How to use Air Quality Downloads page

Date: April 2025

Document History

Version	Date	Author(s)	Remarks
0.1	2023-06-29	AE	
0.2	2023-09-08	AE	
0.3	2023-11-14	AE	
0.4	2024-04-15	AE	
0.5	2024-05-03	AE	
0.6	2024-06-06	AE	
0.7	2024-11-18	AE	
0.8	2025-03-31	MV	
0.9	2025-04-08	AE	
1	2025-12-09	MV	
1.1	2025-12-16	AE	

Contents

Overview

These guidelines explain how to use the new Air quality Download Service from both web application and API.

- **Web application:** https://eeadmz1-downloads-webapp.azurewebsites.net
- **API:** https://eeadmz1-downloads-api-appservice.azurewebsites.net/swagger/index.html

Throughout the document, the different options that exist in the web application to download the data in the most useful way for users will be shown in detail, as well as some examples of how to use the API to download without having to use the web application.

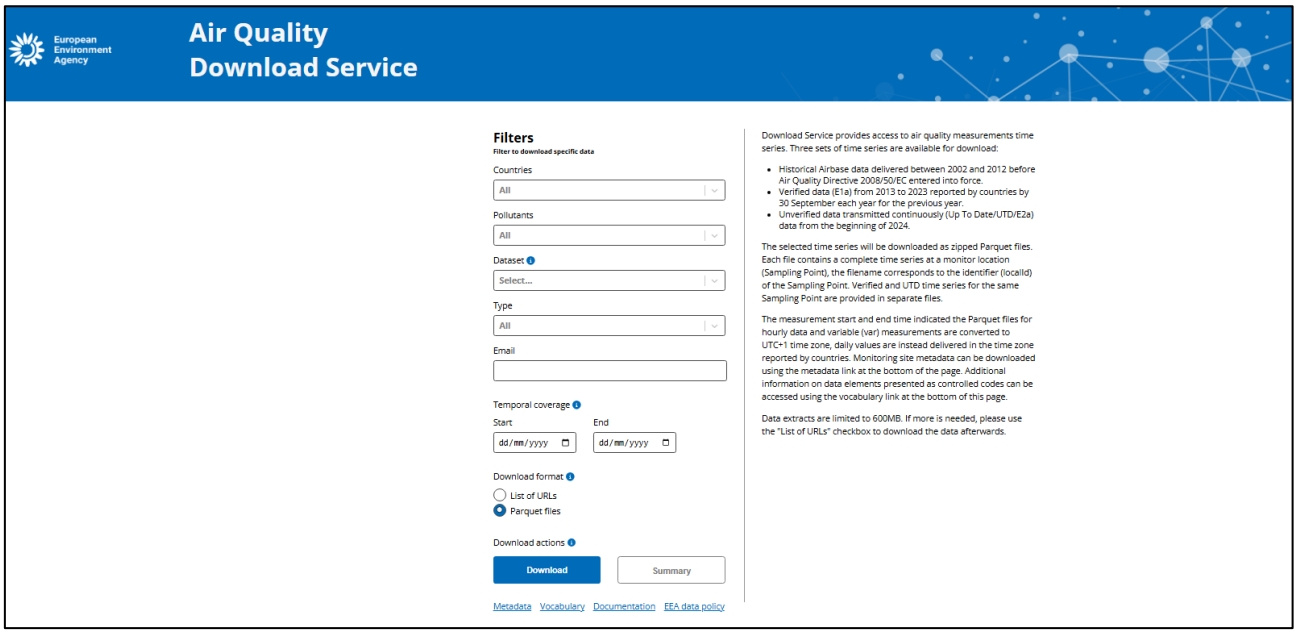
Downloads can be made in two different formats while using the above-mentioned services:

1. **Parquet files:** They contain the complete time series of the corresponding sampling points.
2. **URLs:** The download generates a CSV with the URLs corresponding to the parquet files of each of the desired sampling points.

1. Air quality Download service page

The download page is divided into two main blocks.

1. **Interface**: This feature allows you to filter and select the desired for downloading directly to your browser (on the left side).
2. **Description of the Download Service**: Detailed information about this service is available on the right side of the screen.



In addition, the download page provides information related to the metadata, a link to list of vocabularies describing definitions that appear in the downloaded files and, finally a link to this documentation.

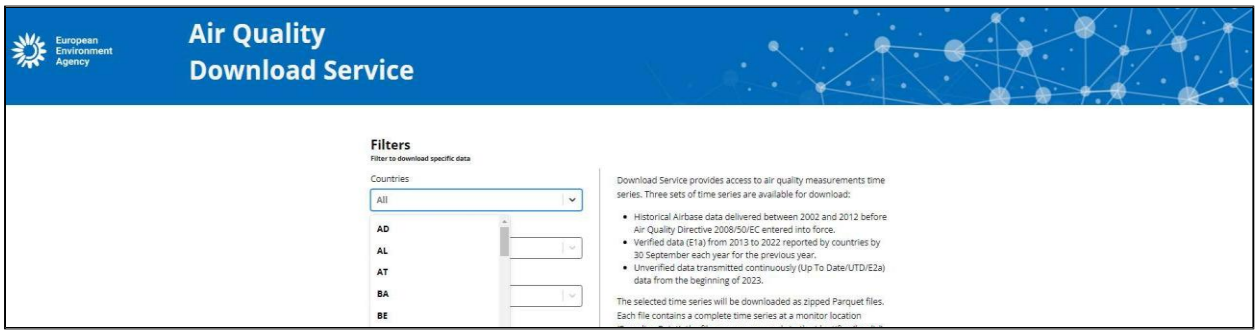
1.1. Filters

In this section, the operation of the filters within the application will be explained. It is important to know that none of the filters are mandatory, so if any of the filters are not filled, the download will be performed on all possible values.

For example, if the pollutants filter is empty, the corresponding files will be download with all the pollutants that exist in the system.

Countries

Using this filter allows the selection of one or more specific countries for which the available data will be downloaded. If no country is selected, data for all countries will be downloaded.



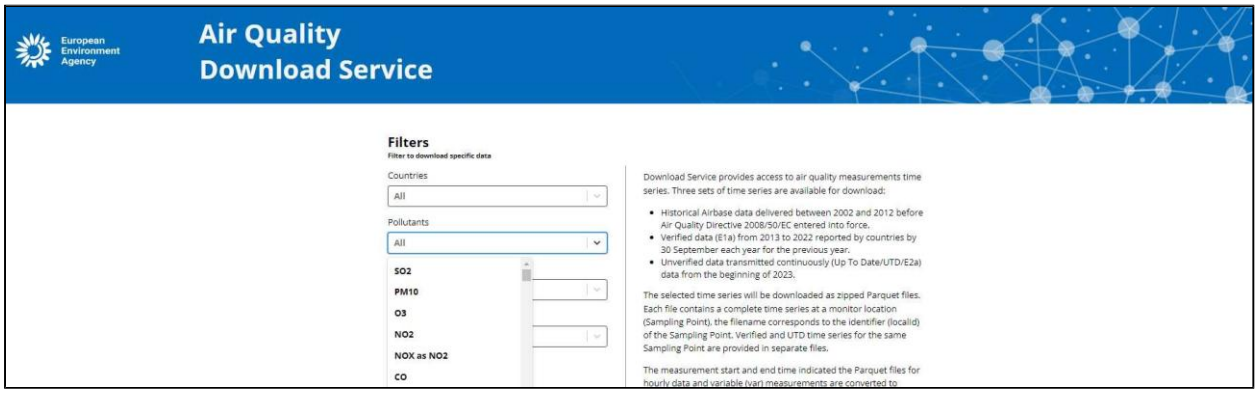
Cities

If any of the selected countries contain cities, a new filter will be displayed containing the corresponding cities. Here, as in the country filter, it is possible to select one or more cities, or leave it blank to return all cities in the country.

The information on the geometries of each city has been extracted from the data source available in Urban audit. With this and with the information that contains the coordinates of where each station is located available in this link, the intersection has been made to determine the stations of each of the cities.

Pollutants

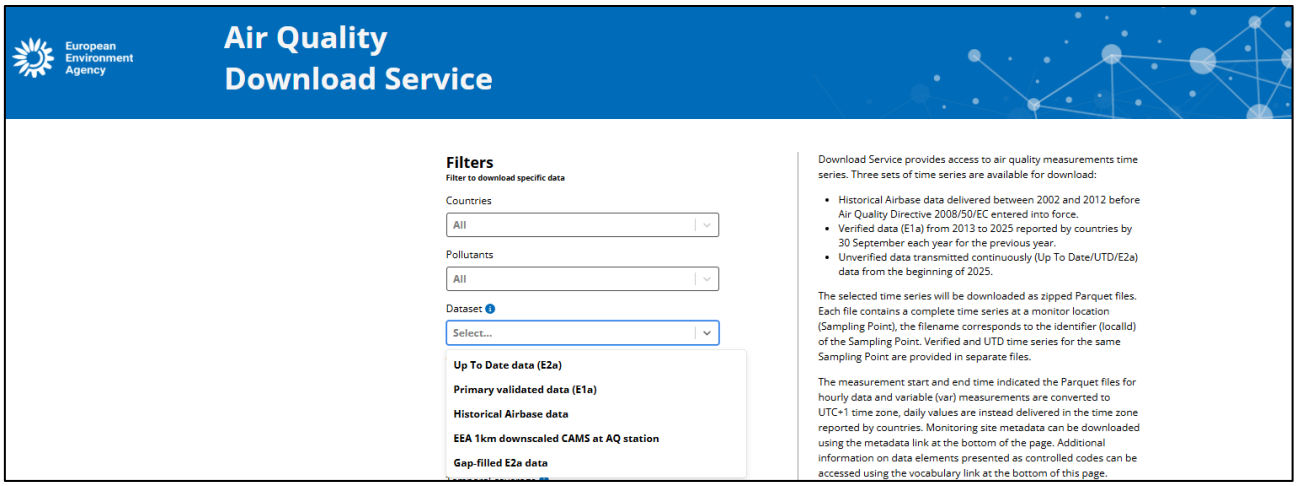
It is possible to filter the information for the desired pollutants. Like the rest of the filters, it is possible to select one or more options, or none and therefore get data for all the pollutants.



Dataset

It is possible to filter and download the verified data (E1a) reported by countries, the most recent unverified data transmitted continuously (Up-To-Date/UTD/E2a), or the historical Airbase data delivered between 2002 and 2012. Additionally, the service offers gap-filled E2a and a downscaled version of CAMS at station level.

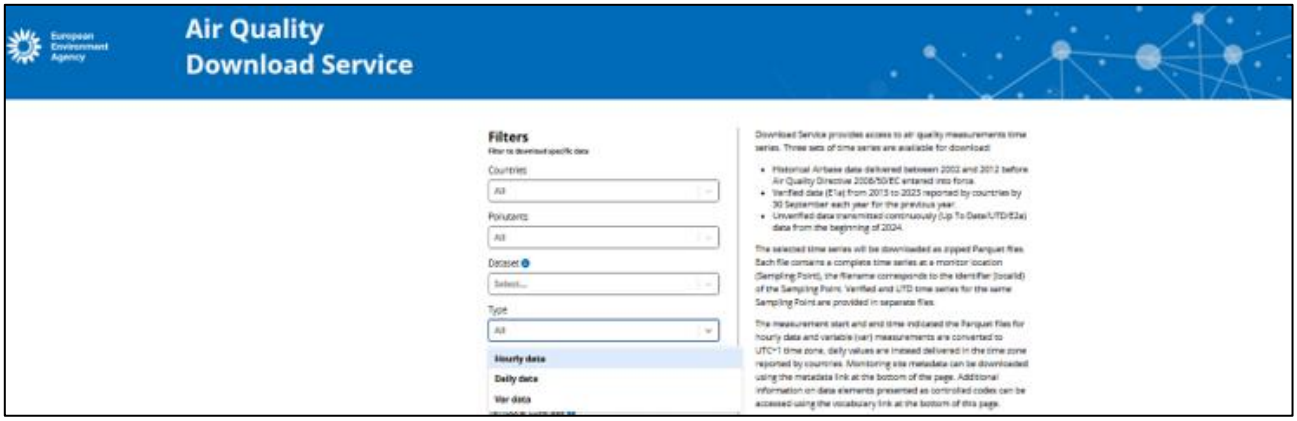
This is possible through the "Dataset" filter and in this case, the value is mandatory.



Type

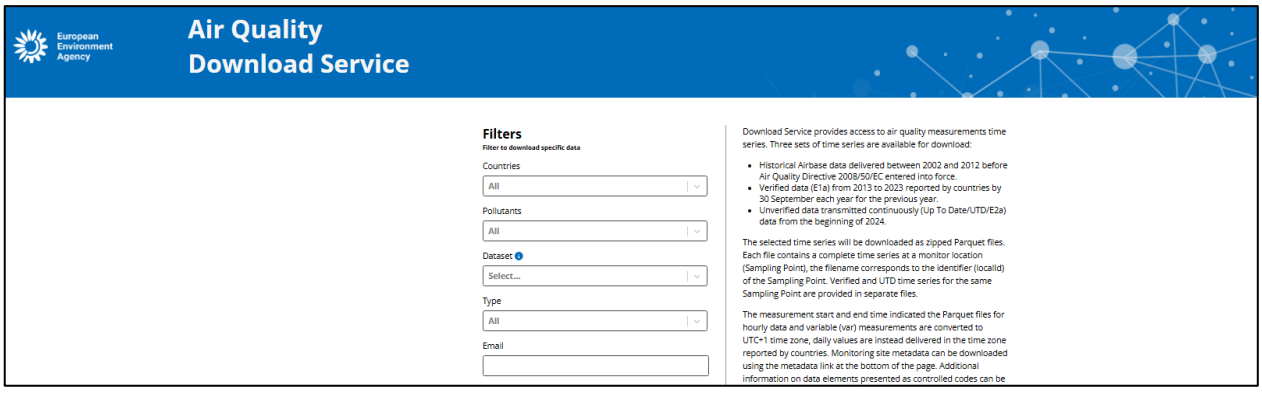
It is possible to filter and download the data by selecting the type of data. This type represents whether the data collected is obtaining the values in hourly, daily or variable intervals (intervals different than the previous observation such as weekly, monthly, etc.). It is possible to select one option, or none and therefore get data for all the types.

Bear in mind that this filter will not affect Summary requests. For performance reasons, Summary considers entire timeseries matching location and dataset, regardless of datetimes and types.



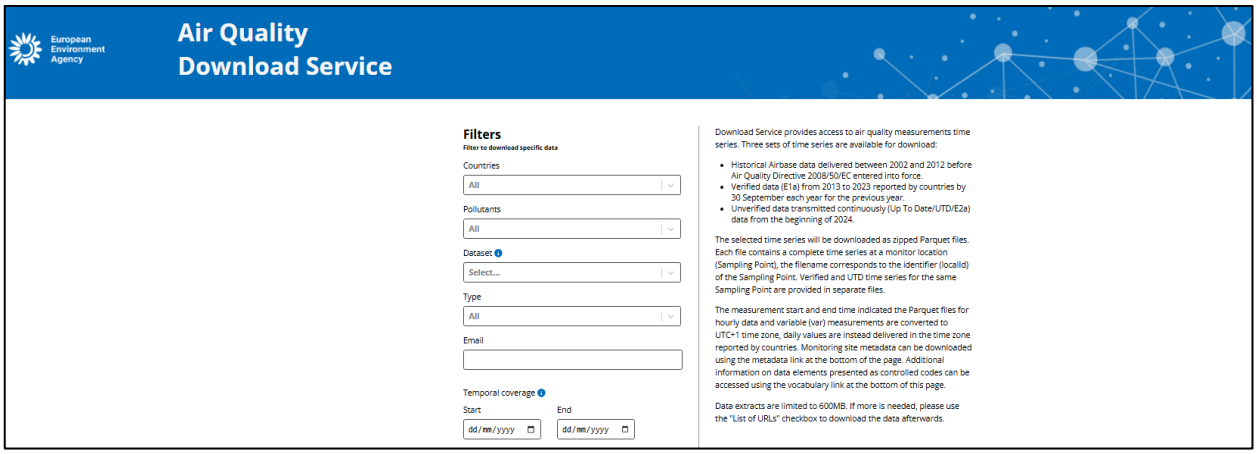
Email

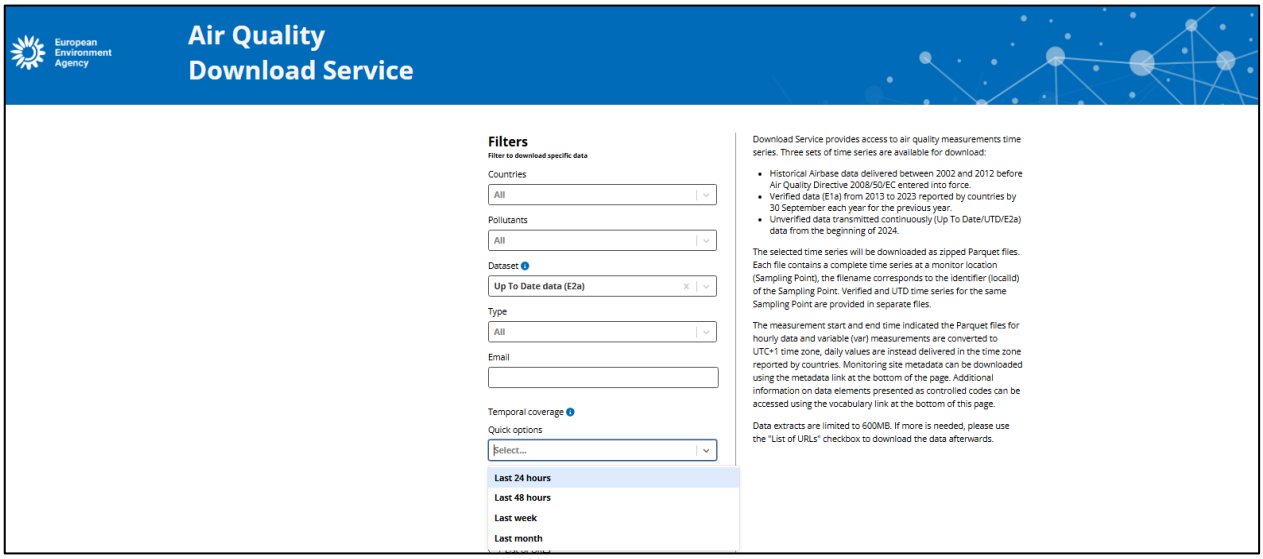
It is possible to enter the email as additional information. This field is optional; however, it is recommended since it is a way to provide information about the user who downloaded the data, and if problems are detected, communication is much easier and direct.



Temporal coverage

It is possible to filter and download the data with a specific temporal coverage. For this it is necessary to select the beginning (Start) and the end (End) of the period to be downloaded.
In case Up To Date data (E2a) has been selected, it is possible to select a typical download temporal coverage. If no temporal coverage is selected full data will be downloaded.





Bear in mind that this filter will not affect List of URLs or Summary requests. For performance reasons, Summary considers entire timeseries matching location and dataset, and List of URLs provides the locations for those entire files, regardless of datetimes.

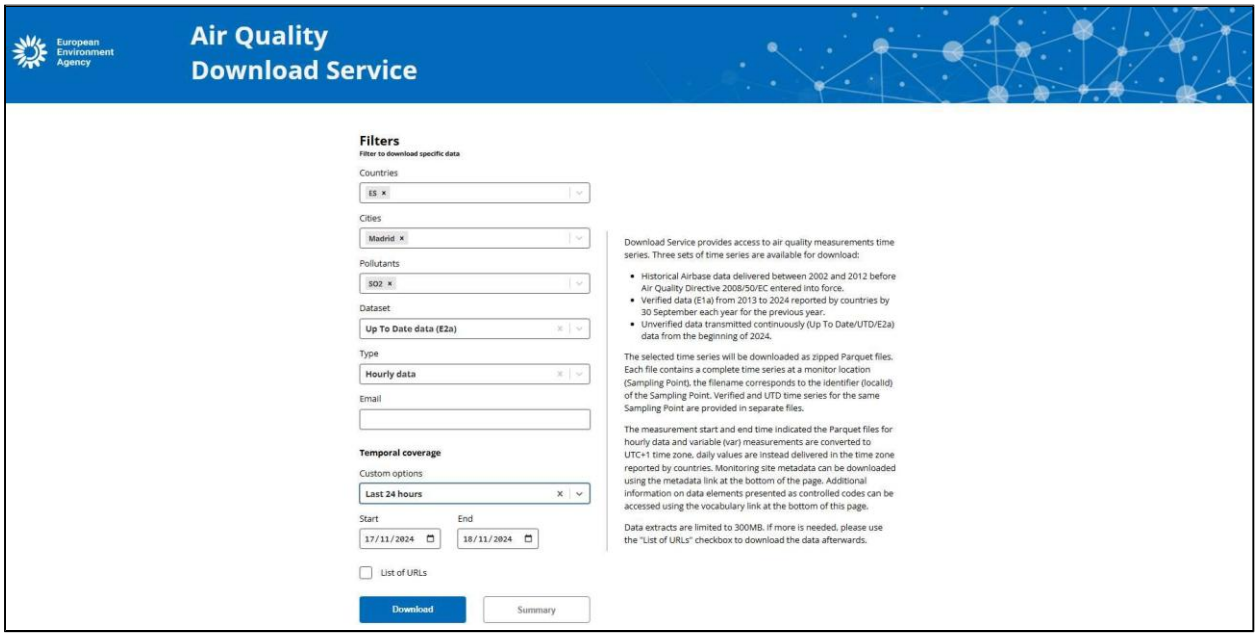
URLs

Depending on what is desired, it is possible to download only the URLs of the selected files instead of the entire parquet file. To do this, activate the radio button. If this option is selected, the output will not consider the datetime filters.

1.2. Download data

Once the desired filters have been selected, to download the data, it is necessary to click the "Download" button and after waiting, the files will be automatically downloaded in the browser.

There is the possibility, instead of downloading, to obtain a summary of the number of files and the size of the download by pressing the "Summary" button. It can be helpful to have an idea of the amount of data and files that are going to be downloaded, and if the maximum limit is exceeds (600 MB), download the URLs to access the files as there is a limit to the download. As mentioned above, to download the URLs, activate the "List of URLs" radio button.

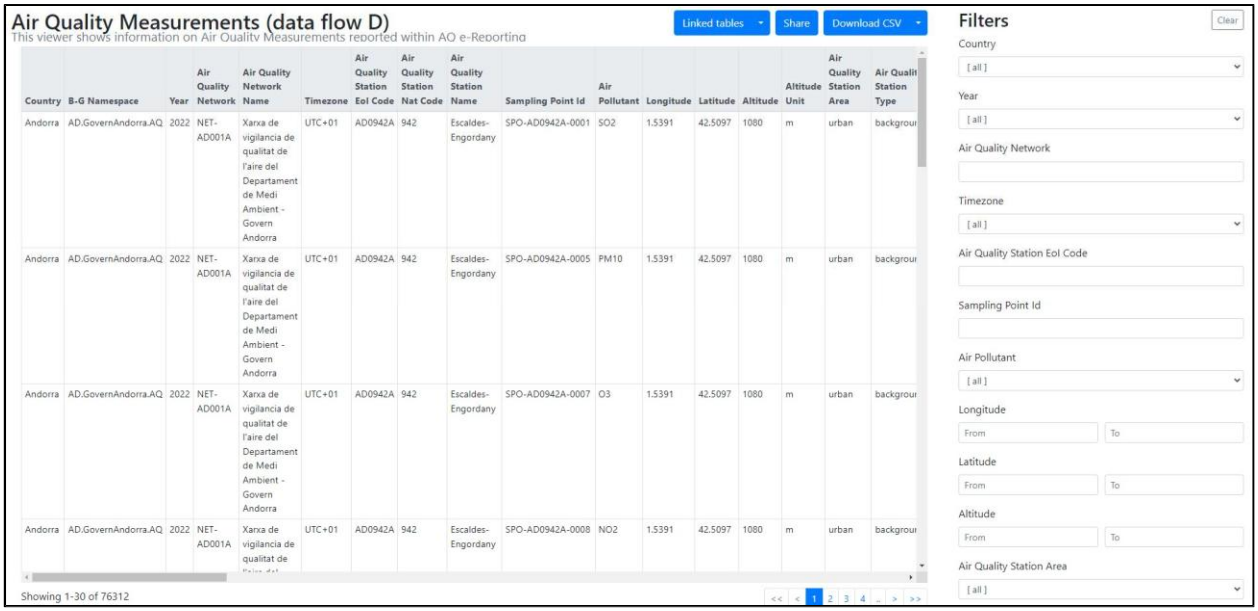


1.3. Metadata

By clicking on the “Metadata”, it is possible to access a new page that contains a summary of all the metadata for all countries. Here you will find an interactive table which shows information on Air Quality Measurements reported within AQ e-Reporting to the Eionet CDR repository.

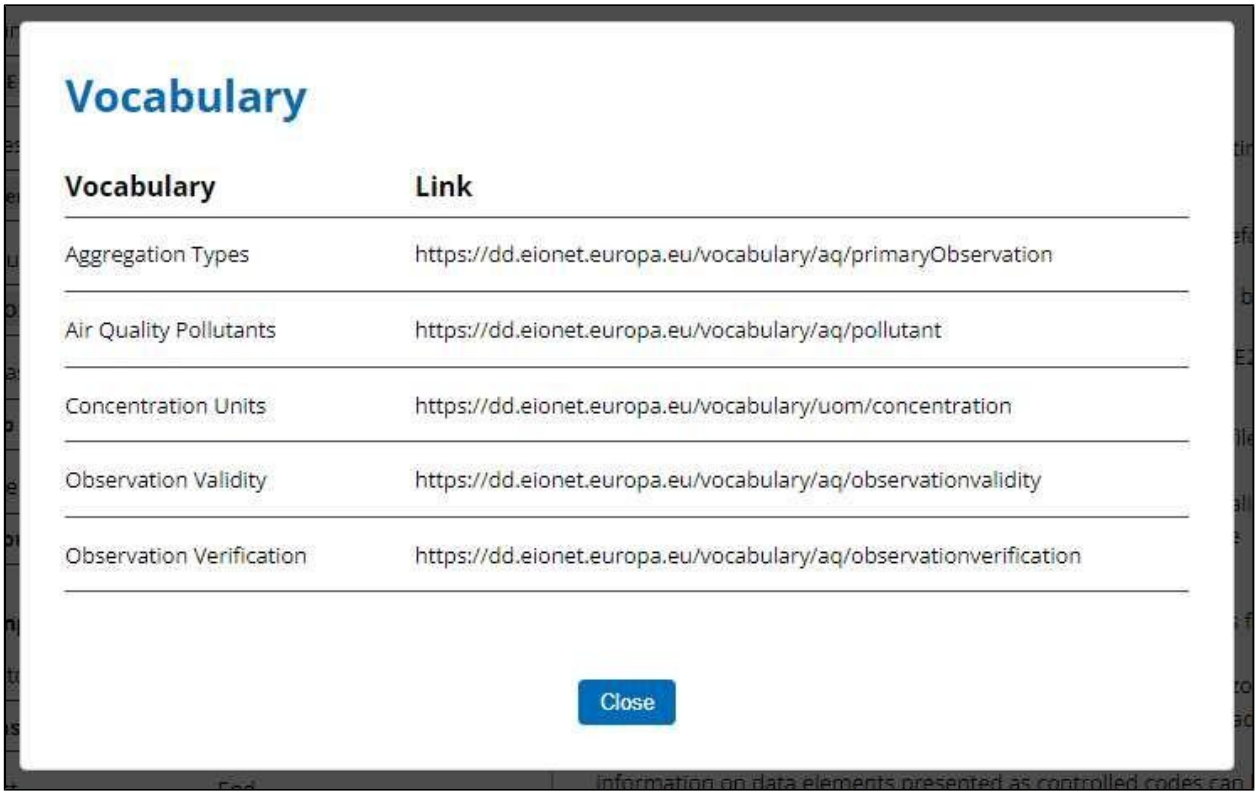
Direct link:
https://discomap.eea.europa.eu/App/AQViewer/index.html?fqn=Airquality_Dissem.b2g.measurements

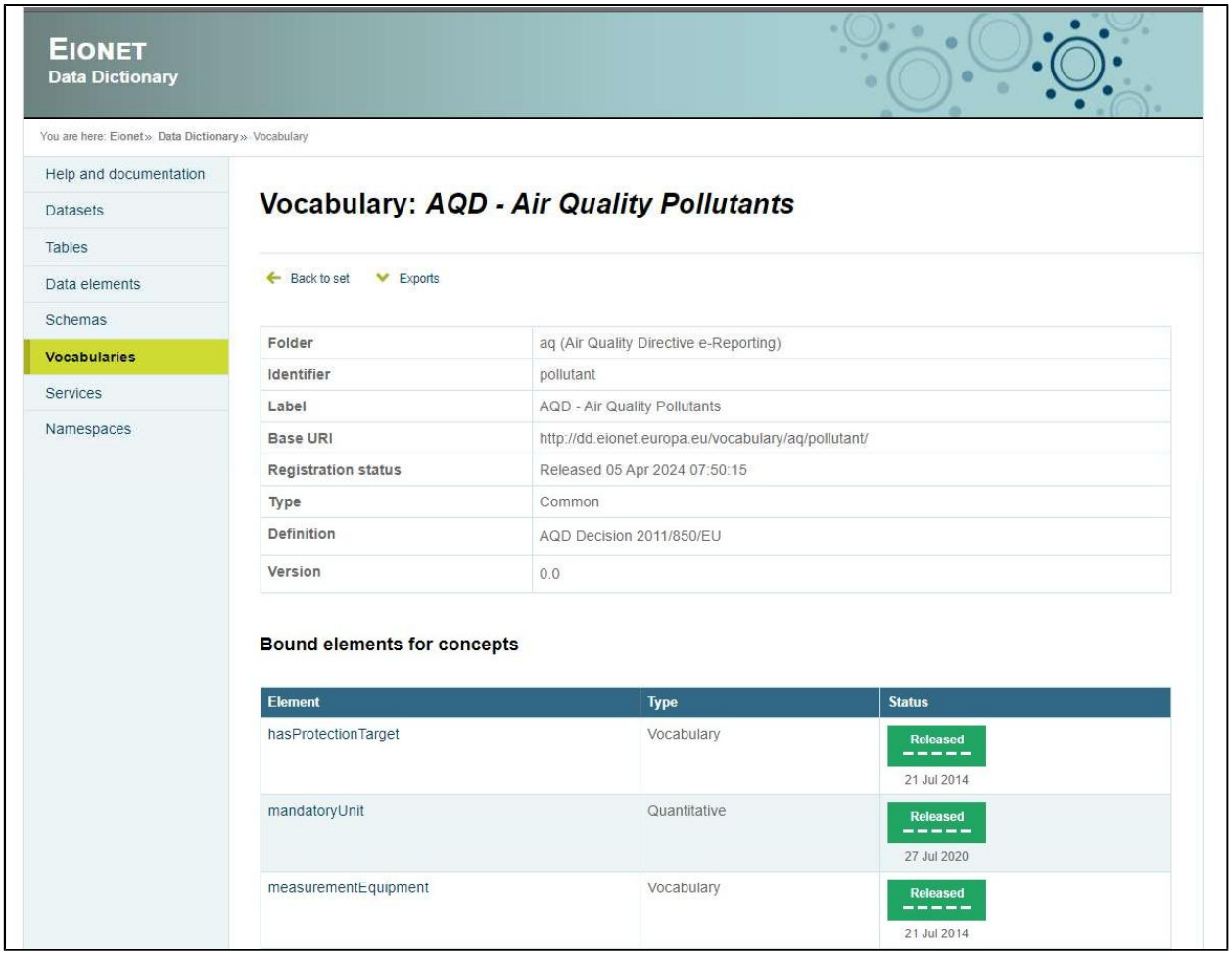
This page allows downloading the metadata and filtering by different options. Bear in mind that on this page, it is only possible to download the metadata sent by each country about the networks, stations, sampling points, etc., but not the air quality measurements themselves. For this, it is necessary to use the download page described in this document.



1.4. Vocabulary

From the "Vocabulary" link, users can find a summary of the various vocabularies that are used and appear in the downloaded parquet files. By clicking on the different links, users will be directed to the Eionet Data Dictionary, which provides the definitions for each of these vocabularies. It must be considered that on this page, it is only possible to download metadata (networks, stations, sampling points, …), but it is not possible to download the data collected by the different stations about the different pollutants. For this, it is necessary to use the download page described in this document.





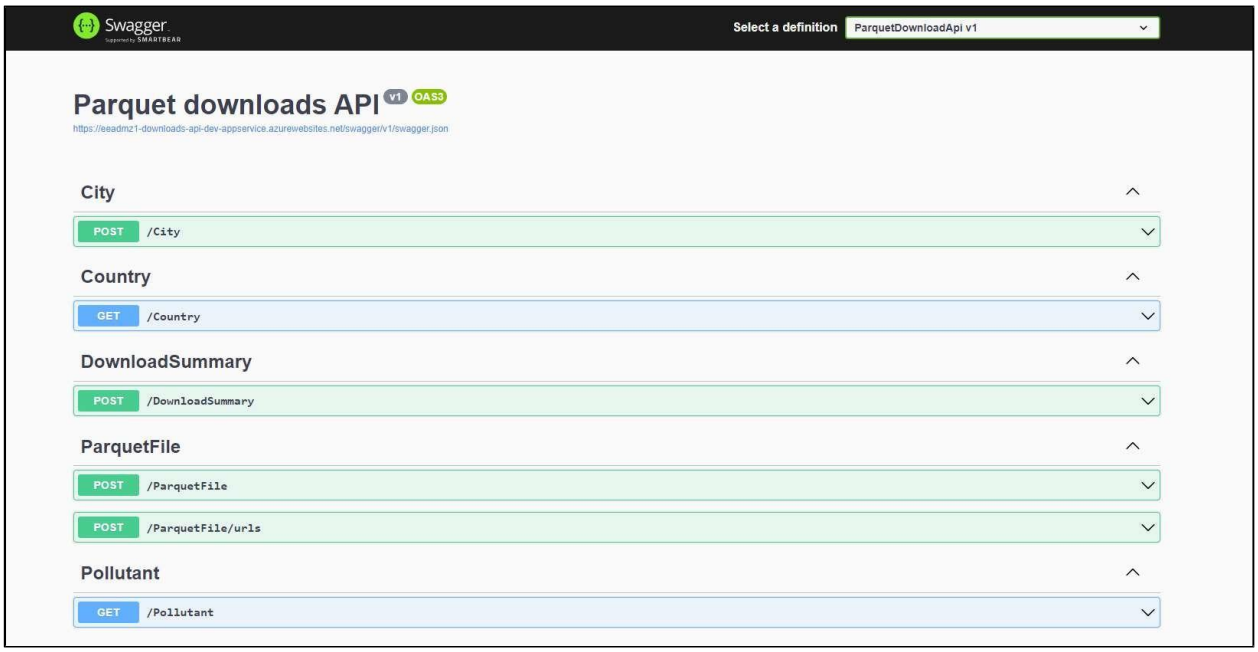
1.5. Documentation

Through the link "Documentation" a new tab opens in the browser accessing this document.

2. Air quality Download service API

There is a possibility of accessing an API to consult the available methods that exist in everything related to downloading files.

- **API:** https://eeadmz1-downloads-api-appservice.azurewebsites.net/swagger/index.html



Here the desired method is selected. To make a request from the API, it is necessary to press the "Try it out" button of the desired endpoint.

The most interesting endpoints are:

- **ParquetFile**: Download zip file with all the filtered parquet files. If the 600MB limit is reached, the status code returned by the endpoint will be 206 instead of 200.
- **ParquetFile/async**: Download in the background. The endpoint returns an URL in which the file to be downloaded will be generated.
- **ParquetFile/urls**: Download csv file with all the URLs corresponding to the filtered parquets.

Both endpoints have the same parameters to be able to perform filtering and obtain the desired information. However, ParquetFile/urls ignores the effect of datetime and aggregation type filters.

After filling in the values of the request, press the "Execute" button. The values must be filled in as follows:

- **countries**: values with country code separated by commas. If no value is entered and an empty list is passed, the filter will not be applied and the data for all the countries will be downloaded. Example: ["IS", "DK", "NO"].
- **cities**: name of cities separated by commas. If no value is entered and an empty list is passed, the filter will not be applied and the data for all cities will be downloaded. Example: ["Berlin", "Madrid"].
- **pollutant**: It is necessary to enter the **URLs** corresponding to each pollutant or de **notation** of each of the pollutants separated by commas. The URL can be obtained by clicking on the different pollutants in this page and copying the value of the **Concept URI**. The notations can be obtained directly from the previous page**.** If no value is entered and an empty list is passed, the filter will not be applied and the data for all pollutants will be downloaded. For example: properties: ["http://dd.eionet.europa.eu/vocabulary/aq/pollutant/3", "http://dd.eionet.europa.eu/vocabulary/aq/pollutant/1"] or ["SO2", "CO"].

- **dataset**: value of the dataset:
    1. Unverified data transmitted continuously (Up-To-Date/UTD/E2a) data from the beginning of 2023.
    2. Verified data (E1a) from 2013 to 2022 reported by countries by 30 September each year for the previous year.
    3. Historical Airbase data delivered between 2002 and 2012 before Air Quality Directive 2008/50/EC entered into force.
    4. Downscaled CAMS forecast at station level
    5. Gap-filled E2a, using linear interpolation for smaller gaps, and a regressor based on similar stations for larger ones.

- **source:** optional value that indicates where the download was made from. If the download is made from the official download website, source will contain the value "website". If it is done from the API directly or from a custom script, the parameter can be omitted or enter values such as "API", "Custom script". The objective is to be able differentiate in the system what it has been downloaded from the web and that directly using the API.

- **dateTimeStart:** value that indicates from which date and time user want to filter the download. If parameter is not pa Format: yyyy-mm-ddTHH:MM:SSZ. Example: 2024-05-27T12:00:19Z. This parameter is ignored in ParquetFile/urls

- **dateTimeEnd:** value that indicates until which date and time user want to filter the download. Format: yyyy-mm-ddTHH:MM:SSZ. Example: 2024-05-28T12:00:19Z. This parameter is ignored in ParquetFile/urls

  If **dataTimeStart** and **dateTimeEnd** parameters are not included in the request, the filter for the temporal coverage will not be applied and the entire set of data will be downloaded.

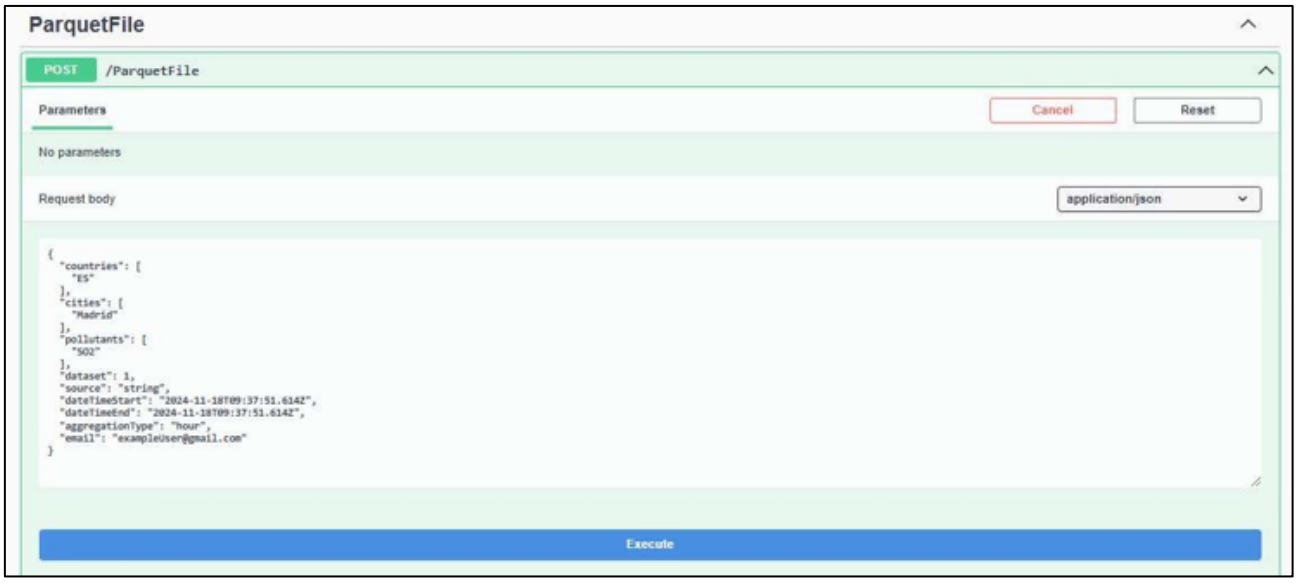
- **aggregationType:** represents whether the data collected is obtaining the values:

    1. Hourly data.

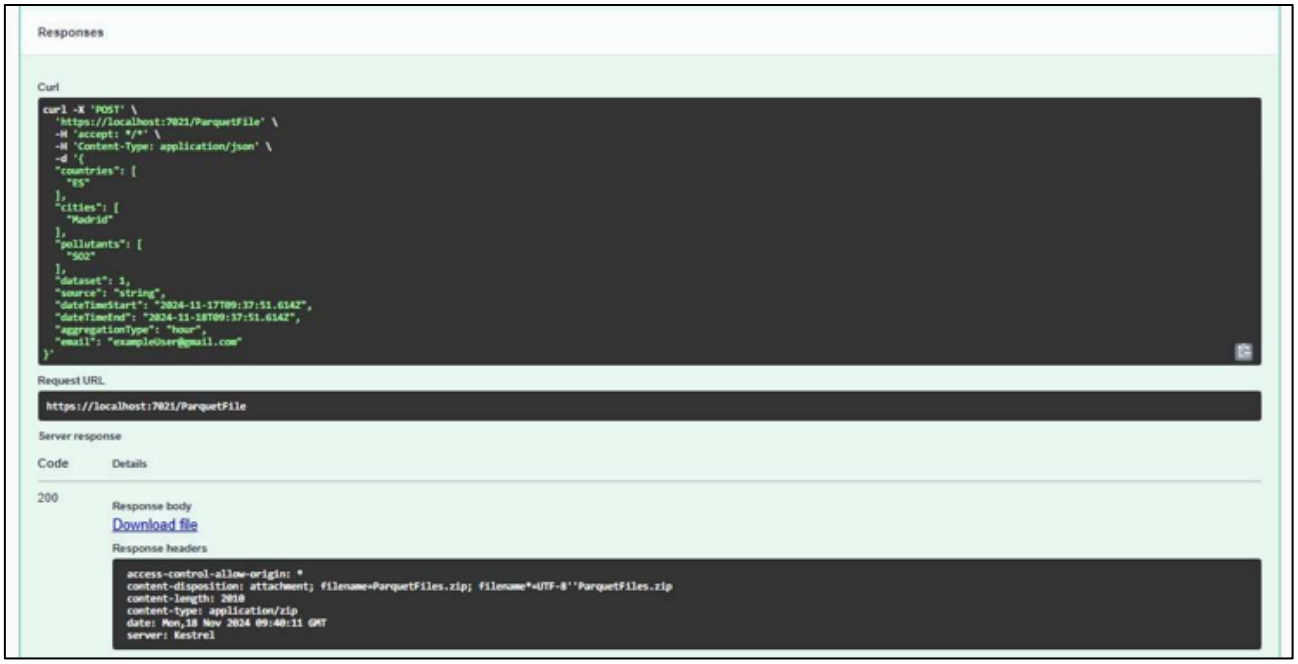
2. Daily data.
3. Variable intervals (different than the previous observations such as weekly, monthly, etc

This parameter is ignored in ParquetFile/urls.

- **email:** Optional field to identify the user who make the download and improve the communication if problems are detected.



Finally, click on "Download file" to get a zipped file with all the downloaded parquet files.



It is worth mentioning that the DownloadSummary endpoint does not consider the datetime or aggregationType filters, even if they are provided.

2.1. Examples of scripts

Python script example

Below is an example of a Python script to download parquet files from Azure with certain filters. Keep in mind that the downloadPath and the fileName refer to where the downloaded file will be saved, so this is different for different users.

```
import requests

apiUrl = "https://eeadmz1-downloads-api-appservice.azurewebsites.net/"
endpoint = "ParquetFile"
downloadPath = "localPath\\"
fileName = "download_data.zip"

# Request body
request_body = {
    "countries": ["ES"],
    "cities": ["Madrid"],
    "pollutants": ["SO2"],
    "dataset": 1,
    "dateTimeStart": "2024-11-17T09:37:00.000Z",
    "dateTimeEnd": "2024-11-18T09:37:00.000Z",
    "aggregationType": "hour",
    "email": "my.own@email.com"
}

# A get request to the API
downloadFile = requests.post(apiUrl+endpoint, json=request_body).content

# Store in local path
output = open(downloadPath+fileName, 'wb')
output.write(downloadFile)
```

Python script example – async endpoint

This example uses the async endpoint, which is more suitable for larger requests. Notice that in this case, two requests are made. The first request starts the process, and returns a URL. This URL is the address where the output will become available once the process is complete. The script ends by polling the output URL until the download becomes available.

```
from datetime import datetime
import time
import requests

apiUrl = "https://eeadmz1-downloads-api-appservice.azurewebsites.net/"
endpoint = "ParquetFile/async"
downloadPath = ""
fileName = "EEA_2023.zip"
request_body = {
    "countries": ["IT"],
    "cities": [],
    "pollutants": ["NO2", "NOX as NO2", "PM10"],
```

```
    "dataset": 2,
    "dateTimeStart": "2023-01-01T00:00:00Z",
    "dateTimeEnd": "2023-01-05T23:59:59Z",
    "aggregationType": "day",
    "email": "my.own@email.com"
}

response = requests.post(f"{apiUrl}{endpoint}", json=request_body)
downloadFile = response.text
print(downloadFile)

t_start = datetime.now()
while True:
    if (datetime.now()-t_start).total_seconds() > 3600: # stop after 1 hour if
the file has not been created
        break

    parquetResponse = requests.get(downloadFile)
    if parquetResponse.status_code==404:
        time.sleep(20)
    else:
        break

with open(downloadPath + fileName, "wb") as fp:
    fp.write(parquetResponse.content)
```

Python example – list of URLs

The third python variant you can use is the list of URLs endpoint. This will return the list of raw data files for the filters selected. You can then download them and apply your own additional filters and transformations in your environment. This case is recommended for the heaviest requests and/or custom processing needs. The list of URLs endpoint is the most stable of all.

```
import os
import requests

apiUrl = "https://eeadmz1-downloads-api-appservice.azurewebsites.net/"
endpoint = "ParquetFile/urls"
downloadFolder = "./e1a"
request_body = {
    "countries": ["IT"],
    "cities": [],
    "pollutants": ["NO2", "NOX as NO2", "PM10"],
    "dataset": 2,
    "dateTimeStart": "2023-01-01T00:00:00Z",
    "dateTimeEnd": "2023-01-05T23:59:59Z",
    "aggregationType": "hour",
    "email": "my.own@email.com"
}

response = requests.post(f"{apiUrl}{endpoint}", json=request_body)
urls = response.text.split("\n")[1:]
```

```
os.makedirs(downloadFolder, exist_ok=True)
for url in urls:
    fileName = url.split("/")[-1]
    with open(f"{downloadFolder}/{fileName}", "wb") as fp:
        fp.write(requests.get(url).content)
```

Postman example

In this second example, how to obtain the parquet files through Postman is described.

Once postman is open, the first thing is to configure the POST call and enter the body with the desired information to be downloaded. It is important to set the body type as JSON.

When the call ends, something like the image below can be seen. Next, press the "Save response to file", which gives the option to save a zipped file with the corresponding parquets files.

2.2. Visualization of Parquet files

There are different online pages to be able to view the parquet files or convert them to CSV to be able to read them without any special tool. Parquet Viewer example of one of them.

3. Parquet schema definition

The Parquet files contain the most relevant information reported by the countries. In the files various attributes can be found and are shown below:

- **Samplingpoint:** Identifier known as "local id" of the sampling point. Has the country code prefix. This identifier is unique to each station and created by national reporters, therefore there is no vocabulary for this parameter.

- **Pollutant:** The pollutant identifier. Find more information in this link: https://dd.eionet.europa.eu/vocabulary/aq/pollutant

- **Start:** Beginning of the time interval in which the information has been reported. Format: yyyy-mm-dd H:M:S

- **End:** End of the time interval in which the information has been reported. Format: yyyy-mm-dd H:M:S

- **Value:** Numerical value that represents the measurement obtained for that pollutant in that time interval.

- **Unit:** Unit in which the measurement obtained is represented. Find more information in this link: https://dd.eionet.europa.eu/vocabulary/uom/concentration

- **AggType:** This is the primary observation used in the measurement and can be hour/day/var. It represents whether the data collected is obtaining the values in hourly, daily or variable intervals (intervals different than the previous observation such as weekly, monthly, etc.). Find more information in this link: https://dd.eionet.europa.eu/vocabulary/aq/primaryObservation

- **Validity:** Represents the validity of the measurement reported for the specific pollutant within a specified time interval. Find more information in this link: https://dd.eionet.europa.eu/vocabulary/aq/observationvalidity

- **Verification:** Represents the verification status of the measurement reported for that pollutant in that time interval. Find more information in this link: https://dd.eionet.europa.eu/vocabulary/aq/observationverification

- **ResultTime:** Represents the date and time in which the information of the file that contained the reported data was generated.

- **DataCapture:** The data capture associated with a primary observation. Percentage of the time for which a sample or observation has been taken.

- **FkObservationLog:** This column has no importance for the end user. This is a column for internal use.