

Accessing Spatial Data using the OGC API Features Interface

[Square brackets enclose RAW materials such as raw ideas, text from previous versions, cited content from other sources etc. that should be used to refine this part of the document.]

1. Overview

In this open educational resource (OER) you will learn how to access geospatial data using web services that implement the OGC API Features interface.

- We will use a standard browser to interact with the human readable parts of the service.
- We will use QGIS to demonstrate that off-the-shelf software is capable of interacting with this type of web service in plug-and-play mode.
- We will use Docker, Python, and Jupyter Notebooks to show how you can write programming code that interacts with this type of web services.

The tutorial is structured as follows:

1. Overview
2. Background
 - Feature Data and Feature Services
 - OGC-API series of interface specifications
 - OGC-API Features interface
3. Practical exercises
 - Use case: accessing INSPIRE Feature Data on Protected Sites
 - Downloading and installing the software
 - Using a standard web browser to interact with an OGC-API Features web service
 - Using QGIS to Access feature data via OGC API Features
 - Using python to interact with an OGC-API Features web service
4. Discussion and Wrap up

This tutorial is designed for students and professionals who want to improve their understanding of Spatial Information Infrastructures. We assume that you have some basic knowledge about geospatial data, QGIS and spatial data analysis. In this case you will need about 30 minutes to use this tutorial.

This Tutorial has been developed in the context of the OER4SDi project at the Institute for Geoinformatics, University of Münster. Authors are Soumya Ganguly and Albert Remke.

You are free to use, alter and reproduce the tutorial (H5P content, textual documents, graphics) under the terms of the CC-BY-SA 4.0 license. Any code provided with the tutorial can be used under the terms of the MIT license. Please see the full license terms:

<https://github.com/oer4sdi/OER-DataAccessVia-OGC-API-Features/blob/main/LICENSE.md>

The OER4SDI project has been recommended by the Digital University NRW and is funded by the Ministry of Culture and Science NRW.”.

2 .Background

2.1 Feature Data and Feature Services

Feature Data represent real world phenomena such as roads, buildings, rivers, and land parcels. Features that share a set of properties can be categorized as Feature Types. One of these properties may be the Geometry that represents the spatial location and extent of the Feature. The Geometry itself may be modeled in various ways, e.g. as points, lines or polygons and is based on a specific Spatial Reference System (SRS).

The ISO 19000 Series of international standards provides a conceptual model of Features and Geometries as well as a set of encoding rules that can be used to create Feature Data for example as XML/GML data.

Since many applications don't need the rich capabilities of the full fledged ISO data model for geographic information ISO 19125 defines a simplified set of rules for modeling and working with “Simple Features”. As an alternative to XML/GML, many applications use GeoJSON for a lightweight JSON encoding of Simple Feature data.

The ISO Feature model is often used as the basis for modeling concrete Feature Types such as roads, rivers, and buildings. These data models are called Application Schemas. They are often defined by textual documents and UML diagrams and come along with XML schema files to support the validation of XML encoded Feature data.

Access to Feature Data can be provided in various ways. One way is to distribute the data in the form of a dataset that is encoded e.g. as XML/GML, GeoJSON, Shape or any other format. Another way is to provide a data access service, i.e., a set of operations, accessible through an Application Programming Interface (API) that supports selective download of the data, data transformations or even manipulation of the data source. Commonly used APIs are the OGC Web Feature Service interface (OGC WFS) and the OGC API Features interface.

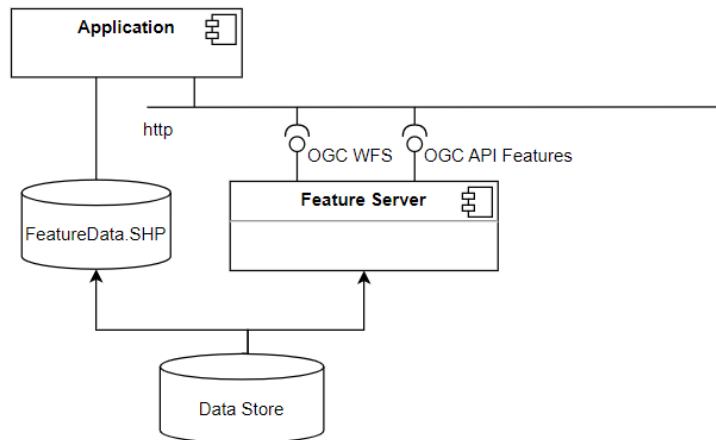


Fig. 1 Access to Feature Data via datasets and services

[comment: link to this drawing:

<https://drive.google.com/file/d/1BdxYJ9iEv2pRKUtEb8cDYVIFH3oyh7nu/view?usp=sharing>]

2.2 Advantages of accessing data with OGC Web Services

OGC Feature Services such as Web Feature Service (WFS) and OGC API – Features offer multiple advantages over conventional OGC feature data formats. These web based services comply with standardized protocols and specifications to ensure seamless data exchange, dynamic querying mechanisms, and effective and uninterrupted data streaming. Let's take a look at the key benefits of using OGC feature services;

1. **Standardized Data Exchange:** OGC Feature Services always adheres to established standards; which aids in maintaining a very high level of interoperability between multiple systems and software implementations. This approach rules out compatibility issues and facilitates a high level of data sharing within the geospatial community.
2. **Ease of Integration:** OGC Web Services is always in sync with established standards in the geospatial information systems (GIS) field. This facilitates the exchange and combination of data from various systems without requiring users to learn different data access methods. Understanding the structure of an OGC web service request suffices, as it remains consistent across different OGC web services.
3. **Dynamic Querying Mechanisms:** OGC Feature Services enables advanced querying capabilities for its users. Aided by this mechanism, users can query out and retrieve specific subsets of data filtered with spatial, temporal or attribute properties.
4. **Efficient Data Processing Workflows:** Data can be accessed on demand, without the need for downloading. The OGC API services facilitate a seamless integration into data processing workflows. This eliminates the need for data to be stored locally, making data processing and analysis workflows more efficient. This is especially valuable when working with tools like Jupyter notebooks or other development IDE's. Data can be

loaded from the hosted published and hosted services. This particular aspect facilitates collaboration and reproducibility among team members.

5. **Community Support:** The OGC standards were developed collaboratively, with input from a diverse community of stakeholders. They were formulated by experts from the geospatial community who ensured that the standards benefited all of the stakeholder community. The OGC Web Services are aligned with OGC standards, which promotes collaborative innovation and development in the geospatial context.

2.3 The OGC-API series of interface specifications

So, what is the difference between the older OGC WFS and the newer OGC API Features interface?

The OGC WFS belongs to the series of classic OGC Web Services, such as the Web Map Service (WMS), the Web Coverage Service (WCS) or the Catalog Service for the Web (CSW). These services typically used SOAP and http to invoke server side operations such as GetCapabilities, DescribeFeatureType, and GetFeature. These remote procedure calls (RPC) resulted in response messages with the requested data as payload.

Different from the RPC style of the classic OGC web services, the new series of OGC API specifications define RESTful web services. One of the key characteristics of RESTful webservices is that http is used as an application protocol, i.e. the semantics of the http methods such as GET, PUT, POST and DELETE are being used to interact with the server side resources rather than just transferring remote procedure calls. Furthermore, the new OGC API specifications follow additional recommendations of the World Wide Web consortium (W3C) that have been published as Spatial Data on the [Web Best Practices](#). For example: “Make your spatial data indexable by search engines” which refers to providing an HTML landing page and crawlable links to metadata and content.

2.4 OGC-API Features interface

OGC API Features is a multi-part standard that provides a unified approach to creating, editing, and querying spatial data on the Web. It establishes requirements and recommendations for APIs, ensuring a standardized way of sharing feature data. Let's explore the key characteristics of OGC API Features and the valuable resources it provides.

Landing Page: The OGC API Features landing page serves as a starting point for further exploring the interface. The landing page serves as a comprehensive overview to further details like general information, features and purposes, and advantages of the API service. Additionally, one can also find the access links to further resources.

/Conformance: To evaluate if the implementation is in compliance with the API specifications, the /conformance endpoint is very important. It enables developers and users to confirm if a service aligns with the defined standards and capabilities. Cross verifying conformance information ensures interoperability and compatibility of the implementation in hand.

/Collections Endpoint: The available feature collections that the API service provides; is made available under the /collections endpoint. The feature collections represent unique sets of geospatial features, filtered and arranged based on criteria like themes, regions, or data sources. From this endpoint a user discovers the collections and follows links to individual resources for further elaboration.

/API Endpoint: The /API endpoint explains the API implementation itself. It contains the API version, title, description, and supported communication protocols. By referring to this resource, you can understand the fundamental aspects of the OGC API Features implementation.

../Items Endpoint: Located at the core of OGC API Features, the ../items endpoint enables users to access and interact with geospatial feature data. To retrieve specific subsets of features using spatial, temporal, and attribute-based filters; this specific endpoint is empowered with powerful querying capabilities which enables efficient data retrieval and analysis.

In terms of structure; OGC API Features is organized into multiple parts, each addressing specific aspects of the standard:

1. **Part 1 - Core:** This part specifies the core capabilities and focuses on fetching features with geometries represented in the WGS 84 coordinate reference system, with the axis order as longitude/latitude.
2. **Part 2:** Part 2 of the standard specifies the capabilities for fetching features with geometries represented in all other coordinate reference systems. It ensures compatibility with a wide range of spatial data representations.

Additional capabilities that address more advanced needs have to be specified in additional parts.

2.5 Accessing Geospatial Data with OGC API Features

Geospatial data from OGC API Features can be accessed and utilized through three methods: browser-based interfaces, programming code, and ready-to-use applications like QGIS. Let's explore the basic workflow and steps involved in accessing geospatial data.

1. **User Interface(Human Interactive Web Browser):** Geospatial data can be accessed and used with a basic web browser or specialized applications sending HTTP Requests and processing Responses.

2. **Programming Code:** We can also access and work on geospatial data with coding languages such as Python and Javascript. Languages like Python maintains their own version of OGC API name PyGeoAPI. Additionally; Python has enriched libraries to perform exploratory data analysis on geospatial data.
3. **Ready-To-Use Applications:** Specialized web applications or desktop applications support OGC API Features. These applications can be configured and connected to the desired OGC API endpoint. Additionally; the applications are well equipped with data visualization features. So once the geospatial data is loaded, users can analyze and work with this data easily.

To access and utilize geospatial data using OGC API Features, follow this basic workflow:

1. Identify the OGC API endpoint, which includes the API version, collection, and query parameters, in a URL starting with "https://".
2. Send an HTTP request to the endpoint using methods like GET, POST, or PUT, specifying the desired operation.
3. Receive the HTTP response, typically in JSON or GeoJSON format, containing the requested data such as matching features or collections.

By leveraging browser interfaces, programming code, or ready-to-use applications, you can efficiently access and work with geospatial data according to your needs.

Further Readings:

[1] An Introduction to OGC API - Features — OGC e-Learning 2.0.0 documentation [Online]. Available: <http://opengeospatial.github.io/e-learning/ogcapi-features/text/basic-main.html> [Accessed: July 17, 2023].

2.6 INSPIRE Geoportal supports OGC API Feature Interface

INSPIRE, the Infrastructure for Spatial Information in Europe, has defined application schemas for a number of INSPIRE data themes such as administrative units, transportation networks, and protected areas. Member states are required to provide access to public sector data related to these data themes through Discovery, View and Download services. While OGC WFS is still a common way of providing INSPIRE download services for feature data, the newer and more lightweight OGC API Features interface is expected to replace OGC WFS in the near future.

3. Practical exercises

3.1 A practical use case: accessing INSPIRE Feature Data on Protected Sites

The following practical exercise will help you understand different ways to access OGC API Feature Data. First task is to select a suitable dataset to explore. We should also try to ensure

that our selected data source is complying to the OGC standards and following the industry wide best practices. Keeping it mind, let's move forward with INSPIRE Geoportal as the preferred datasource. We are going to use the Protected Sites data theme and try to explore different possibilities to access INSPIRE Feature Data. We will first try to publish this data with our own API service. Next we will try to access the Feature Data with a web browser. After that , we will take a deep dive and try to load the Feature Data as layers in QGIS. This will help us understand how to visualize data and get deeper insights from INSPIRE Feature Data. Additionally, it will help us draw constructive conclusions about Protected Sites being investigated. Finally, we will try to access the published Feature Data with Jupyter notebook and Python. This section is for deep diving into the data visualization process. The exercise will teach us how to perform exploratory data analysis from accessed Feature Data and draw reasonable conclusions.

[add: here we should outline the practical use case that serves as an example for the need of data and for the usage of a web service for accessing that data. Focus is on accessing (INSPIRE) feature data via OGC API Features web service.

Verify: that the dataset on Protected Sites conforms to the INSPIRE PS data specification; just look up the metadata

]

3.2 Preparing software and data for the exercises

As to provide a safe runtime environment with all the software and data in place for this tutorial we will use pre-configured docker images. Don't worry if you're not already familiar with Docker, we'll lead you through the installation process step by step:

1. **Download Docker Desktop:** Visit the Docker website and download the Docker Desktop installer for your operating system. Choose the appropriate version for Windows or Mac
2. **Run the Installer:** Locate the downloaded Docker Desktop installer file and double-click on it to run the installer.
3. **Follow the Installation Wizard:** The Docker Desktop installer will guide you through the installation process. Follow the on-screen instructions, accept the license agreement, and choose the desired installation options.
4. **Start Docker Desktop:** Once the installation is complete, Docker Desktop should automatically start. Look for the Docker whale icon in your system tray or applications menu and click on it to open Docker Desktop.
5. **Clone the Repository:** Use Git to clone your repository to a local directory on your machine.
 - a. [Download](#) the zip file for the code and unzip it in a location that you want to use as a WorkingDirectory.
 - b. **For advanced users:** Open a terminal or command prompt, navigate to the desired directory, and run the following command:
 - c. `git clone https://github.com/oer4sdi/OER-DataAccessVia-OGC-API-Features.git`

6. **Compose and build the docker environment:** Goto the terminal and move to the project directory. Type **`docker-compose up --build`** to build and start the docker environment.

Docker Architecture Used

The below figure is the diagrammatic representation of the Docker Environment we will be using. In this Docker Environment, we will be using two docker containers; the first one for Pygeoapi Docker Image and the second one for Python Visualization packages and Jupyter Notebook.

- **In the Pygeoapi docker container;**
 - The latest version of geopython-pygeoapi docker image is installed.
 - Additionally, in our Project Directory we will have a file named; local.config. Information related to the new datasets that needs to be published should be added in the resource section of this config file. It will overwrite the default config file used by Pygeoapi docker image.
 - Again, the data folder which contains the actual data needs to be loaded in the Docker Environment.
 - Any other relevant additional dependencies will already be installed inside this container.
 - App Folder: A volume reference for an app folder is also used. It contains the python notebook for performing the spatial data visualization.
- **In the Python Jupyter Notebook Docker container,**
 - We are using the Python minimalist docker image.
 - Additionally, we will install the data visualization requirements from the requirements.txt file referenced in the Dockerfile.
 - For the geospatial visualization setup to work efficiently, additional dependencies might be installed in this docker container.

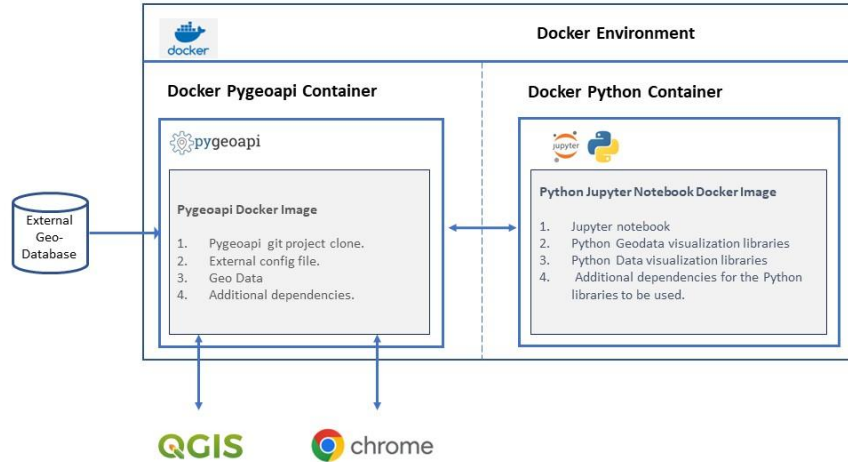


Fig 2: Docker Environment

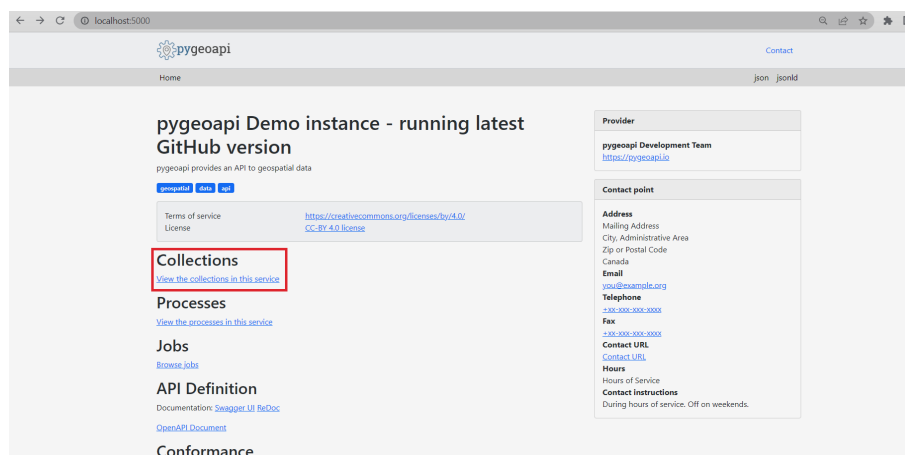
QGIS Installation

1. Go to the QGIS website at <https://qgis.org/en/site/forusers/download.html> and click on the Download button.
2. On the Downloads page, you will see several options for different operating systems. Choose the appropriate option for your system, either Windows, macOS or Linux.
3. Once you have chosen the appropriate option, you will be taken to the download page for that version. Click on the download link for your operating system.
4. When the download is complete, open the installation file. For Windows, the file will have a .exe extension. For macOS, the file will have a .dmg extension.
5. Follow the instructions in the installer to install QGIS on your system. The installer will guide you through the process of selecting the components you want to install, choosing the installation location, and other options.
6. Once the installation is complete, launch QGIS by clicking on the shortcut on your desktop or in the Start menu.
7. When QGIS starts up, you may be prompted to select a default projection. Choose the projection that is most appropriate for your location.
8. You're now ready to use QGIS! You can start by loading some data into the program and exploring the various tools and features that are available.

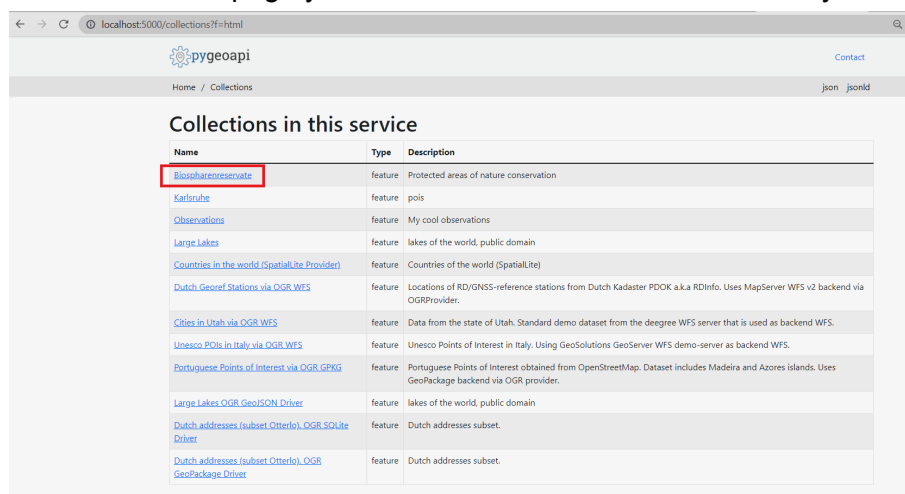
3.3 Using a standard web browser to interact with the OGC-API Features web service

Interacting with the OGC API Features web service through a standard web browser offers several advantages for human users. The PyGeoAPI services, once installed and running, are easily accessible by inputting the URL "http://localhost:5000/" into the browser's address bar. The landing page provides a user-friendly interface, making it simple for users to navigate through the available features. Let us follow a step by step process to understand how it works.

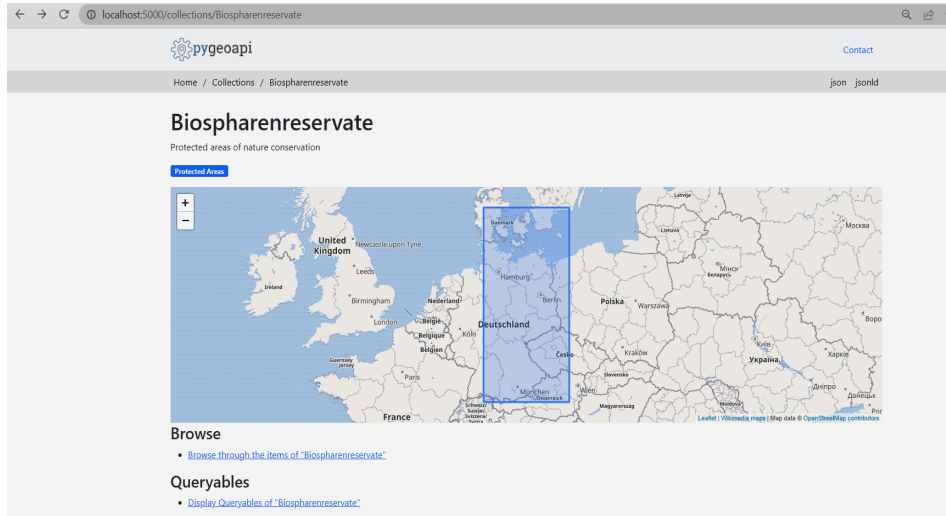
1. Firstly, follow the above installation steps with docker. Now, once the docker container is up and running; the PyGeoAPI service instance will be available and running in your local system. Input <http://localhost:5000/> on your browser url tab. The browser will open the PyGeoAPI landing page.



2. Next move to the Collections tab by clicking on **“View the collections in this service”**. In the collections page you will see all the features available to you via this API service.



3. To explore a specific feature, users can choose their preferred option from the list. Let's take a closer look at the Biosphärenreservate feature. The web browser displays the feature data in a clear and understandable manner, allowing users to inspect the protected areas of nature conservation in detail.

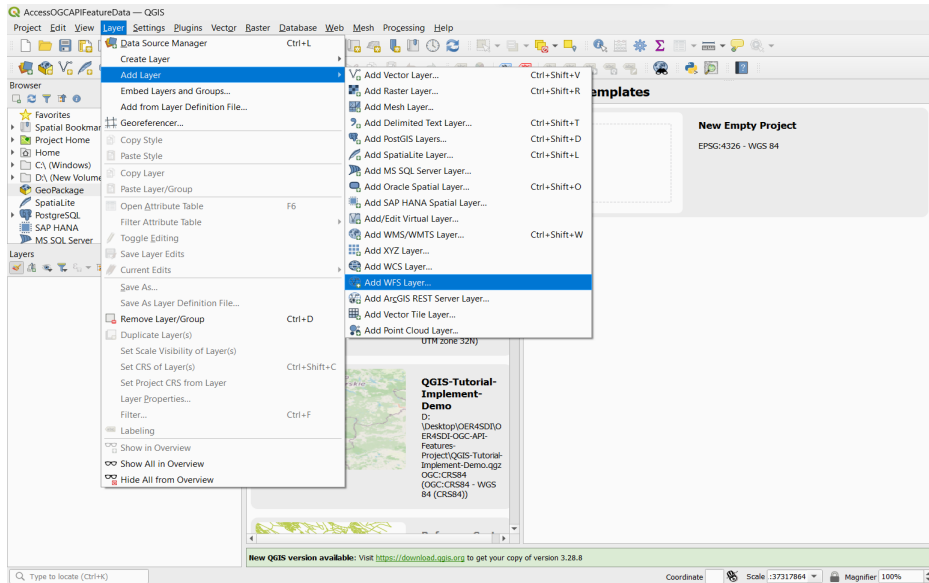


The OGC API Features' design prioritizes direct interaction with human users, which significantly improves usability. Users, without the need for complex queries or technical knowledge, can effortlessly browse, search, and retrieve geospatial information.

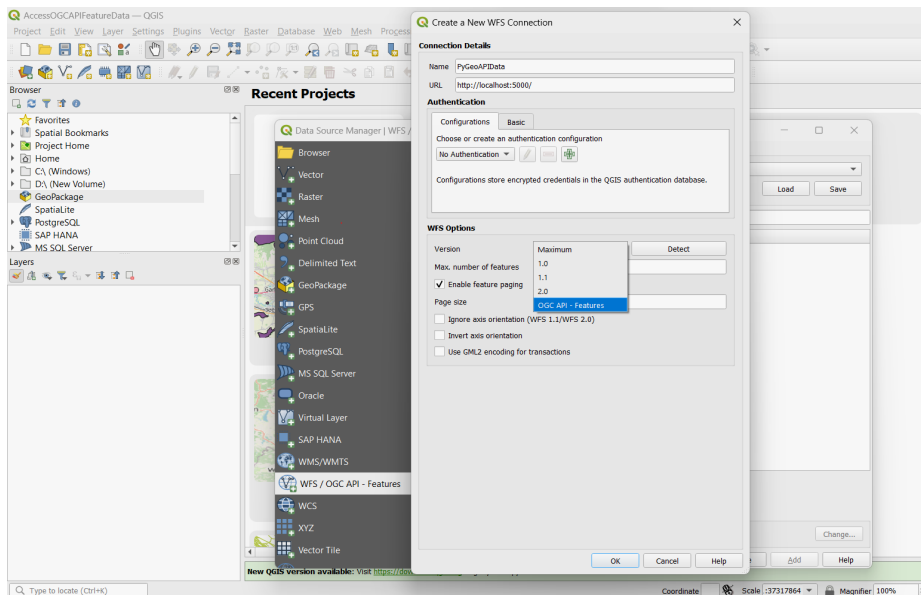
3.4 Using QGIS to access feature data via OGC API Features

The OGC API feature data can also be accessed and visualized with GIS softwares like QGIS and ARCGIS Pro. In this tutorial, we will explore the process to access and visualize an OGC API feature with QGIS. Let's use the same PyGeoAPI service we have up and running in our local system.

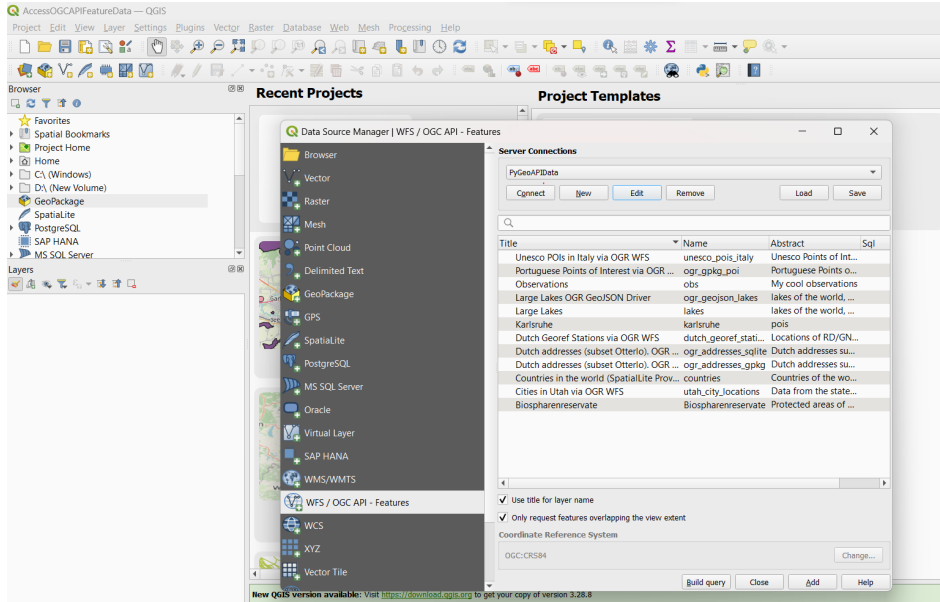
1. Firstly, we need to copy the url of the OGC API service which is accessible from the browser. It is **`http://localhost:5000/`**.
2. Next, we Launch QGIS and select the "Add Layer" button in the toolbar, or go to "Layer" > "Add Layer" > "Add Layer".



3. To add a layer from an OGC Feature service, select the "Add WFS Layer" option. In the dialog box, select "New" and enter the Name and URL for the published Pygeoapi service(OGC API service) connection. The URL will be <http://localhost:5000/>.
4. Next, select the appropriate connection type (e.g., OGC API - Features, WFS, etc.) from the Version dropdown menu.



5. Click on "Connect" to establish a connection to the service. If authentication is required, enter the necessary credentials.
6. Once connected, the available layers from the OGC Feature service will be displayed. Select the layers you want to add by selecting the Layer and clicking on the Add option.



3.5 Using Python to interact with the OGC-API Features web service

Introduction: The final part of this tutorial deals with,

1. A low level implementation of OGC API service. We will use Python PyGeoAPI services for this purpose.
2. Integrate PyGeoAPI with Python Jupyter Notebook and fetch the OGC API feature Data published. For this we need to integrate PyGeoAPI with the Python Jupyter Notebook environment.
3. Finally, we will use Python Data Visualization libraries; especially geospatial libraries like Geopandas to perform spatial data visualization.

Video to be inserted

4. Discussion and Wrap up

[

- summarize the learning journey
- highlight the special characteristics of OGC API Features from a user perspective
- refer to some examples (operational web services based on OGC API Features)

]

[
still missing in the OER: elements that activate the learner (tasks, check your knowledge..)
]

Plan:

Python notebook already designed. Further activity in the python notebook and Hands-on. In case, some special instructions needs to be added, it will be documented.

Video to be inserted.

Wrap Up

1. Do a revisions of the exercises we did.
2. Discuss future scopes.
3. Pros/cons
4. Have a exercise/assignment for students

DataSources:

https://inspire-geoportal.ec.europa.eu/download_details.html?view=downloadDetails&resourceId=%2FINSPIRE-4fed3eb0-06fa-11ea-8480-525400695e9c_20221103-110203%2Fservices%2F1%2FPullResults%2F196081-196100%2Fdatasets%2F20&expandedSection=metadata