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

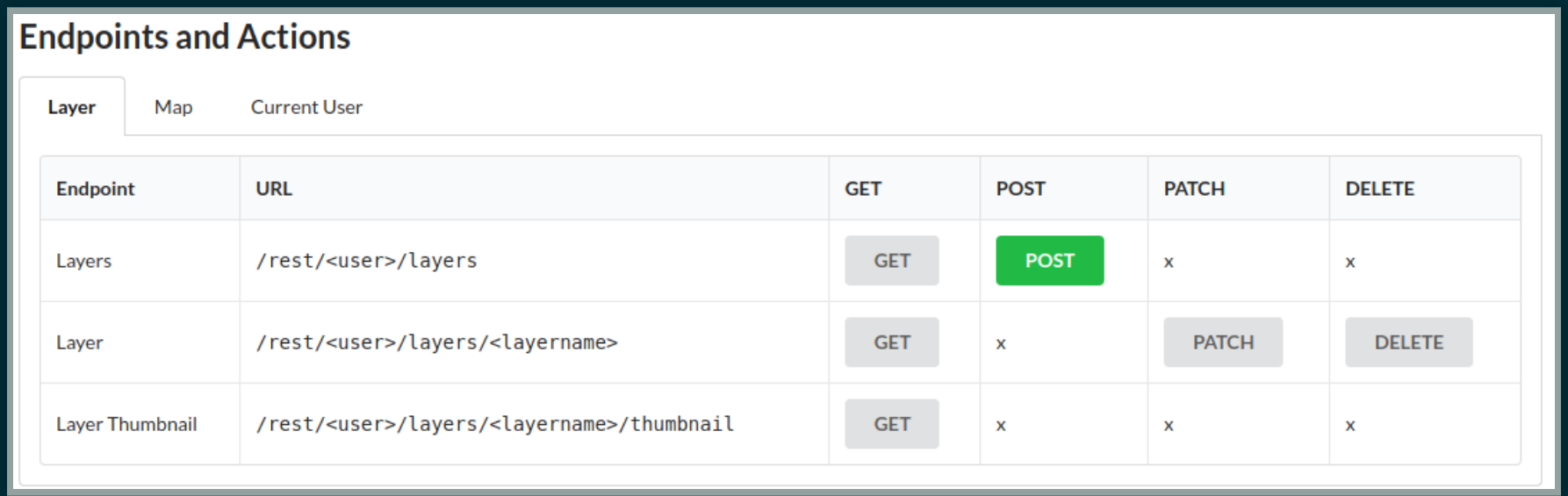
- `layername`, `mapname`
  - unique identification of layer (map) within all layers (maps) of given user
  - it is either chosen or automatically generated when the layer (map) is published
  - it can not be changed

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**



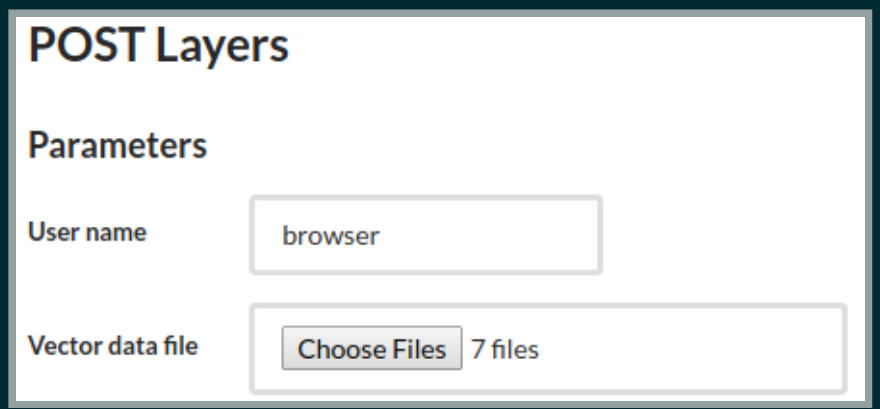
Endpoints and Actions

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

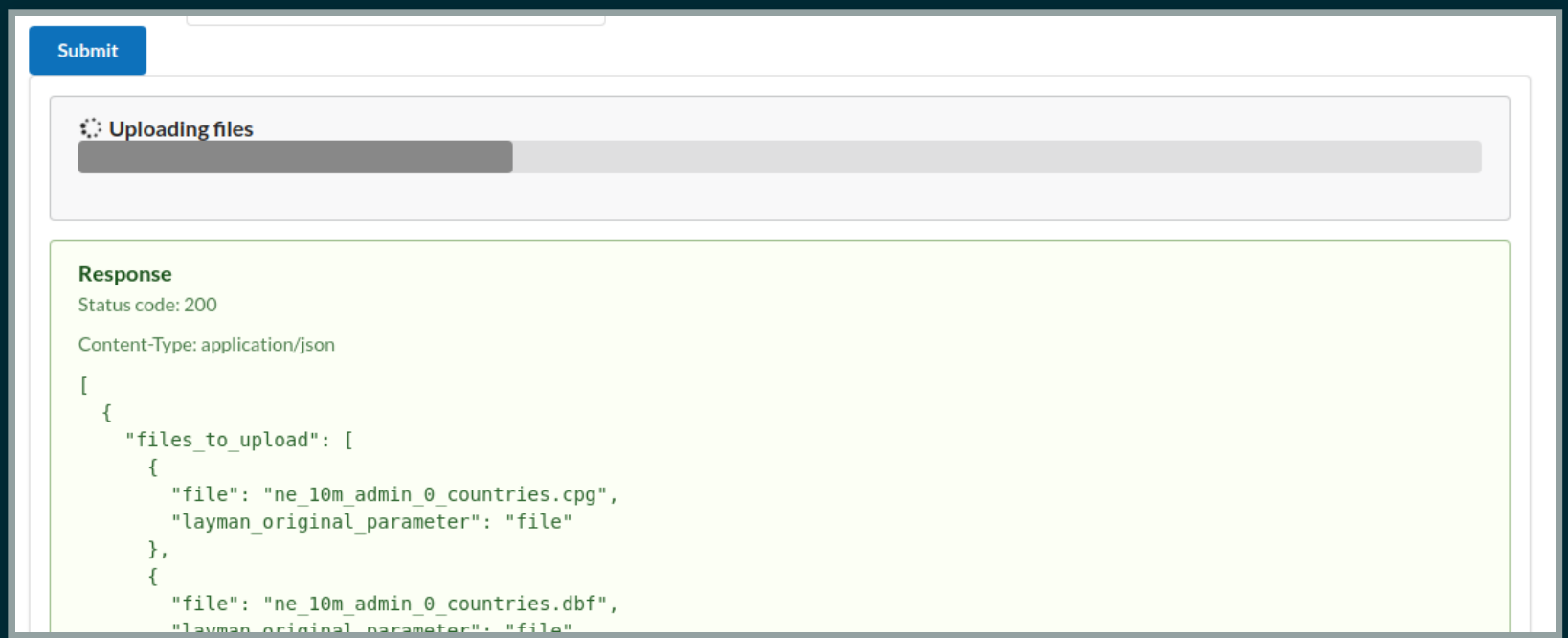
5. Choose all seven **ne_10m_admin_0_countries.*** files at **Vector data file** field



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**

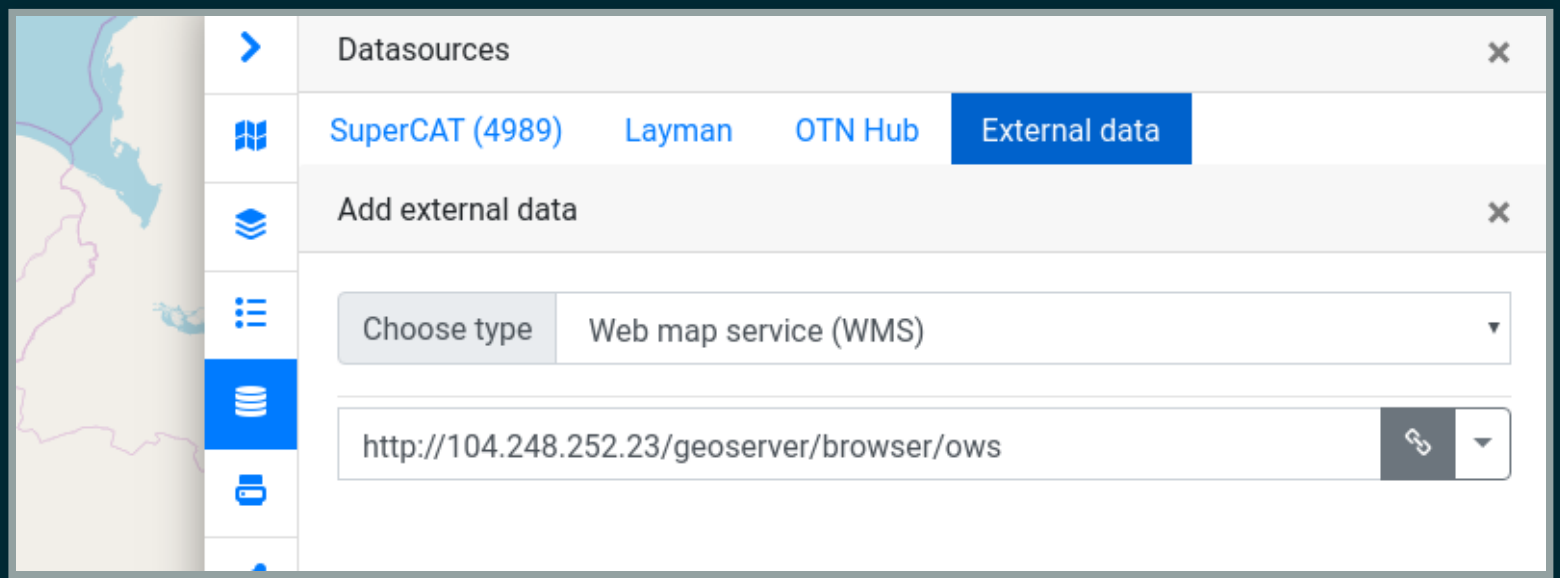
GET INFORMATION ABOUT SINGLE LAYER AGAIN


```
{
  ...
  "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

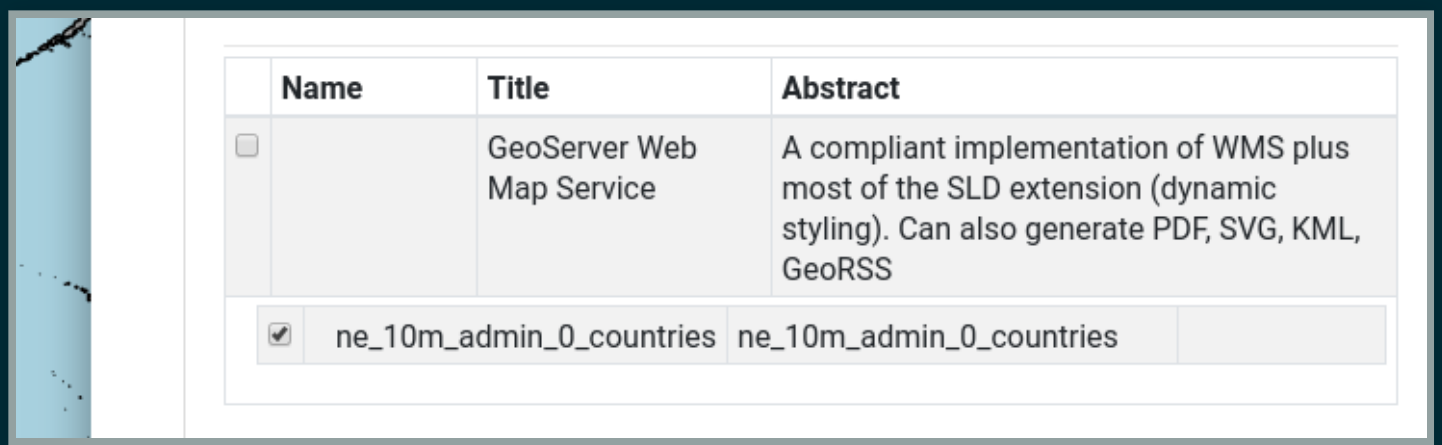
1. Visit https://ng.hslayers.org/examples/datasources/?hs_panel=datasource_selector
2. Set
    - **Choose type:** Web map service (URL)
    - **External data source (URL):** http://<your IP address>/geoserver/browser/ov
3. Click on gray chain icon at bottom right

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