Fetching and Processing Spatial Data from a WFS Server

Overview

This script retrieves and processes spatial data from a **Web Feature Service (WFS)** hosted by the UK Environment Agency. It performs two main tasks:

1. **Fetching available layers** using a `GetCapabilities` request.
2. **Downloading and saving each layer's data** as a **GeoPackage (`.gpkg`)** file.

The purpose of this script was to extract geospatial files for further analysis. Once the data is stored as geospatial data file(s), we can then run those files through the eda_report script to produce the analysis report.

It uses:

- `requests` for HTTP requests.
- `xml.etree.ElementTree` to parse XML responses.
- `geopandas` to handle and save spatial data.

Step 1: Fetching Available Layers (`GetCapabilities`)

The script starts by requesting the **capabilities document** from the WFS server, which provides metadata about available datasets. The request retrieves an XML document containing information about all available layers.

If the request is successful, the script:

- Parses the XML response to find `<FeatureType><Name>` tags.
- Extracts the names of the available layers.
- Prints the retrieved layers.

If the request fails, it prints an error message and does not proceed.

Step 2: Fetching and Processing Each Layer (`GetFeature`)

For each extracted layer, the script:

1. Constructs a `GetFeature` request to fetch the spatial data in **GeoJSON format**.
2. Sends the request to the WFS server.
3. If the response is successful:

- Parses the JSON response.
- Converts the spatial data into a **GeoDataFrame** using `geopandas`.
- Displays the first few records (`.head()`) and checks the **Coordinate Reference System (CRS)**.
- Saves the data as a **GeoPackage (`.gpkg`)** file for further GIS analysis.

If a layer fails to download or process, an error message is displayed.

Output and File Storage

Each processed layer is saved as a **GeoPackage (`.gpkg`)** file. The filenames are generated dynamically by replacing any `:` characters in the layer names with `_`, ensuring they are valid file names.

In [ ]:
```
import requests
import xml.etree.ElementTree as ET
import geopandas as gpd

# Step 1: GetCapabilities - Fetch and store available layers
capabilities_url = "https://environment.data.gov.uk/spatialdata/casi-and-lidar-habi

response = requests.get(capabilities_url)
if response.status_code == 200:
    # Parse the XML response
    root = ET.fromstring(response.text)

    # Find all layers (FeatureType Name tags)
    namespaces = {'wfs': 'http://www.opengis.net/wfs/2.0'}
    layers = [
        elem.text for elem in root.findall(".//{http://www.opengis.net/wfs/2.0}Feat
    ]

    # Print the list of layers
    print("Available layers:")
    for layer in layers:
        print(layer)
else:
    print(f"Error fetching capabilities: {response.status_code} - {response.text}")
    layers = []  # Set to empty list if fetching fails

# Step 2: Iterate through each layer and fetch data
wfs_url = "https://environment.data.gov.uk/spatialdata/casi-and-lidar-habitat-map/w

for layer_name in layers:
    print(f"\nProcessing layer: {layer_name}")

    # Specify parameters for the layer
    params = {
        "service": "WFS",
        "version": "2.0.0",
        "request": "GetFeature",
        "typeName": layer_name,  # Use the current layer name
        "outputFormat": "application/json",  # Ensure JSON format
    }

    # Fetch data
    r = requests.get(wfs_url, params=params)
```

```
    if r.status_code == 200:
        try:
            response_json = r.json()  # Parse JSON
            data = gpd.GeoDataFrame.from_features(response_json["features"], crs="E

            # Display and process data
            print(data.head())
            print(f"CRS: {data.crs}")

            # Save to file
            output_file = f"{layer_name.replace(':', '_')}.gpkg"
            data.to_file(output_file, driver="GPKG")
            print(f"Layer '{layer_name}' saved to '{output_file}'")
        except Exception as e:
            print(f"Error processing layer '{layer_name}': {e}")
    else:
        print(f"Error fetching layer '{layer_name}': {r.status_code} - {r.text}")
```