

FDSN webservice

Summary

FDSN webservice is a standardized protocol making possible to retrieve data and metadata through a widely used web protocol (the one you are using every day to browse websites): *HTTP*.

Webservice deliver time series in miniseed format and metadata in FDSN StationXML format. In addition, some metadata information can be requested in text format. Data inventories are fully available through *station* webservice, from network to channel levels.

IPGP FDSN Web Services

URL: <http://ws.ipgp.fr/fdsnws/>

IPGP Data Center runs FDSN webservice version 1.1.

URL of available services are:

- *fdsn-station*: <http://ws.ipgp.fr/fdsnws/station/1/>
- *fdsn-dataselect*: <http://ws.ipgp.fr/fdsnws/dataselect/1/>

Choosing your webservice client

In computer network terminology, the webservice is available by requesting a *server*. The software tool requesting the server is called a *client*.

- Simple requests can be handled manually through your browser.
- Automated or more complex requests can be achieved using command-line programs such as *wget* or *curl* available under most *Unix-like* systems.
- *Application Programming Interfaces* (API) are available for programming languages such as *Python*, *Java*, or *Matlab*. APIs make possible to request data directly from your codes and to handle data as high-level objects.

Using your web browser

Using your favorite web browser, click on links below, or edit/add/remove any parameter directly in the address bar.

By default, the output format is stationXML.

List all networks hold at IPGP Data Center:

<http://ws.ipgp.fr/fdsnws/station/1/query?level=network>

List all stations:

<http://ws.ipgp.fr/fdsnws/station/1/query?level=station>

List all stations of the G network:

<http://ws.ipgp.fr/fdsnws/station/1/query?network=G&level=station>

Get the instrumental response of RER station:

<http://ws.ipgp.fr/fdsnws/station/1/query?network=G&station=RER&level=response>

Get miniseed data from channel TAM 00.BHZ for the specified period:

<http://ws.ipgp.fr/fdsnws/dataselect/1/query?network=G&station=TAM&location=00&channel=BHZ&starttime=2018-06-01T00:00:00&endtime=2018-06-02T00:00:00>

Using on-line commands

Using *wget* or *curl* on the console, you may easily automate data retrieval through a *shell* script:

```
wget -O TAM.00BHZ.mseed "http://ws.ipgp.fr/fdsnws/dataselect/1/query?network=G&station=TAM&location=00&channel=BHZ&starttime=2018-06-01T00:00:00&endtime=2018-06-02T00:00:00"
```

The requested miniseed data will be downloaded into file TAM.00BHZ.mseed.

Note: the ULR must be surrounded by quotation marks in order to prevent the shell interpreting the some reserved characters such as "&".

Querying using the POST method

In the examples above, request parameters are encoded into the URL, using *key=value* pairs and special symbol "&". This is known as the *GET* method. The *GET* method allows a straightforward approach for building simple queries and performing queries directly from your navigator address bar.

POST method allows submitting more complex queries, combining multiple channels or timeframes. For this, you need to edit a request file with your favorite text editor, or generate it within your script.

Example: the request file *mydatarequest.txt*, contains :

```
WI BIM 00 BHZ 2019-01-07T00:00:30.457000 2019-01-07T14:35:30.457000
WI BIM 00 HHZ 2019-01-07T00:00:30.457000 2019-01-07T14:35:30.457000
WI ABD 00 HHZ 2019-01-07T00:00:30.457000 2019-01-07T14:35:30.457000
WI ABD 00 HHN 2019-01-07T00:00:30.457000 2019-01-07T14:35:30.457000
WI ABD 00 HHE 2019-01-07T00:00:30.457000 2019-01-07T14:35:30.457000
```

submit the request with *wget*:

```
wget --post-file=mydatarequest.txt -O mytest.mseed "http://ws.ipgp.fr/fdsnws/dataselect/1/query"
```

POST method is the recommended approach when scripting your requests, because less error-prone with parameters syntax and encoding.

Using Python

You may use *Obspy* library (<https://docs.obspy.org/>) to natively handle seismic data into your *Python* scripts.

Example:

```
from obspy.fdsn import Client
from obspy import UTCDateTime
client = Client("http://ws.ipgp.fr")
t = UTCDateTime("2010-02-27T06:45:00.000")
st = client.get_waveforms("G", "TAM", "00", "BHZ", t, t + 60 * 60)
st.plot()
```