundaries, the geocat team manages subtemplates with defined UUID :

- Municipalities : **geocatch-subtpl-extent-hoheitsgebiet-{gmdnr}** where **gmdnr** = BFS municipality number
- Districts : **geocatch-subtpl-extent-bezirk-{bznr}** where **bznr** = BFS district number
- Cantons : **geocatch-subtpl-extent-kantonsgebiet-{ktnr}** where **ktnr** = BFS canton number
- Countries : **geocatch-subtpl-extent-landesgebiet-{code_iso}** where **code_iso** = country ISO code (CH, FL)

Dealing with keywords

Keywords work in a similar way as the subtemplates

Keywords are stored in multiple thesaurus (separately than the metadata) and are linked to metadata with the [thesaurus ID](#) and [keywords ID](#).

The content comes only from the thesaurus, when a keyword is updated, the keyword is updated in all metadata using it.

Thus, if we want to edit a metadata in a M2M way, it is useless to edit the content of keywords. It will be overridden by the thesaurus when saving the metadata.

Dealing with keywords

```
<gmd:descriptiveKeywords
xlink:href="local://srv/api/registries/vocabularies/keyword?skipdescriptivekeywords=true&
thesaurus=local.theme.geocat.ch&id=http://geocat.ch/concept#e6485c01-fe69-485e-b194-
035f682463db,http://custom.shared.obj.ch/concept#ae677a16-f81a-4533-9243-a87831115079
&lang=ger,fr,ita,eng,roh">
</gmd:descriptiveKeywords/>
```

thesaurus={thesaurus ID} e.g. local.theme.geocat.ch

id={keyword ID},{Keyword ID},...

e.g. <http://geocat.ch/concept#e6485c01-fe69-485e-b194-035f682463db>

e.g. <http://custom.shared.obj.ch/concept#ae677a16-f81a-4533-9243-a87831115079>

Dealing with keywords

How can we get thesaurus and keywords ID ?

Easiest way is to use the GUI editor, select manually keywords and then explore the metadata in the editor XML view to fetch thesaurus and keyword ID and see how geocat makes the link to keywords.

Or...

Browse the thesaurus directly in the admin panel.

<https://www.geocat.ch/geonetwork/srv/eng/admin.console#/classification>

Q&A