



## Easily publish your QGIS projects on the web with QWC2

Sandro Mani  
Sourcepole AG, Zürich  
[www.sourcepole.ch](http://www.sourcepole.ch)





# Outline

- › Introduction to QWC2
- › Introduction to qwc-services
- › News from the project



# QGIS Web Client 2

- **QWC2 is the official web application for publishing QGIS projects on the web**
  - First public release in 2016
  - Modern, responsive UI
  - Focus on usability
  - Plugin architecture, easily extensible
  - State of the art technologies: ReactJS, OpenLayers 6



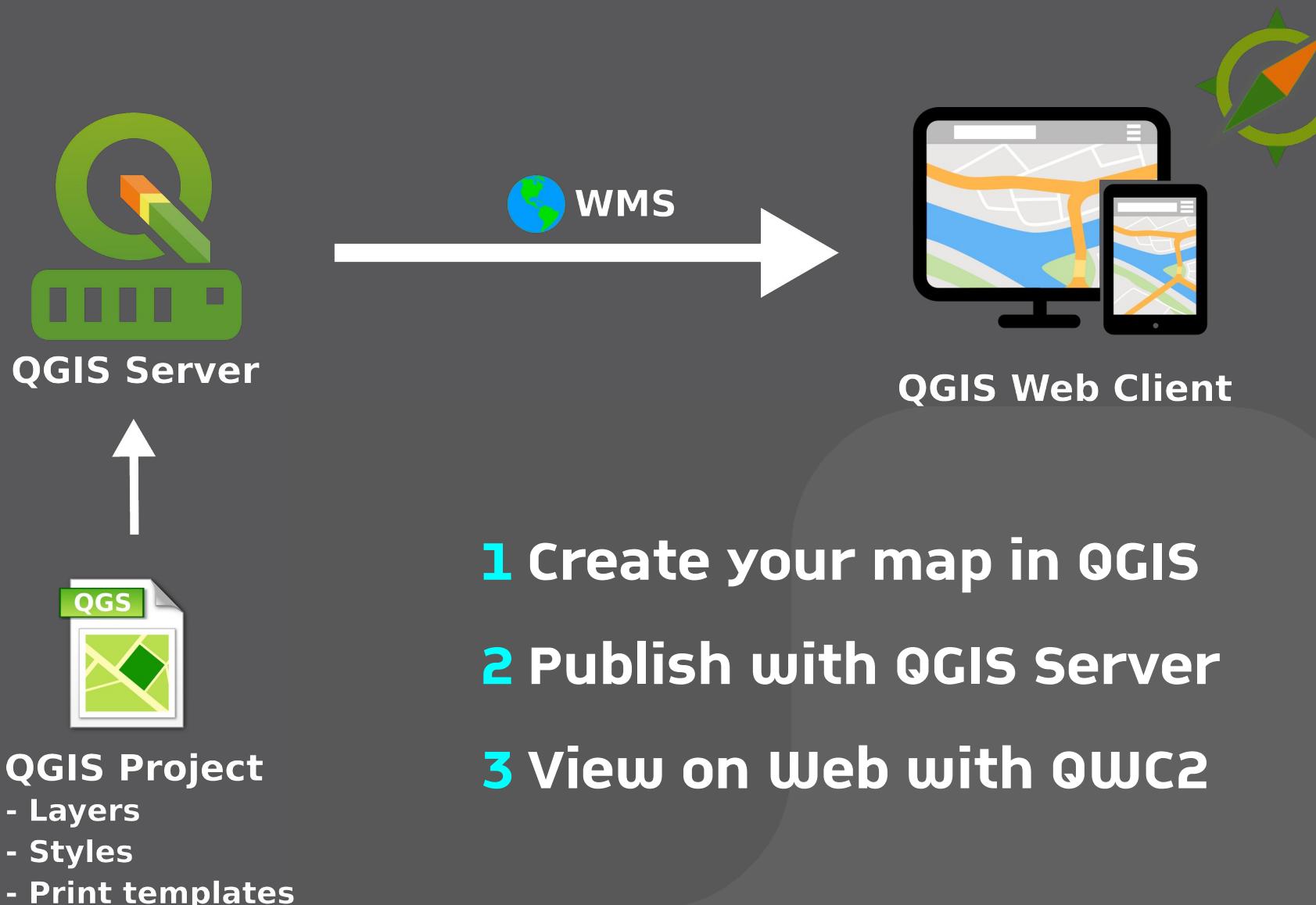


# OGIS Web Client 2

The screenshot displays the OGIS Web Client 2 interface, featuring a map of Europe with satellite imagery and administrative boundaries. The top navigation bar includes a QWC Demo logo, a search bar, and various map tools. A prominent yellow button labeled "Toggle API demo" is visible. On the left, a sidebar provides access to themes, layers, layer catalogs, sharing options, bookmarks, map tools, printing, help, and login. The main map area shows countries outlined in orange. To the right, a "Layers & Legend" panel is open, listing several layers: Demo (checked), Edit Demo (checked), Demo Points (checked), Demo Lines (checked), Demo Polygons (unchecked), Geographic lines (checked), Country names (checked), States and Provinces (checked), Countries (checked), Show layer map tips (unchecked), Compare top layer (unchecked), and Import layer (button). The bottom of the screen shows coordinate information (1929481 5892447, WGS 84 / Pseudo Mercator) and a scale (1 : 25000000). A "Demo attribution" link is also present.



# QGIS Web Client 2 · Overview





# QGIS Web Client 2 · Core features

- Theme switcher (QGIS projects)
- Layertree
- Feature Info
- Search with configurable providers
- Measurement tools
- Redlining (sketching)
- Permalink sharing
- Geolocation
- PDF printing
- Map compare
- Importing external layers (WMS, WFS, WMTS, KML)
- Javascript API



# QWC2 and qwc-services

- **QWC2 can be used as a stand-alone viewer on top of QGIS Server alone**
  - Static JS/HTML/CSS application
  - <https://github.com/qgis/qwc2-demo-app>
- **Or it can be extended with additional services**

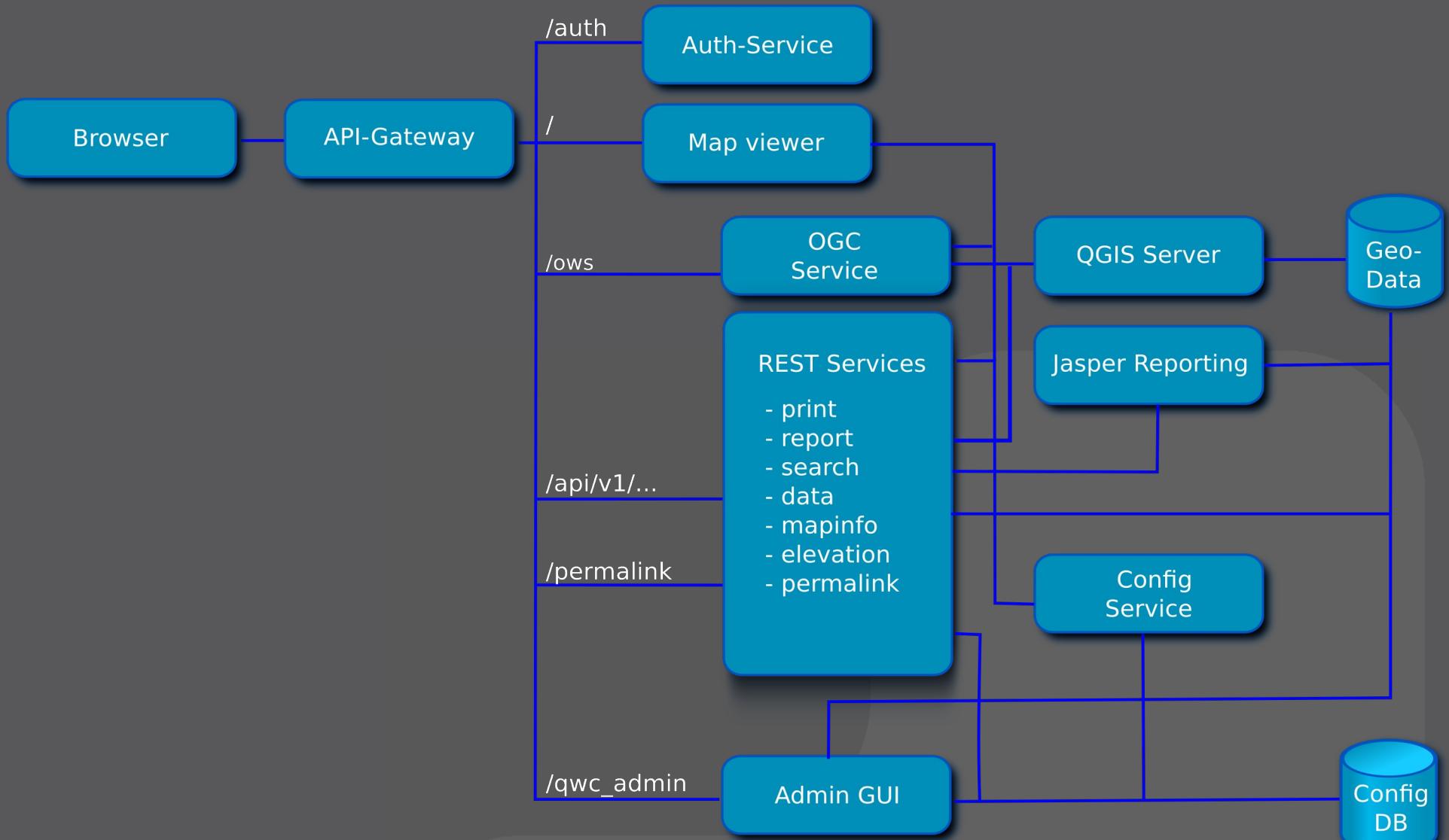
‣ **qwc-services**



- **Ecosystem of microservices for extending QWC2**
  - Collection of loosely-coupled services
  - Communication between services over HTTP/REST
  - Mostly Python-Flask services
- **Easily deployed as Docker containers**
  - <https://github.com/qwc-services/qwc-docker>
- **Or custom setup, i.e. via Apache + mod wsgi**



# qwc-services • Architecture





- **Functionalities available by extending QWC2 with qwc-services**
  - User and resource management
  - Authentication (DB, LDAP, ...)
  - Fulltext search
  - Editing
  - Compact permalinks, user bookmarks
  - Height profile
  - Custom feature-info templates
  - Reporting (Jasper)
  - Mapinfo popup



# qwc-services • Core services

- **QWC OGC service**

- Forwards requests to QGIS server enforcing permissions

- **QWC map viewer**

- Enforces permissions to the QWC2 themes and viewer configuration

- **QWC Admin GUI**

- Backend for managing users and resource permissions (maps, layers, edit datasets, ...)

- **QWC Config Generator**

- Combines theme capabilities and permissions into static service configuration files



- **Modular authentication services**
  - Issue JWT tokens
- **QWC DB Auth**
  - Authentication against user DB
- **QWC LDAP Auth**
  - Authentication with LDAP/Active Directory
- **and others (i.e. Kerberos, SAML)**

- **QWC Data Service**
  - Allows read/write access to configured datasets in a PostGIS DB
  - Backend for the QWC2 editing functionality
- **QWC Permalink Service**
  - Stores and resolves compact permalinks
- **QWC Fulltext Search Service**
  - Faceted fulltext search with Apache Solr
- **QWC MapInfo Service**
  - Personalized, layer-independent information (context-menu)



## ‣ Docker

- One service per container
- Execution in controlled environment
- Easy to create and manage
- Distribution over centralized registry
- <https://www.docker.com/>

## ‣ Kubernetes

- System for automating the deployment, scaling and management of container applications
- <https://kubernetes.io/>



## News from the project



## OWC2 news - Snapping

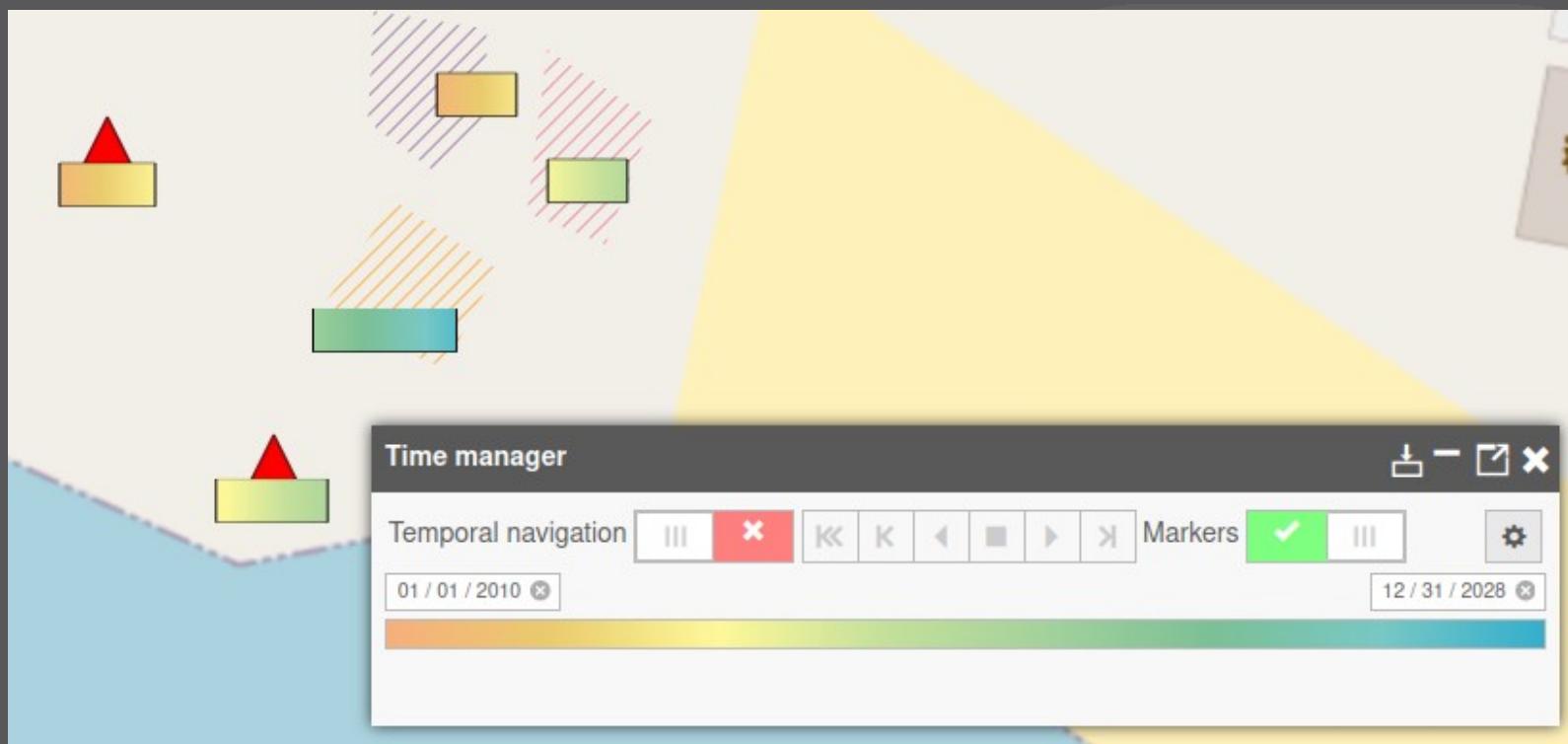
- Snap to theme layers when editing, measuring and sketching
- Individually configurable for each tool and layer
- Loads snap geometries over GetFeatureInfo
- Interactive no-delay snapping





# OWC2 news · Time manager

- Time slider for layers with time dimensions (WMS-T)
- Uses time dimensions configured in QGIS layer properties





# OWC2 news · Attribute table

- › View and edit attributes of geometry and geometryless datasets
- › Add / edit / remove records
- › Zoom
- › Filtering

Attribute Table

Layer Fluffy animals

C + ⌂ ⌂ ⌂

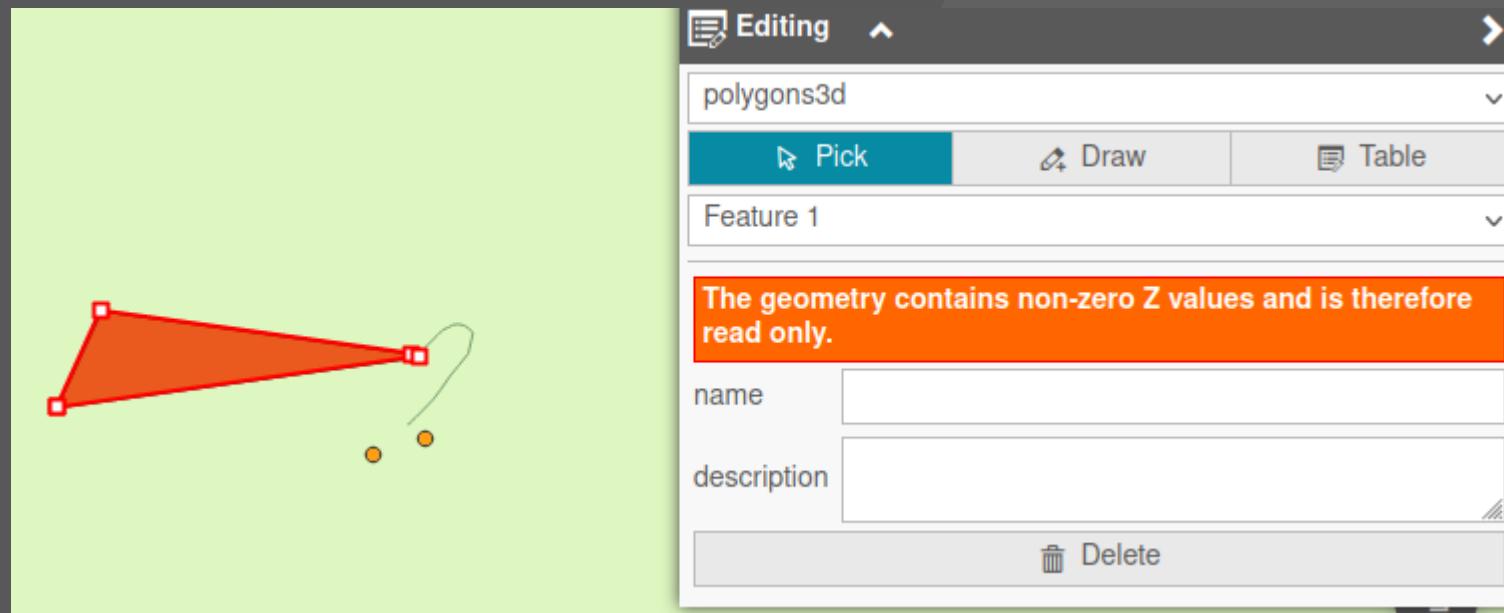
	id	name	age	birthdate	size	hungry	sleepy	remarks	Animal	picture
<input type="checkbox"/>	2	Minni	0	04 / 25 / 2011	5	<input type="checkbox"/>	<input type="checkbox"/>		Cat	<a href="#">minni.jpg</a>
<input type="checkbox"/>	1	Mily	2	02 / 13 / 2019	6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Select	<a href="#">11.jpg</a>
<input type="checkbox"/>	3	asd	1	mm / dd / yyyy	6	<input type="checkbox"/>	<input type="checkbox"/>		Cat	<input type="button" value="Browse..."/>

1 50 per page id ~



# OWC2 news · Editing · Z geometries

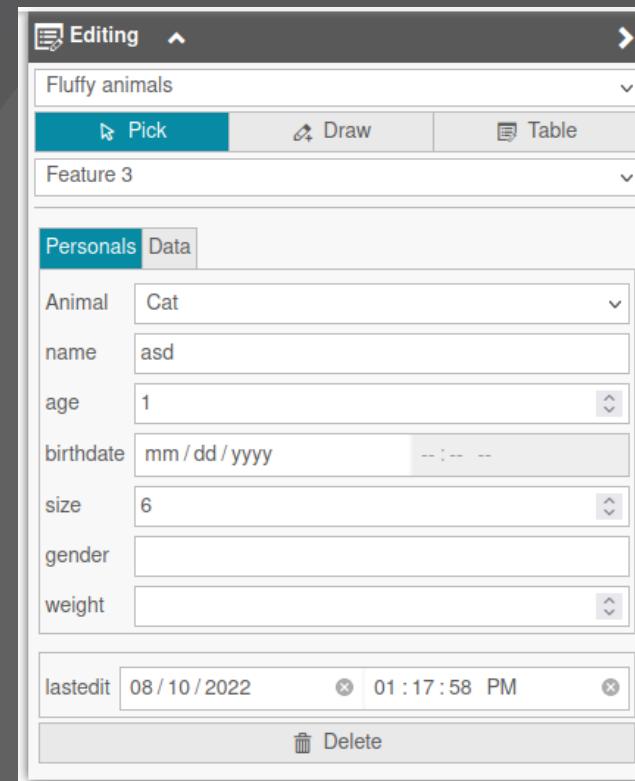
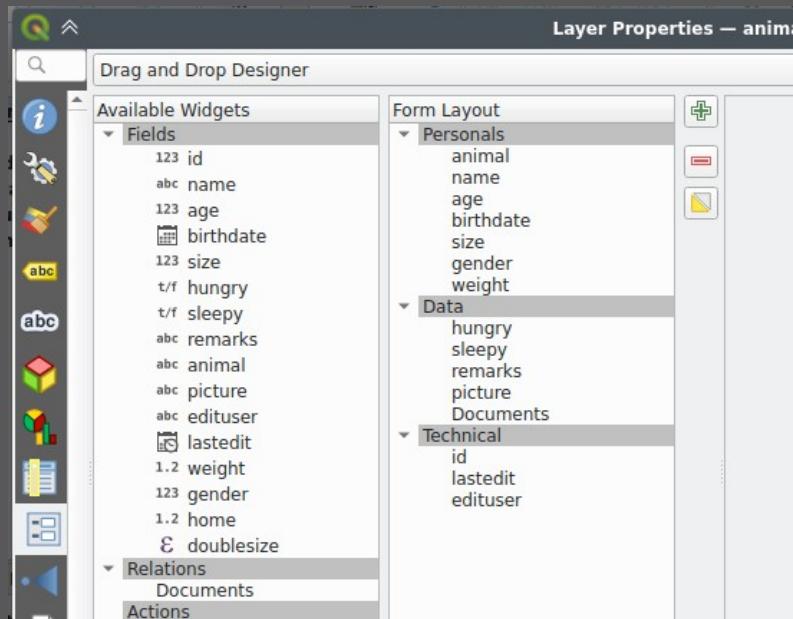
- Allow editing Z geometries if all Z values are zero
- Allow editing only attributes if geometry contains non-zero Z values





# OWC2 news · Attribute forms

- **qwc-config-generator translates the QGIS form configuration (autogenerated and DnD) to Qt UI forms, which are read by OWC2**
- **Manually provided UI forms are also passed on to OWC2**





## OWC2 news · Feature form

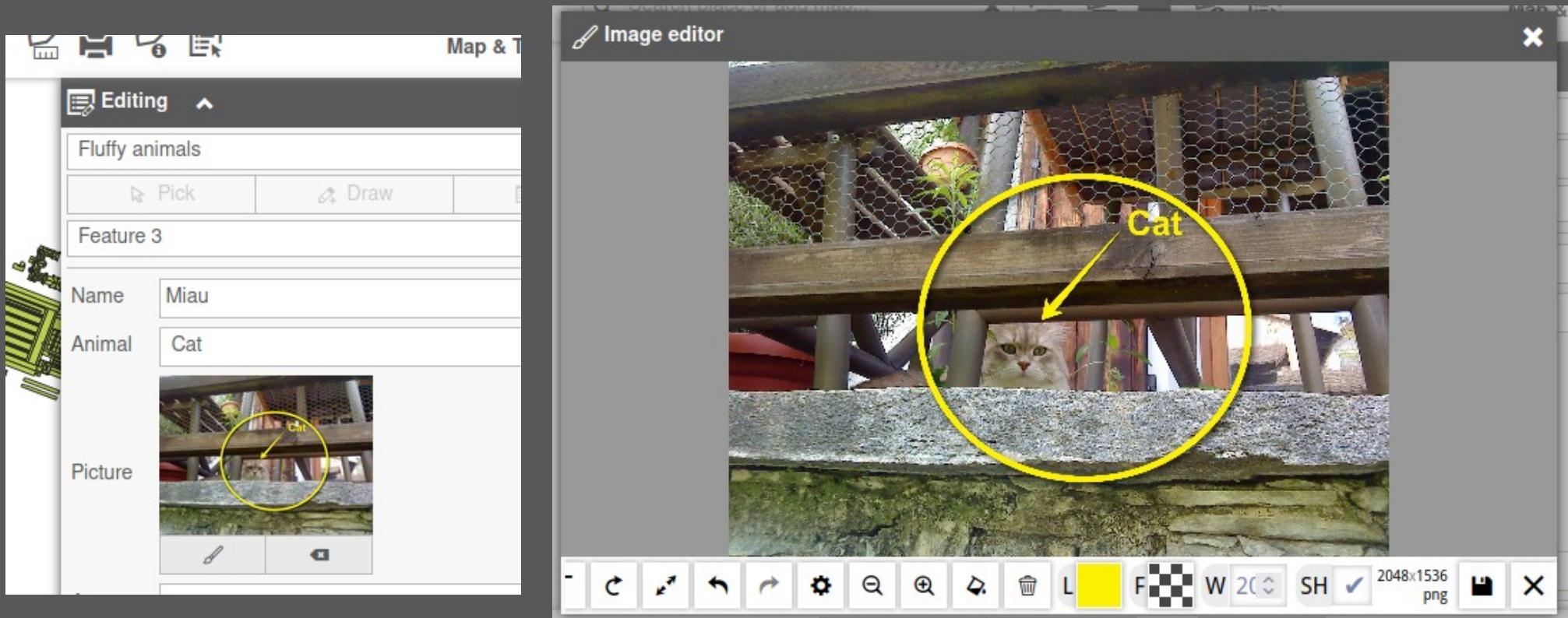
- Similar to Identify tool, but displays result in an editable attribute form
- Queries features over the data-service rather than over GetFeatureInfo
- Uses edit forms
- Can be used as default Identify tool

The screenshot shows the 'Feature form' dialog in QGIS Web Client 2. At the top, there's a toolbar with icons for zoom, search, and print. The main title bar says 'Feature form'. Below it is a dropdown menu showing 'Fluffy animals: Minni', 'Fluffy animals: Mily', and 'Fluffy animals: Minni', with the last one highlighted. The main area has a preview image of a cat. Below the image are several form fields:

- Picture: A preview image of a cat.
- Age: A numeric input field containing '0'.
- Birthdate: A date input field containing '04/25/2011'.
- Gender: Radio buttons for 'Male' and 'Female', with 'Male' selected.
- Size: A slider input field with a value of '0' and a unit of 'meters'.
- Flags: Two checkboxes labeled 'Hungry' and 'Sleepy', neither of which is checked.

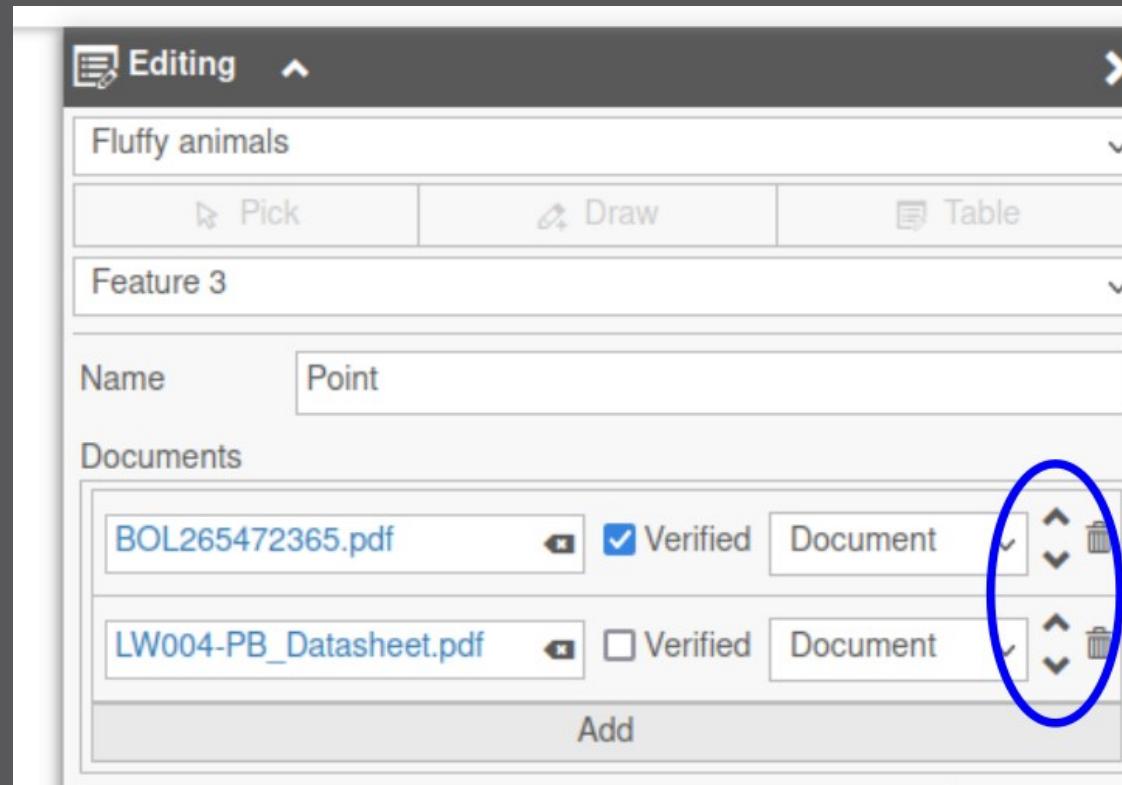


- Integrated image editor for editing uploaded images



- **Sortable 1:N relations**

- Only with manually provided Qt Designer forms
- QWidget name nrel\_<reldataset>\_<fk>\_<sortfield>



- Nested feature forms (joins and 1:N relations)
  - QPushButton with name  
`featurelink_<linkdataset>_<fieldname>`  
`featurelink_<linkdataset>_<reltable>_<fieldname>`
  - Allows editing 1:N relation records in nested form

**Revisions**

Identifier:	Rev01
NBIdent:	GR0100000001
Description:	Descr
Comment:	
State:	in_Bearbeitung
Opening date:	08 / 22 / 2022
Date of entry into force:	mm / dd / yyyy
Associated fixpoints	<div style="border: 1px solid #ccc; padding: 5px; border-radius: 5px; margin-bottom: 5px;">                     12765210                 </div> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>Pick</span> <span>Create</span> </div>

**Revisions**

Id	Fixpoint nr: 12765210
N	<a href="#">Identification and geometry</a>
D	<a href="#">Description</a>
C	<a href="#">Images</a>
S	<a href="#">Administration</a>
O	
D	
A	

**Type:** LFP2

**NBIdent:** CH03000012

**Position [m]:** E: 2770639.1  
N: 1137335.4

TSP

**Precision position [m]:** 0.030

**Reliability position:** ja

**Linked revision:** asd

**CN:** 1276

**Coordinate measurement:** gemessen

**Altitude [m]:** 1359.210

**Precision altitude [m]:** 0.045

**Reliability altitude:** ja

[Disable fixpoint](#)

[Close](#)



## ‣ Tweak edit form before display

```
registerFormPreprocessor("<dataset>", (form, feature, callback) => {  
    form.fields.<field>.property.enabled = false;  
});
```

## ‣ External fields

- Fields computed from external data sources (i.e. Country from feature point)
- QWidget with name ext\_<fieldname>

```
form.externalFields.<fieldname> = (<ReactFragment>);
```

## ‣ Buttons with custom triggers

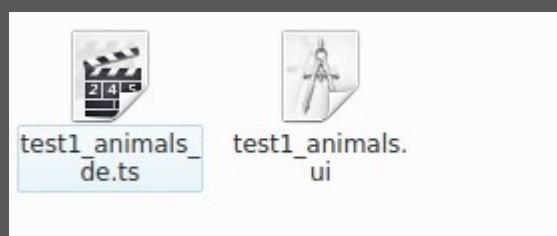
- Bind custom functions to buttons in forms
- QPushButton with name btn\_<buttonname>

```
form.buttons.<buttonname>.onClick = () => { ... };
```



## ‣ Localized edit forms

- Create translations of form UIs with Qt Linguist
- Helper script at qwc2/scripts/translateui.sh
- qwc-map-viewer will return a localized form if available



The screenshot shows the QGIS Web Client interface. On the left, there's a file browser window displaying two files: 'test1\_animals\_de.ts' and 'test1\_animals.ui'. The 'test1\_animals\_de.ts' file has a blue selection highlight. On the right, a detailed view of a form titled 'Fluffy animals' is shown. The form contains the following fields:

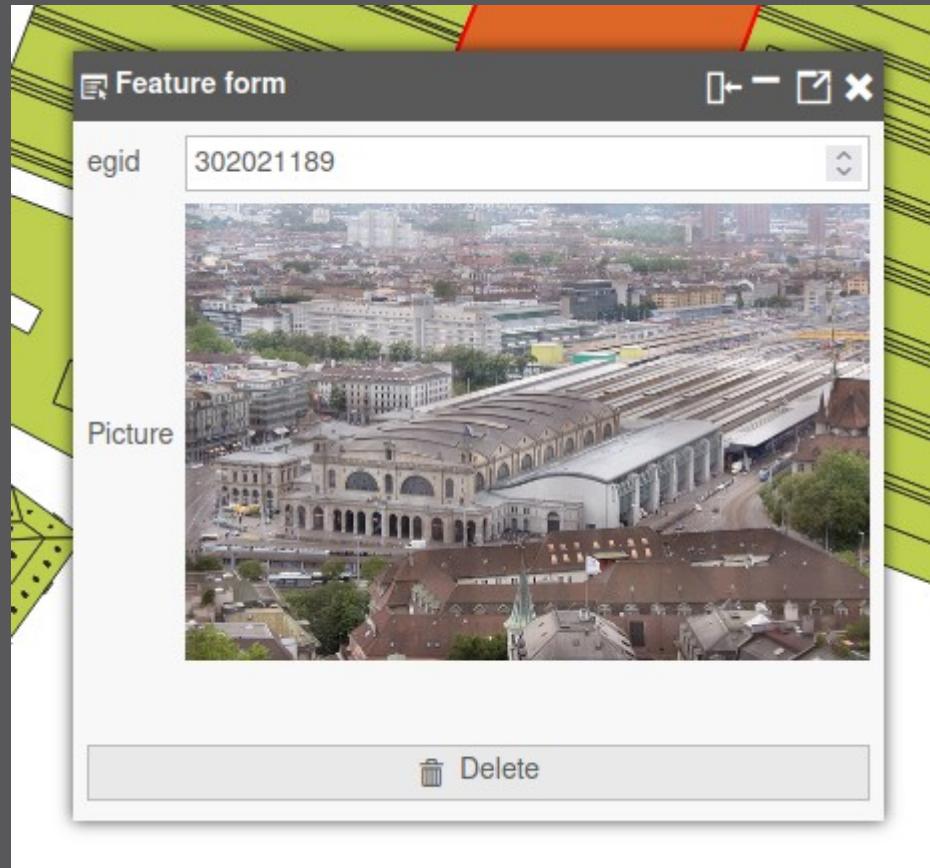
- Name: asd
- Tier: Cat
- Bild: Browse... No file selected.
- Alter: 1

A 'Löschen' (Delete) button is located at the bottom of the form. Below the form, the QGIS Web Client's developer tools are visible, showing the network tab with a request to the URL: `http://localhost:8088/assets/forms/autogen/test1_animals.ui?v=1661329315&lang=de-CH`. The network tab also shows various headers like XHR, Fonts, Images, Media, WS, Other, and a Disable Cache checkbox.



# OWC2 news · Forms · Inline images

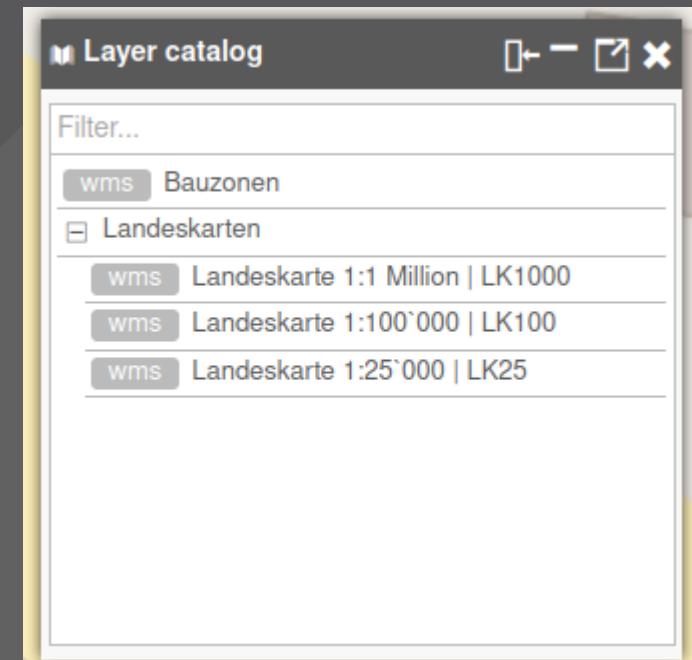
- Display inline image for field containing an URL
  - QLabel with name img\_<fieldname>



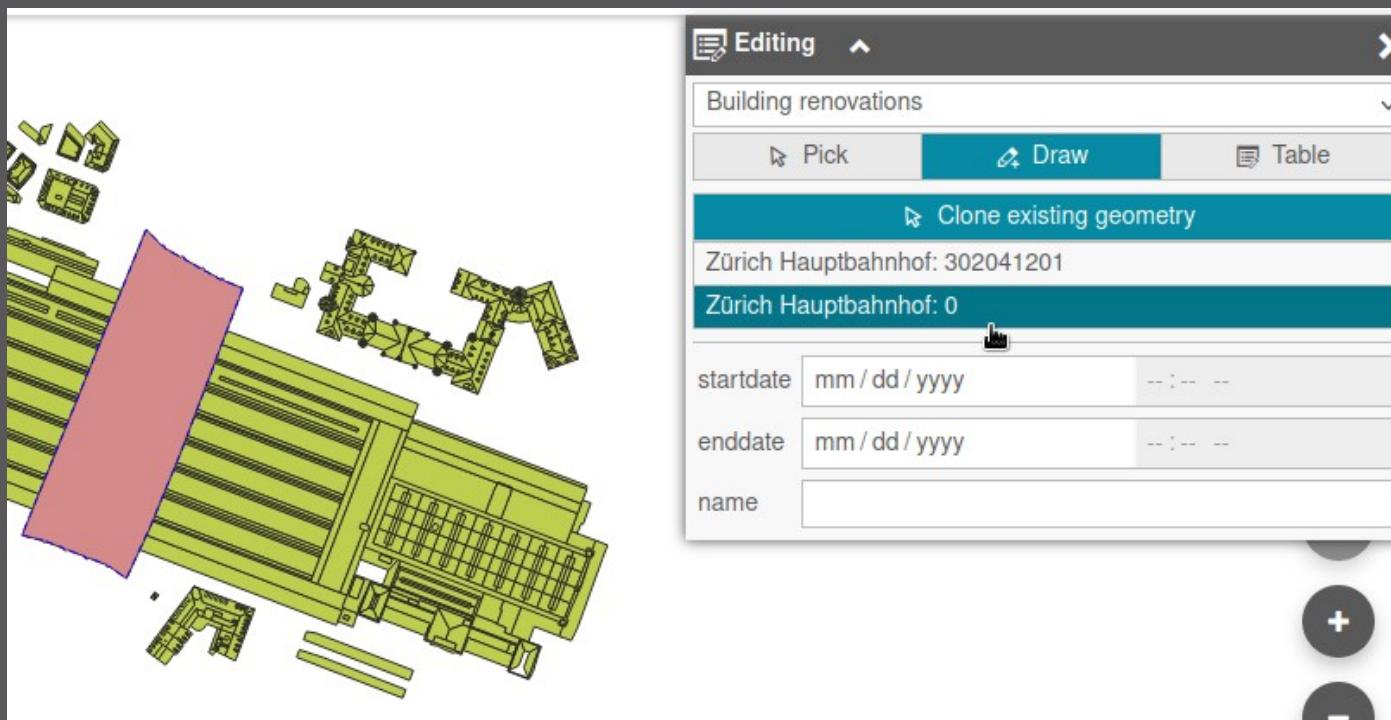


- Catalog for external layers from static configuration

```
{  
  "catalog": [  
    {  
      "title": "Bauzonen",  
      "resource": "wms:http://wms.geo.admin.ch#ch.are.bauzonen"  
    },  
    {  
      "title": "Landeskarten",  
      "sublayers": [  
        {  
          "title": "Landeskarte 1:1 Million | LK1000",  
          "resource": "wms:http://wms.geo.admin.ch#ch.swisstopo.pixelkarte-farbe-pk1000.noscale"  
        },  
        {  
          "title": "Landeskarte 1:100'000 | LK100",  
          "resource": "wms:http://wms.geo.admin.ch#ch.swisstopo.pixelkarte-farbe-pk100.noscale"  
        },  
        {  
          "title": "Landeskarte 1:25'000 | LK25",  
          "resource": "wms:http://wms.geo.admin.ch#ch.swisstopo.pixelkarte-farbe-pk25.noscale"  
        }  
      ]  
    }  
  ]  
}
```

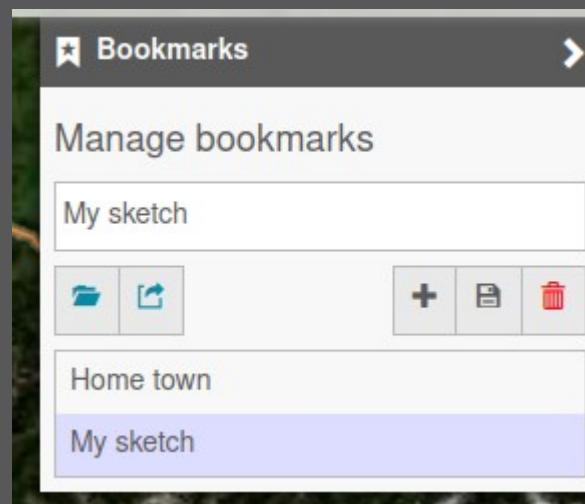


## ‣ Clone geometry for new feature





- Persistent permalinks stored per-user





# References

- › **Homepage**
  - › <https://qwc-services.github.io/>
- › **Main repositories**
  - › <https://github.com/qgis/qwc2-demo-app>
  - › <https://github.com/qwc-services/qwc-docker>
- › **Try it out**
  - › <https://qwc2.sourcepole.ch>
  - › <https://qgiscloud.com>
  - › <https://map.geo.gi.ch>
  - › <https://geo.so.ch/map/>



FOSS4G 2022

# Thank you!



Sandro Mani