WHAT IS LAYMAN AND HOW IT WORKS

JIŘÍ KOZEL

Workshop 4

Big and Open Data and Innovative Hubs in Agriculture, Transport and Rural Development

Czech University of Life Sciences, Prague

January 29, 2020

ABOUT ME

- 14 years of work experience in geoinformatics
- started at Masaryk University
  - GIS specialist
  - software developer
  - system analyst
  - leader of GIS group @ Institute of Computer Science

ABOUT ME

- full-time **freelancer** since 2016
  - maitainer of OpenMapTiles for 2 years
  - BrnoUrganGrid
  - technical leader of GIS4DIS project
  - Layman

ABOUT ME

- geospatial data
- data modeling, analysis, and processing
- web systems
- maps
- automation and optimization

- open-source tools
- pen & paper

WORKSHOP 4

https://github.com/jirik/layman-workshop

- Karel Charvát - Purpose of workshop
- Jiří Kozel - What is Layman and how it works
- Raitis Berzins - Map composition
- Jiří Kozel, Jiří Kvapil - How to install Layman in cloud
- Jiří Kozel - Layman API
- Jiří Kozel - Authentication and authorization
- Jiří Kozel - Interaction with Metadata (Micka)
- Jan Vrobel - QGIS plugins for accessing maps and map composition from server
- Jan Vrobel - QGIS plugin for Web data publishing using Layman
- Raitis Berzins - HSLayers NG as client for Layman

WHAT IS LAYMAN?

Ask ~~Google~~ DuckDuckGo!

JAKE LAYMAN

LAYMAN

*someone who is **not a professional** in a given field*

from Wikitionary

SO THE LAYMAN IS NOT SO FAMOUS...

...YET!

WHAT IS LAYMAN?

- web service for publishing geospatial data online through REST API
- developed since 2018 as part of Databio and Sieusoil projects
- written in Python, published under GNU-GPL license at GitHub
- https://github.com/jirik/layman

HOW LAYMAN WORKS?

1. Input
    - vector data in **GeoJSON** or **ShapeFile** format
    - cartographic style in OGC Styled Layer Descriptor or Symbology Encoding format
2. **Layman's Magic**
3. Output
    - standardized OGC APIs
        - Web Map Service
        - Web Feature Service
        - Catalogue Service

LAYMAN KEY FEATURES

Layman supports two main models of geospatial data:

- **Layer** is created from combination of vector data (GeoJSON or ShapeFile) and visualization (SLD or SE style).
- **Map** is collection of layers described in JSON format. Also known as **map composition**.

LAYMAN KEY FEATURES

There are **multiple client applications** for communication with Layman through its REST API:

- simple web client shipped with Layman
- QGIS desktop client
- HSLayers library

LAYMAN KEY FEATURES

Layman`s **security system** uses two well-known concepts: authentication and authorization.

Common configuration

- **authentication** based on widely used OAuth2 protocol
- **authorization** ensuring that only owner of the data may edit it.

LAYMAN KEY FEATURES

- Large data files can be easily uploaded from browser thanks to chunk upload.
- Asynchronous processing ensures fast communication with REST API.
- Processing tasks can be distributed on multiple servers.

LAYMAN KEY FEATURES

- Layman stands on the shoulders of widely used programs like Flask, PostgreSQL, PostGIS, GDAL, GeoServer, Celery, and Redis.

LAYMAN'S MAGIC

1. wait till all data is uploaded
2. start asynchronous tasks
    1. import vector data into PostgreSQL
    2. publish vector data to GeoServer (WMS, WFS)
    3. publish style to GeoServer (SLD, SE)
    4. generate thumbnail
    5. publish metadata to Micka (CSW)

WHAT IS LAYMAN?

SUMMARY

- web service for publishing geospatial data online through REST API
- developed since 2018 as part of Databio and Sieusoil projects
- written in Python, published under GNU-GPL license at GitHub
- https://github.com/jirik/layman

HOW TO INSTALL LAYMAN

JIŘÍ KOZEL

Workshop 4

Big and Open Data and Innovative Hubs in Agriculture, Transport and Rural Development

Czech University of Life Sciences, Prague

January 29, 2020

WORKSHOP 4

https://github.com/jirik/layman-workshop

- Karel Charvát - Purpose of workshop
- Jiří Kozel - What is Layman and how it works
- Raitis Berzins - Map composition
- Jiří Kozel, Jiří Kvapil - How to install Layman in cloud
- Jiří Kozel - Layman API
- Jiří Kozel - Authentication and authorization
- Jiří Kozel - Interaction with Metadata (Micka)
- Jan Vrobel - QGIS plugins for accessing maps and map composition from server
- Jan Vrobel - QGIS plugin for Web data publishing using Layman
- Raitis Berzins - HSLayers NG as client for Layman

WARNING!

THIS PART IS GOING TO BE TECHNICAL!

STEPS

1. Provide your **public SSH key**
2. Connect to your remote machine
3. Install requirements
4. Install Layman

MATERIALS

https://github.com/jirik/layman-workshop

LAYMAN'S REST API

JIŘÍ KOZEL

Workshop 4

Big and Open Data and Innovative Hubs in Agriculture, Transport and Rural Development

Czech University of Life Sciences, Prague

January 29, 2020

WORKSHOP 4

https://github.com/jirik/layman-workshop

- Karel Charvát - Purpose of workshop
- Jiří Kozel - What is Layman and how it works
- Raitis Berzins - Map composition
- Jiří Kozel, Jiří Kvapil - How to install Layman in cloud
- Jiří Kozel - Layman API
- Jiří Kozel - Authentication and authorization
- Jiří Kozel - Interaction with Metadata (Micka)
- Jan Vrobel - QGIS plugins for accessing maps and map composition from server
- Jan Vrobel - QGIS plugin for Web data publishing using Layman
- Raitis Berzins - HSLayers NG as client for Layman

WARNING!

THIS PART IS ALSO TECHNICAL!

WHAT IS REST?

Representational state transfer

- is a software architectural style
- that allows requesting systems to access and manipulate web resources
- by using a uniform and predefined set of stateless operations.

WHAT IS REST?

In case of Layman

- **web resources** are **layers** and **maps**
- **operations** are standard HTTP methods
  - GET, POST, PATCH, and DELETE

LAYMAN REST API OPERATIONS

HTTP method	type of operation
POST	publish new resource
GET	get information about existing resource
PATCH	edit existing resource
DELETE	delete existing resource

LAYMAN REST API RESOURCES

- **Layer**
  - `/rest/<username>/layers`
  - `/rest/<username>/layers/<layername`
- **Map**
  - `/rest/<username>/maps`
  - `/rest/<username>/maps/<mapname>`
- detailed documentation

LAYMAN REST API RESOURCES

- `username`
  - unique identification of user (owner of resources) within Layman
  - depending on configuration, user's identity is either checked by OAuth2 provider, or it is not checked at all
    - user's identity is not checked in default demo configuration
  - it can not be changed

LAYMAN REST API RESOURCES

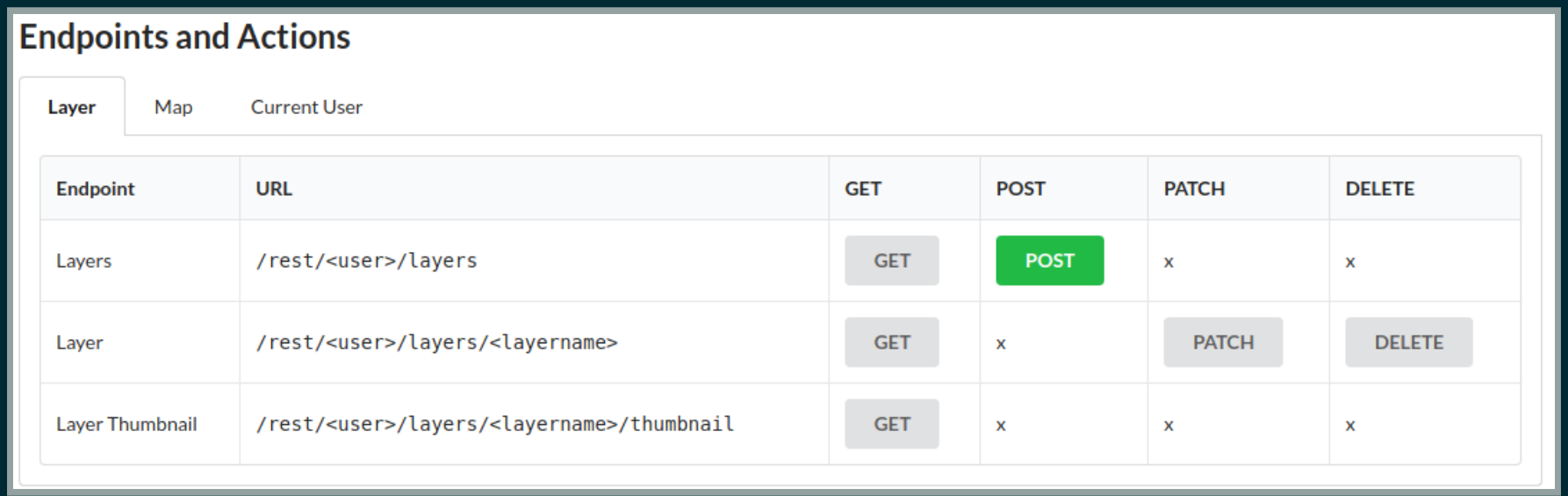
- `layername`, `mapname`
  - unique identification of layer (map) within all layers (maps) of given user
  - it is either chosen or automatically generated when the layer (map) is published
  - it can not be changed

PUBLISH NEW LAYER

1. Download some NaturalEarth data
    - Countries 1:10M
2. Unzip it
3. Visit http://<your IP address>/ in your web browser

PUBLISH NEW LAYER

4. Choose resource **Layer**, endpoint **Layers**, method **POST**



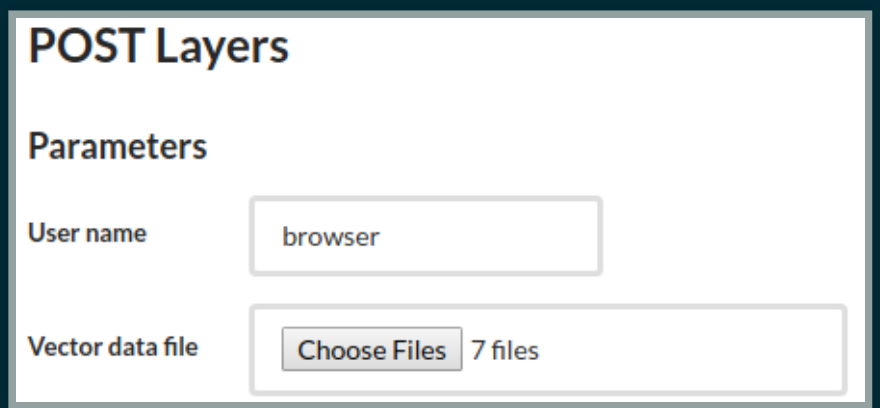
Endpoints and Actions

Layer | Map | Current User

Endpoint	URL	GET	POST	PATCH	DELETE
Layers	`/rest/<user>/layers`	GET	POST	x	x
Layer	`/rest/<user>/layers/<layername>`	GET	x	PATCH	DELETE
Layer Thumbnail	`/rest/<user>/layers/<layername>/thumbnail`	GET	x	x	x

PUBLISH NEW LAYER

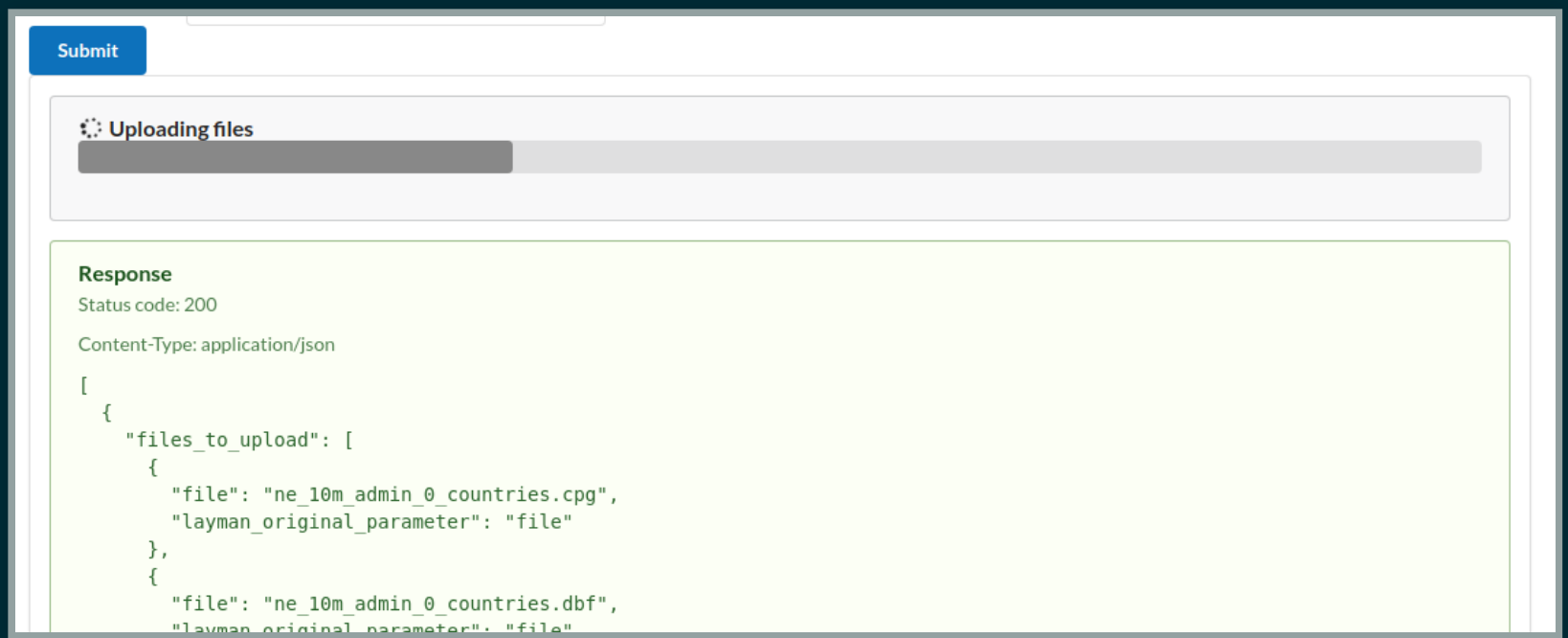
5. Choose all seven **ne_10m_admin_0_countries.*** files at **Vector data file** field



6. Click **Submit**

PUBLISH NEW LAYER

7. Wait till uploading is finished



PUBLISH NEW LAYER

8. In the meantime you can check the response


```
[
  {
    "files_to_upload": [
      {
        "file": "ne_10m_admin_0_countries.cpg",
        "layman_original_parameter": "file"
      },
      ...
    ],
    "name": "ne_10m_admin_0_countries",
    "url": "/rest/browser/layers/ne_10m_admin_0_countries",
    "uuid": "a8c6f6f4-1254-49fd-8223-5ed8f4fa185f"
  }
]
```

GET LIST OF ALL LAYERS

1. Choose resource **Layer**, endpoint **Layers**, method **GET**
2. Click **Submit**


```
[
  {
    "name": "ne_10m_admin_0_countries",
    "url": "/rest/browser/layers/ne_10m_admin_0_countries",
    "uuid": "a8c6f6f4-1254-49fd-8223-5ed8f4fa185f"
  }
]
```

GET INFORMATION ABOUT SINGLE LAYER

1. Choose resource **Layer**, endpoint **Layer**, method **GET**
2. Enter name of the layer to **Layer name** field
    - `ne_10m_admin_0_countries`
3. Click **Submit**

GET INFORMATION ABOUT SINGLE LAYER


```
{
  ...
  "metadata": {
    "csw_url": "http://micka:80/csw",
    "record_url": "http://104.248.252.23:3080/record/basic/m-a8c6
  },
  "name": "ne_10m_admin_0_countries",
  ...
  "wfs": {
    "url": "http://localhost:8600/geoserver/browser/ows"
  },
  "wms": {
    "url": "http://localhost:8600/geoserver/browser/ows"
  }
}
```

SET CORRECT PROXY BASE URL OF GEOSERVER

1. Visit http://<your IP address>/geoserver in your web browser
2. Login using username **admin**, password **geoserver**
3. In left menu, click on **Global** under **Settings**
4. Set **Proxy Base URL** to http://<your IP address>/geoserver/
5. Scroll down and click **Submit**

GET INFORMATION ABOUT SINGLE LAYER AGAIN

1. Choose resource **Layer**, endpoint **Layer**, method **GET**
2. Enter name of the layer to **Layer name** field
    - `ne_10m_admin_0_countries`
3. Click **Submit**

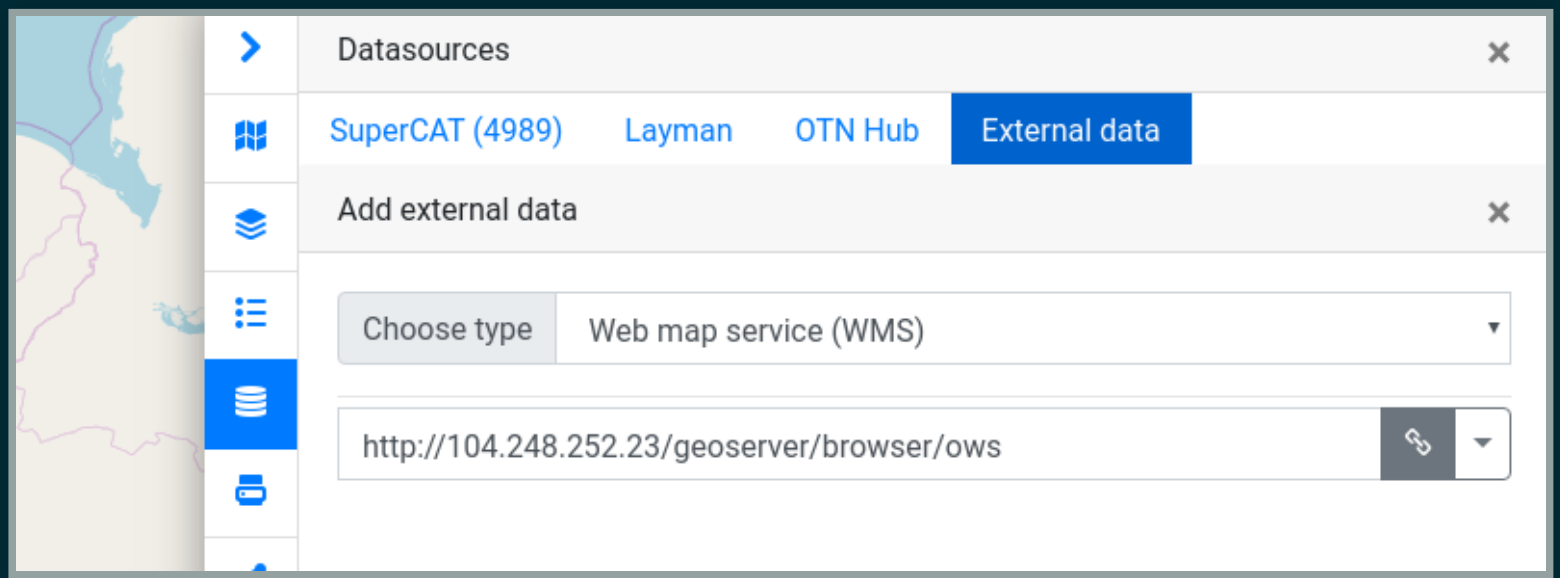
GET INFORMATION ABOUT SINGLE LAYER AGAIN


```
{
  ...
  "metadata": {
    "csw_url": "http://micka:80/csw",
    "record_url": "http://104.248.252.23:3080/record/basic/m-a8c6
  },
  "name": "ne_10m_admin_0_countries",
  ...
  "wfs": {
    "url": "http://104.248.252.23/geoserver/browser/ows"
  },
  "wms": {
    "url": "http://104.248.252.23/geoserver/browser/ows"
  }
}
```

VIEW LAYERS IN MAP

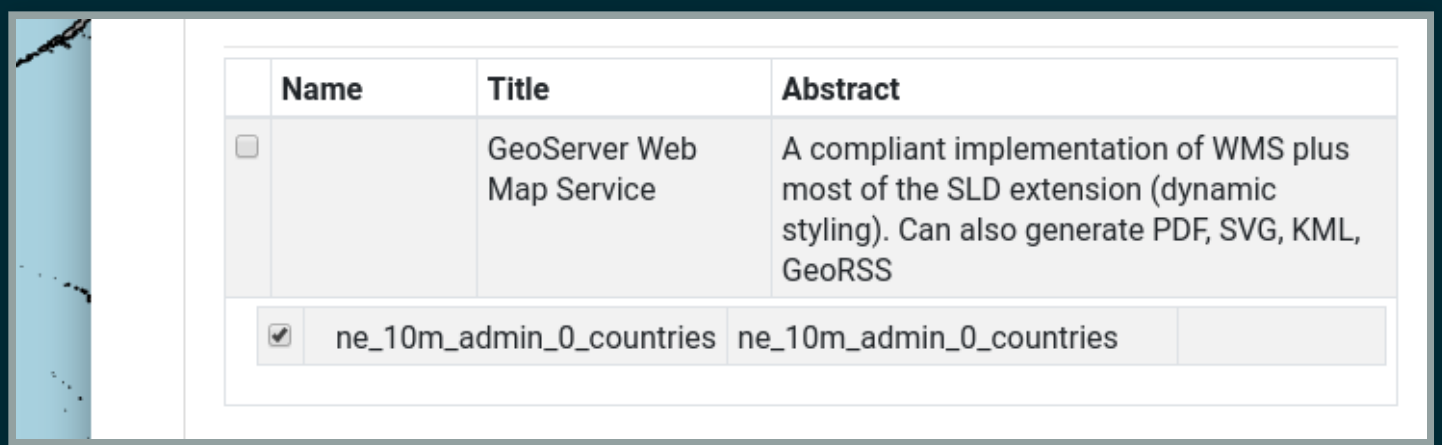
1. Visit https://ng.hslayers.org/examples/datasources/?hs_panel=datasource_selector
2. Set
    - **Choose type:** Web map service (URL)
    - **External data source (URL):** http://<your IP address>/geoserver/browser/ov
3. Click on gray chain icon at bottom right

VIEW LAYERS IN MAP



VIEW LAYERS IN MAP

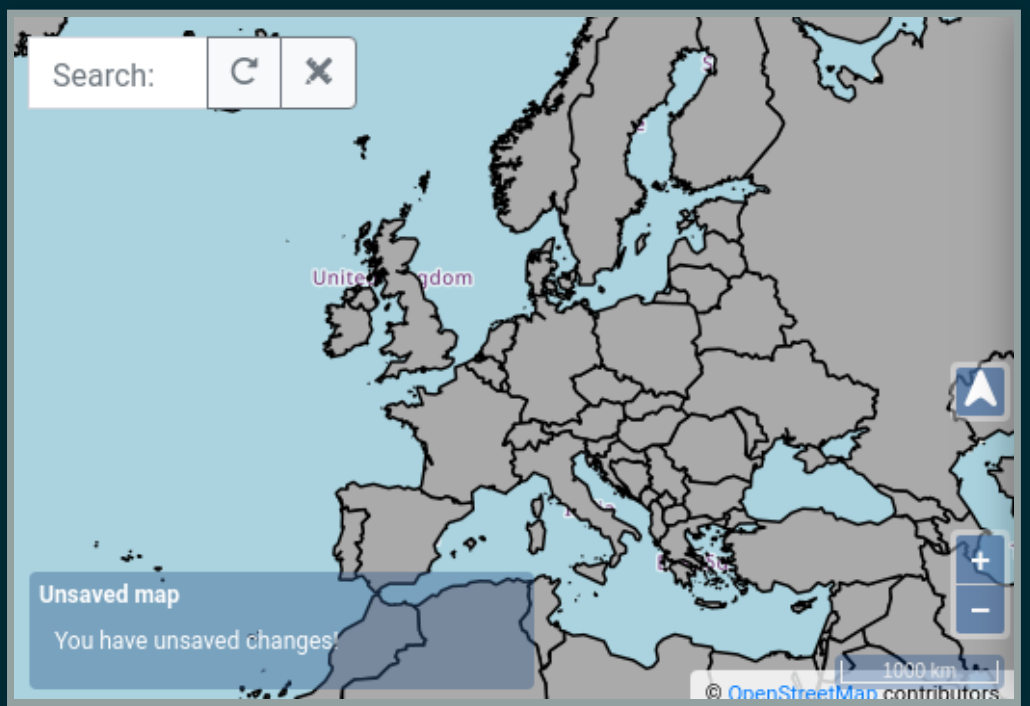
4. Check layers you want to see



5. Click on blue plus icon at bottom right

VIEW LAYERS IN MAP

7. Browse the map



AUTHENTICATION AND AUTHORIZATION

JIŘÍ KOZEL

Workshop 4

Big and Open Data and Innovative Hubs in Agriculture, Transport and Rural Development

Czech University of Life Sciences, Prague

January 29, 2020

WORKSHOP 4

https://github.com/jirik/layman-workshop

- Karel Charvát - Purpose of workshop
- Jiří Kozel - What is Layman and how it works
- Raitis Berzins - Map composition
- Jiří Kozel, Jiří Kvapil - How to install Layman in cloud
- Jiří Kozel - Layman API
- Jiří Kozel - Authentication and authorization
- Jiří Kozel - Interaction with Metadata (Micka)
- Jan Vrobel - QGIS plugins for accessing maps and map composition from server
- Jan Vrobel - QGIS plugin for Web data publishing using Layman
- Raitis Berzins - HSLayers NG as client for Layman

WARNING!

THIS PART IS ALSO TECHNICAL!

AUTHENTICATION

- process of obtaining and ensuring identity of user from incoming request to REST API
- performed by chain of zero or more authentication modules
- if no module succeeds, user is considered **anonymous**

AUTHENTICATION

Two basic options

- use **no authentication module**, so every user is considered as anonymous
  - default
- **OAuth2** module with Liferay as authorization server
  - detailed documentation

AUTHORIZATION

- process that decides if authenticated user has permissions to perform the request to REST API
- performed by single authorization module
- if the user does not have enough permissions, an "Unauthorised access" exception is raised

AUTHORIZATION

Types of operations	corresponding HTTP method
read	GET
write	POST, PUT, PATCH, DELETE

AUTHORIZATION

Two basic options

- **read everyone, write everyone**
  - everyone including anonymous user is able to read and write to anybody`s workspace
  - default
- **read everyone, write owner**
  - everyone including anonymous user is able to read anybody`s workspace, but only user that owns the workspace is able to write

CHECK CURRENT USER

1. Visit http://<your IP address>/ in your web browser
2. Choose **Current User**, endpoint **Current User**, method **GET**
3. Click **Submit**

CHECK CURRENT USER


```
{
  "authenticated": false,
  "claims": {
    "iss": "http://layman:8000/",
    "name": "Anonymous",
    "nickname": "Anonymous"
  }
}
```

MATERIALS

https://github.com/jirik/layman-workshop

LOG IN

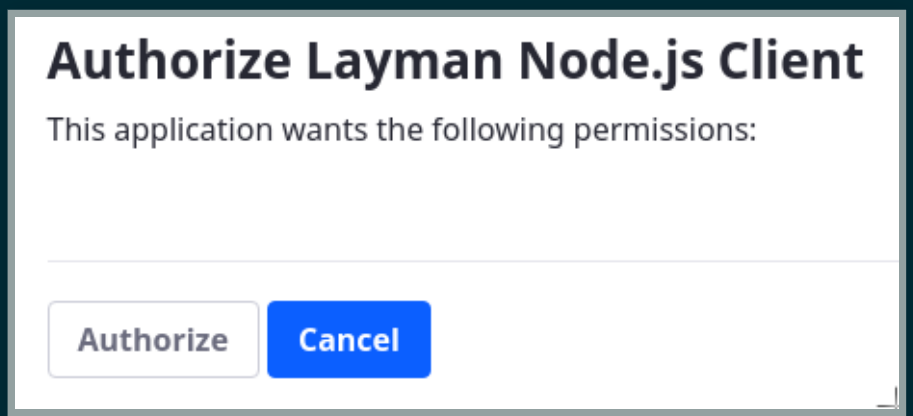
1. Visit http://<your IP address>/ in your web browser
2. Click **Log In**
    - you are forwarded to different domain where Liferay is running

LOG IN

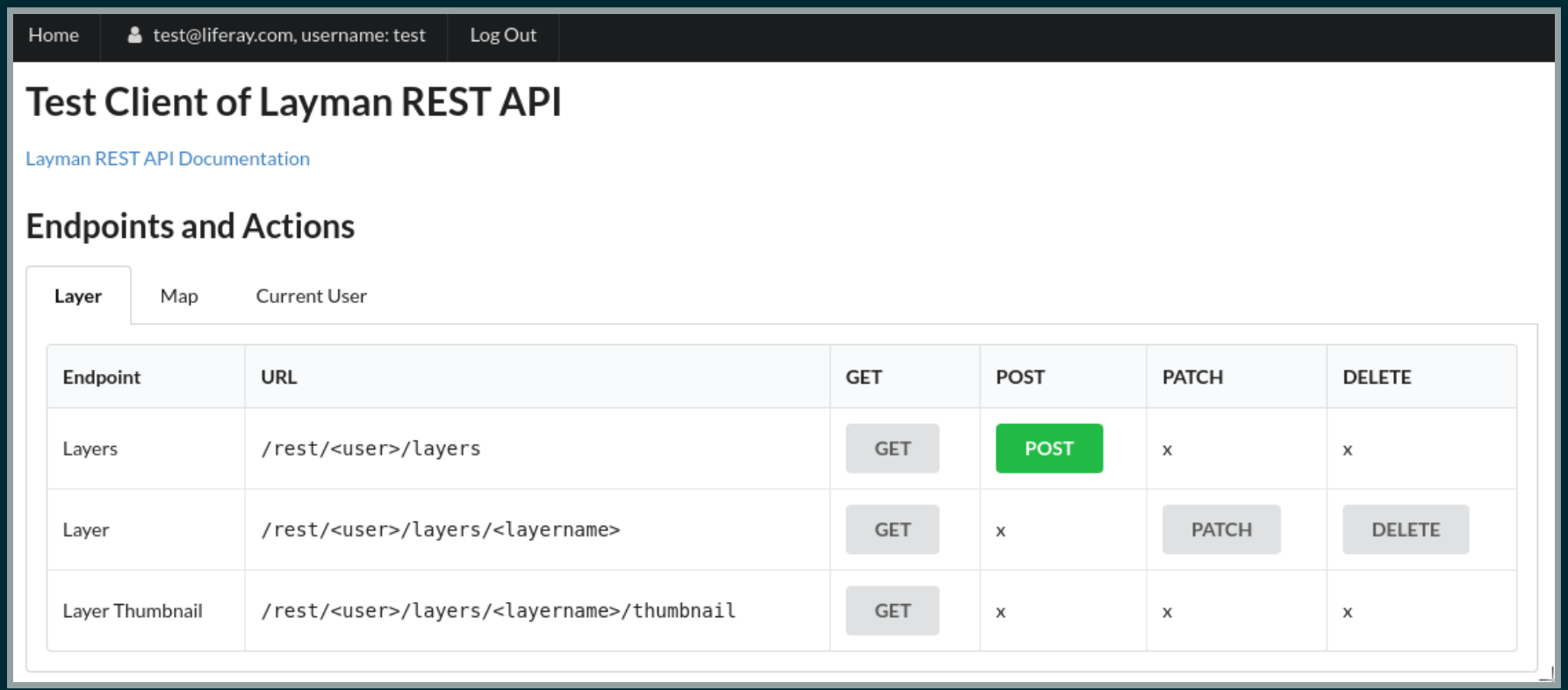
2. Set
    - Email Address: test@liferay.com
    - Password: test
    - Remember Me: checked
3. Click **Sign In**

LOG IN

5. Click **Authorize**



LOG IN


Home
test@liferay.com, username: test
Log Out
Test Client of Layman REST API
Layman REST API Documentation
Endpoints and Actions
Layer
Map
Current User
Endpoint
URL
GET
POST
PATCH
DELETE
Layers
/rest/<user>/layers
GET
POST
x
x
Layer
/rest/<user>/layers/<layername>
GET
x
PATCH
DELETE
Layer Thumbnail
/rest/<user>/layers/<layername>/thumbnail
GET
x
x
x

CHECK CURRENT USER AGAIN

1. Choose **Current User**, endpoint **Current User**, method **GET**
2. Click **Submit**

CHECK CURRENT USER AGAIN


```
{
  "authenticated": true,
  "claims": {
    "email": "test@liferay.com",
    "email_verified": true,
    "family_name": "Test",
    "given_name": "Test",
    "iss": "http://167.172.174.152:8082/o/oauth2/authorize",
    "middle_name": "",
    "name": "Test Test",
    "preferred_username": "test",
    "sub": "20139",
    "updated_at": 1580279122416
  },
  "username": "test"
}
```

INTERACTION WITH METADATA (MICKA)

JIŘÍ KOZEL

Workshop 4

Big and Open Data and Innovative Hubs in Agriculture, Transport and Rural Development

Czech University of Life Sciences, Prague

January 29, 2020

WORKSHOP 4

https://github.com/jirik/layman-workshop

- Karel Charvát - Purpose of workshop
- Jiří Kozel - What is Layman and how it works
- Raitis Berzins - Map composition
- Jiří Kozel, Jiří Kvapil - How to install Layman in cloud
- Jiří Kozel - Layman API
- Jiří Kozel - Authentication and authorization
- Jiří Kozel - Interaction with Metadata (Micka)
- Jan Vrobel - QGIS plugins for accessing maps and map composition from server
- Jan Vrobel - QGIS plugin for Web data publishing using Layman
- Raitis Berzins - HSLayers NG as client for Layman

WARNING!

THIS PART IS ALSO LITTLE BIT TECHNICAL!

METADATA

- Layman is able to publish **partial metadata** records to OGC Catalogue Service Micka
- Records are partial because Layman does not know all metadata properties
- currently, only **Layer** metadata are published, and only on POST request
    - full support in May 2020

GET INFORMATION ABOUT SINGLE LAYER

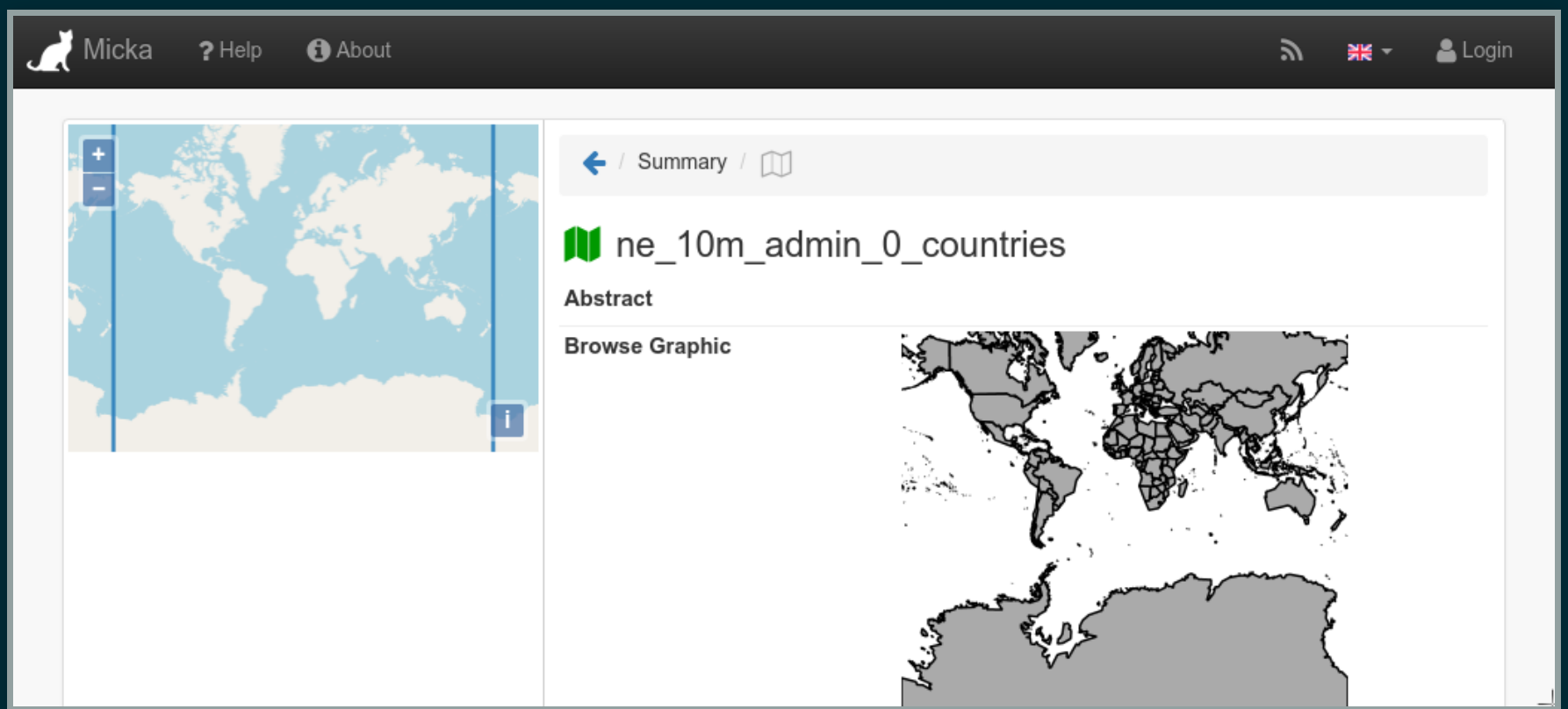
1. Choose resource **Layer**, endpoint **Layer**, method **GET**
2. Enter name of the layer to **Layer name** field
    - `ne_10m_admin_0_countries`
3. Click **Submit**

GET INFORMATION ABOUT SINGLE LAYER

```
{
  ...
  "metadata": {
    "csw_url": "http://micka:80/csw",
    "record_url": "http://104.248.252.23:3080/record/basic/m-a8c6
  },
  "name": "ne_10m_admin_0_countries",
  ...
}
```

SHOW METADATA RECORD

1. Copy `metadata.record_url` value and open this address in the browser



EDIT METADATA RECORD IN MICKA

5. Login using name **editor**, password **editor**
6. Edit metadata record.