geocat.ch

THE SWISS **GEO**GRAPHIC
**CAT**ALOGUE

# geocat.ch API

29.04.2025

Olivier Curdy
olivier.curdy@swisstopo.ch

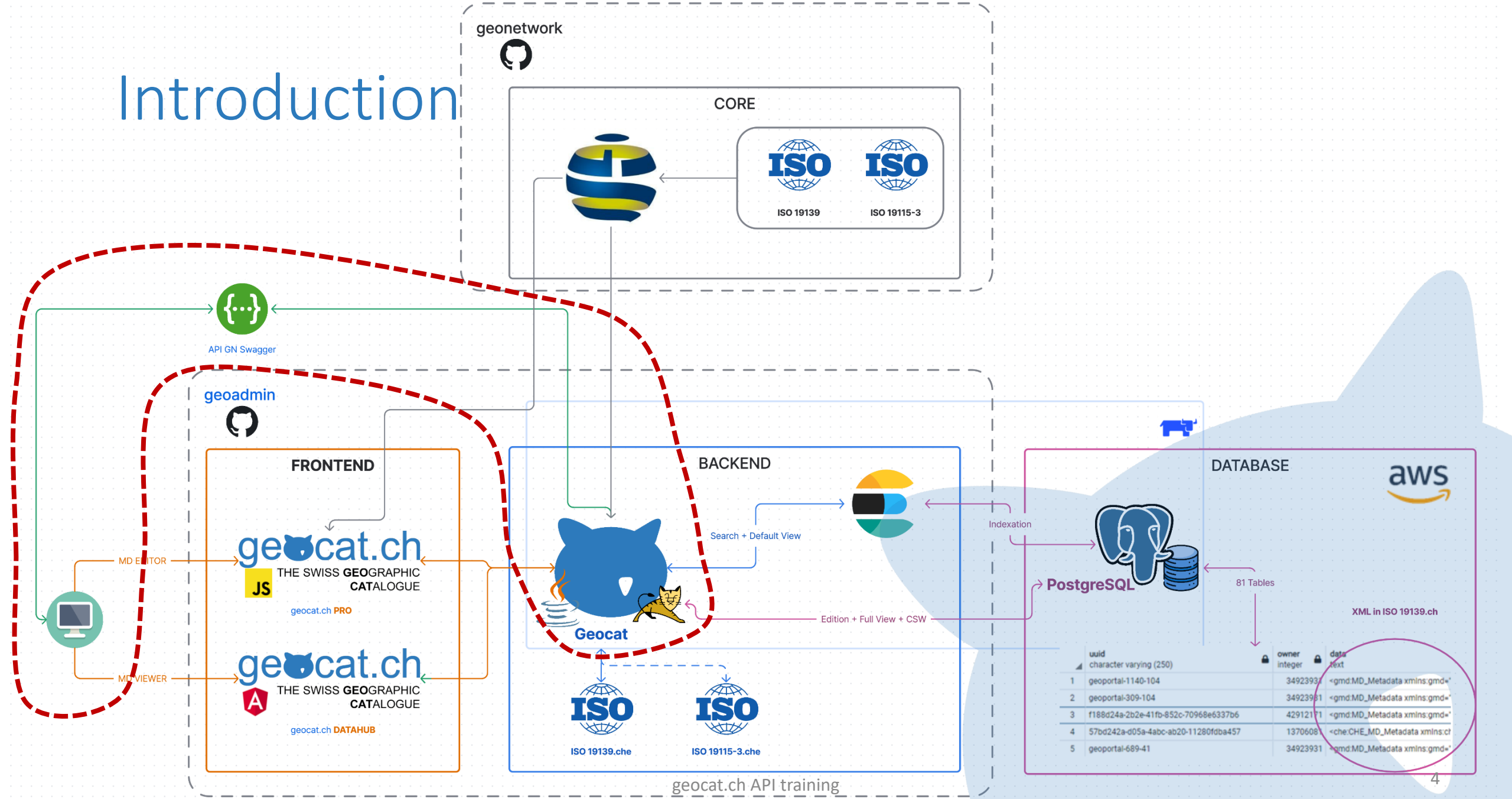
# Plan

- Introduction
- Login
- Search
- Download metadata
- Edit metadata
- Add metadata
- Best practices

# Introduction

- geocat.ch is a modified instance of GeoNetwork 4.2.3
- geocat.ch 3 instances with 2 websites
  - Prod:
    - Datahub: https://www.geocat.ch/datahub
    - Pro: https://www.geocat.ch/geonetwork
  - Int:
    - Datahub: https://geocat-int.dev.bgdi.ch/datahub
    - Pro: https://geocat-int.dev.bgdi.ch/geonetwork
  - Dev (under development for version 4.4.6)
- Metadata are stored in XML following the iso19139.che schema (iso19139 schema for Switzerland)
  - New schema ISO 19115-3:2018.che (eCH-0271) in progress

# Introduction



# Login - Introduction

- Many API requests require authentication and sufficient rights. geocat uses an authentication mechanism based on:
  - a **username** and **password** (basic authentication)
  - an **XSRF token** (Cross-Site Request Forgery Token) which must be included in cookies and headers for POST, PUT or DELETE requests.
- Programming languages with concept of **session** in a library to call API requests to store:
  - Basic authentication into a session
  - Retrieve the token from the cookies after a GET to “srv/api/me” for example
  - Headers with the XSRF token into a session object.
- Only certain authorised users with specific rights can modify metadata

# Login - Request

- Request “me”:
  - Base:
    - URL: /srv/api/me
    - HTTP Method: **GET**
  - Authentication:
    - Authentication Method: Basic
    - HTTP Authentication Username: geocat username
    - HTTP Authentication Password: geocat password
  - Query String Parameter:
    - None
  - Headers
    - Accept: application/json

# Login - Examples

- GUI:
  - INT: https://geocat-int.dev.bgdi.ch/geonetwork/srv/eng/catalog.search#/home
  - PROD: https://www.geocat.ch/geonetwork/srv/eng/catalog.search#/home
- Swagger:
  - INT: https://geocat-int.dev.bgdi.ch/geonetwork/doc/api/index.html#/me/getMe
  - PROD: https://www.geocat.ch/geonetwork/doc/api/index.html#/me/getMe
- **Python**
  - **https://github.com/geoadmin/geocat-api-test/blob/master/1_get_token.py**

# Search - Introduction

- geocat.ch allows powerful searches to be carried out on all available metadata, using Elasticsearch-type queries.
- ES index GeoNetwork metadata XML with a lot of information
- This search is carried out via a POST type request, with a JSON body detailing the criteria.
- More information on Elasticsearch “ _search” endpoint

# Search - Request

- Request “search”:
  - Base:
    - URL: /search/records/_search
    - HTTP Method: **POST**
  - Authentication:
    - None
  - Query String Parameter:
    - Query: Json body (see example)
    - *Bucket : metadata (by default), or selection bucket*
  - Headers
    - Accept: application/json
    - Content-Type: application/json

simple_query_string

# Search – JSON query

- Pagination 0-100 firsts results
- Sort results by ascending ID
- Track total of matches
- Filter must have:
  1. Term “ecab” in
     - A generic space (any) for broad searching (title, description…)
     - A structured object (resourceTitleObject) for multilingual titles
  2. Owner 3 (Canton of Fribourg)
  3. Keyword “opendata.swiss”)
  4. Only metadata (Ø Templates)

```
{
  "from": 0,
  "query": {
    "bool": {
      "must": [
        {
          "simple_query_string": {
            "query": "ecab",
            "fields": ["any.*", "resourceTitleObject.*"],
            "default_operator": "OR"
          }
        },
        {
          "term": {
            "groupOwner": "3"
          }
        },
        {
          "term": {
            "tag.default": "opendata.swiss"
          }
        },
        {
          "terms": {
            "isTemplate": ["n"]
          }
        }
      ]
    }
  },
  "track_total_hits": true,
  "sort": {"_id": "asc"},
  "size": 100
}
```

# Search - Examples

- GUI:
  - INT: https://geocat-int.dev.bgdi.ch/geonetwork/srv/eng/catalog.search#/search
  - PROD: https://www.geocat.ch/geonetwork/srv/eng/catalog.search#/search
- **Swagger:**
  - INT: https://geocat-int.dev.bgdi.ch/geonetwork/doc/api/index.html#/search/search
  - **PROD: https://www.geocat.ch/geonetwork/doc/api/index.html#/search/search**
- Python
  - https://github.com/geoadmin/geocat-api-test/blob/master/2_search.py

# Download - Introduction

- geocat.ch allows you to download a metadata file in XML or MEF format (ZIP archive).
- Only the UUID of the file is required.
- The metadata records are downloaded in the 19139.che schema.
- The MEF format is useful for retrieving a file with its possible resources (logos, attachments, keyword library, etc.) and can be downloaded as a compressed zip file.

# Download - Request

- Request “records formatters”:
  - Base:
    - URL XML: /srv/api/records/{metadataUuid}/formatters/xml
    - URL MEF: /srv/api/records/{metadataUuid}/formatters/zip
    - HTTP Method: **GET**
  - Authentication:
    - None
  - Query String Parameter:
    - uuid: metadata uuid
  - Headers
    - XML -> Accept: application/xml
    - MEF -> Accept: application/x-gn-mef-2-zip

# Download - Examples

- GUI:
  - INT: https://geocat-int.dev.bgdi.ch/geonetwork/srv/eng/catalog.search#/metadata/deaf9025-3ace-4f02-8f46-bcc874543dc4
  - PROD: https://www.geocat.ch/geonetwork/srv/eng/catalog.search#/metadata/deaf9025-3ace-4f02-8f46-bcc874543dc4
- Swagger:
  - INT:
    - XML: https://geocat-int.dev.bgdi.ch/geonetwork/doc/api/index.html#/records/getRecordAs
    - MEF: https://geocat-int.dev.bgdi.ch/geonetwork/doc/api/index.html#/records/getRecordAsZip
  - PROD:
    - XML: https://www.geocat.ch/geonetwork/doc/api/index.html#/records/getRecordAs
    - MEF: https://www.geocat.ch/geonetwork/doc/api/index.html#/records/getRecordAsZip
- **Python:**
  - **XML: https://github.com/geoadmin/geocat-api-test/blob/master/3_get_xml_records.py**
  - MEF: geocat-api-test/4_get_mef_records.py at master · geoadmin/geocat-api-test

# Edit - Introduction

- Single editing is possible, but we will look at batch editing (which also works for a single record)
- Batch editing need:
  - XPath: language that allows a specific node (element, attributes…) to be localised in an XML following the three structure.
    - XPath syntax documentation
  - Value: text or XML with specific tags
    - <gn_add>add text or XML at xpath location</gn_add>
    - <gn_replace>replace tag content at xpath location</gn_replace>
    - <gn_delete>delete content at xpath location</gn_delete>

simple_query_string

# Edit - Request

- Request “Batch editing”:
  - Base:
    - URL: /srv/api/records/batchediting
    - URL Preview: /srv/api/records/batchediting/preview
    - HTTP Method: **PUT** (Preview: POST)
  - Authentication:
    - Authentication Method: Basic
  - Query String Parameter:
    - uuid: metadata uuid
    - Json query (see example)
      - XPath: path to the node to be edited in the XML
      - Value: text or XML with tags add/replace/delete
  - Headers
    - Accept: application/json
    - Content-Type: application/json

# Edit – XPath

| Syntax | Meaning | Definition | Example |
|---|---|---|---|
| / | Root node | Selects from the absolute root of the document | */gmd:identificationInfo/che:CHE_MD_DataIdentification/gmd:resourceConstraints*<br>***resourceConstraints* and children** |
| // | Any descendant (recursive) | Selects nodes anywhere in the document (depth-agnostic) | *//gmd:resourceConstraints*<br>**All nodes *resourceConstraints* in the XML three** |
| ./ | Current context | Selects relative to the current node | *./gmd:useConstraints*<br>**Direct children of the root *resourceConstraints/useConstraints*** |
| .// | Descendants of context | Selects descendants of the current node | .//gmd:otherConstraints<br>**Searches below the root *otherConstraints* at any level** |
| @ | Attribute selector | Selects attributes of elements | /gmd:identificationInfo/che:CHE_MD_DataIdentification/gmd:resourceConstraints@xlink:href<br>**Attribut *xlink:href* of contraints (if exists)** |
| * | Wildcard | Matches any element node | */gmd:identificationInfo/*/gmd:resourceConstraints*<br>**Use a wildcard to ignore the level *che:CHE_MD_DataIdentification*** |
| [] | Predicate (filter) | Filters nodes based on conditions | */gmd:identificationInfo/che:CHE_MD_DataIdentification/gmd:resourceConstraints[gmd:useConstraints]*<br>**Only constraints with a child *useContraints*** |

# Edit - Examples

- GUI:
  - INT: https://geocat-int.dev.bgdi.ch/geonetwork/srv/eng/catalog.edit#/batchedit
  - PROD: https://www.geocat.ch/geonetwork/srv/eng/catalog.edit#/batchedit
- Swagger:
  - INT: https://geocat-int.dev.bgdi.ch/geonetwork/doc/api/index.html#/records/batchEdit
  - PROD: https://www.geocat.ch/geonetwork/doc/api/index.html#/records/batchEdit
- Python
  - https://github.com/geoadmin/geocat-api-test/blob/master/5_edit_records.py
- FME

# Upload - Introduction

- geocat allows to upload a xml or MEF/ZIP file.
- The metadata must be in iso19139.che schema.
- In Swagger, importing an XML file does not work. Use curl or a Python script.
- Upload a metadata is not directly published (possibility to do it via API too).
- Possibility to overwrite a record with the same uuid or to create a new one.

# Upload - Request

- Request “records insert file”:
  - Base:
    - URL: /srv/api/records
    - HTTP Method: **POST**
  - Authentication:
    - Authentication Method: Basic
  - Query String Parameter:
    - metadataType : METADATA (or TEMPLATE, SUB_TEMPLATE…)
    - uuidProcessing : NOTHING (same uuid, possible conflict!), GENERATEUUID (new uuid), OVERWRITE
    - group : group number of the metadata
  - Headers
    - Accept : application/json
    - XML -> file : text/xml
    - MEF -> file : application/zip

# Upload - Examples

- GUI:
  - INT: https://geocat-int.dev.bgdi.ch/geonetwork/srv/eng/catalog.search#/metadata/deaf9025-3ace-4f02-8f46-bcc874543dc4
  - PROD: https://www.geocat.ch/geonetwork/srv/eng/catalog.search#/metadata/deaf9025-3ace-4f02-8f46-bcc874543dc4
- Swagger:
  - INT:
    - XML: https://geocat-int.dev.bgdi.ch/geonetwork/doc/api/index.html#/records/getRecordAs
    - MEF: https://geocat-int.dev.bgdi.ch/geonetwork/doc/api/index.html#/records/getRecordAsZip
  - PROD:
    - XML: https://www.geocat.ch/geonetwork/doc/api/index.html#/records/getRecordAs
    - MEF: https://www.geocat.ch/geonetwork/doc/api/index.html#/records/getRecordAsZip
- **Python**
  - **https://github.com/geoadmin/geocat-api-test/blob/master/6_upload_record.py**

# Best practices

- Use session to store cookies + XSRF token in python or other.
- Validate your metadata before publication :
  - https://www.geocat.ch/geonetwork/doc/api/index.html#/records/validateRecord
- Use the INT environment for testing.
- Use batch editing for bulk changes and use ‘Preview’ request first.