Data access @ IPGP-DC

Constanza Pardo, Claudio Satriano

IPGP Data Center

This work is licensed under a
Creative Commons Attribution 4.0 International License.

The IPGP Data Center

The IPGP Data Center

Manage and distribute geophysical data from IPGP to support the geophysical research community

Requirements:

- A long term archive
- High quality of continuous datasets
- Reliable and efficient access to the data
- Integration to National and International structures:
  - French Seismologic and Geodetic Network (Résif)
  - European Integrated Data Archive (EIDA/ORFEUS)
  - European Plate Observatory System (EPOS)
  - International Federation of Digital Seismograph Networks (FDSN, IRIS-DMC)
  - Planetary Data System (NASA)

CC BY

The IPGP Data Center

- Up to end of 2010: GEOSCOPE Data Center
  - **2008**: **VOLOBSIS Project**, specification for integration of IPGP Seismological and Volcanological Data

- 2011 - present: IPGP Data Center - http://datacenter.ipgp.fr

Engineers
Vincent Conejero
Cyril Guinet

Coordination
Constanza Pardo, technical leader
Claudio Satriano, scientific leader

datacenter@ipgp.fr

CC BY

The IPGP Data Center

Services provided

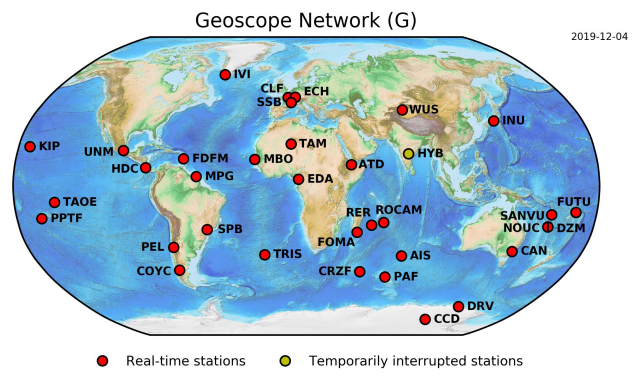
- Data management and long term archive
- Data distribution through standard interfaces
- Web site development and hosting
- Interconnection with other Data Centers and Portals
- Digital Object Identifier (DOI) declaration (through INIST)
- Data Formats expertise

CC BY

Data available

http://volobsis.ipgp.fr

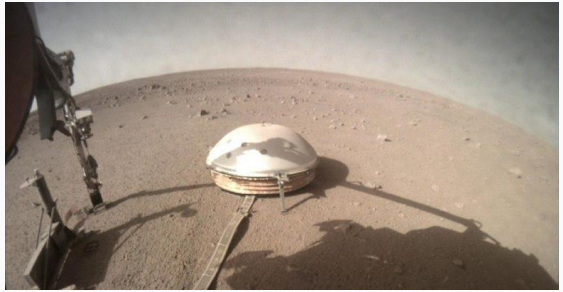
- GNSS
- Seismology (validated, in real-time)
- Volcanology
- Derived products:
  - Volcano reports
  - Events
  - DOI (data, products, observatories)



http://geoscope.ipgp.fr

- Seismology
- Strong motion
- Temperature, pressure
- Data validated and in real-time
- Derived products:
  - earthquake information
  - PZ & response files
  - DOI: **10.18715/GEOSCOPE.G**

InSight Mission - XB 2016-2022



https://www.seis-insight.eu/en/science

- Seismology:
  - Scientific
  - Housekeeping
- Weather (soon)
- Derived products:
  - Event catalogues
  - DOI: **10.18715/SEIS.INSIGHT.XB_2016**
- SEIS Ground Segment
- Private services: Mars SEIS Data Service (MSDS) and SEIS Data Portal (SDP)

CC BY

How cite the data

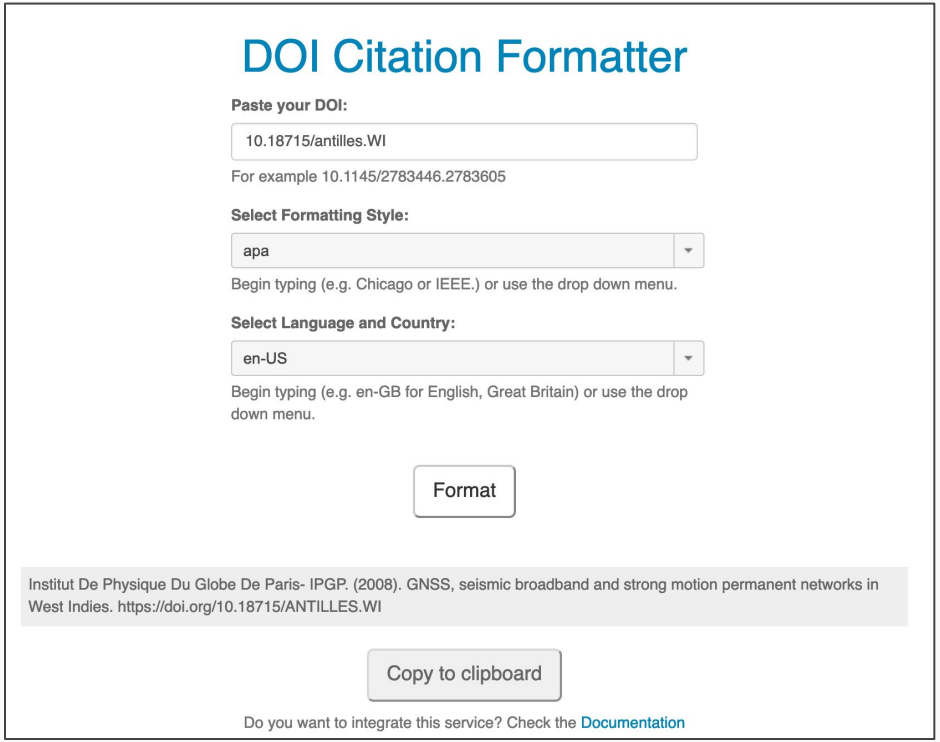
DOIs assigned by Network:

10.18715/GEOSCOPE.G
10.18715/guadeloupe.GL
10.18715/martinique.MQ
10.18715/reunion.PF
10.18715/SEIS.INSIGHT.XB_2016
10.18715/antilles.WI

Formatting Citations:

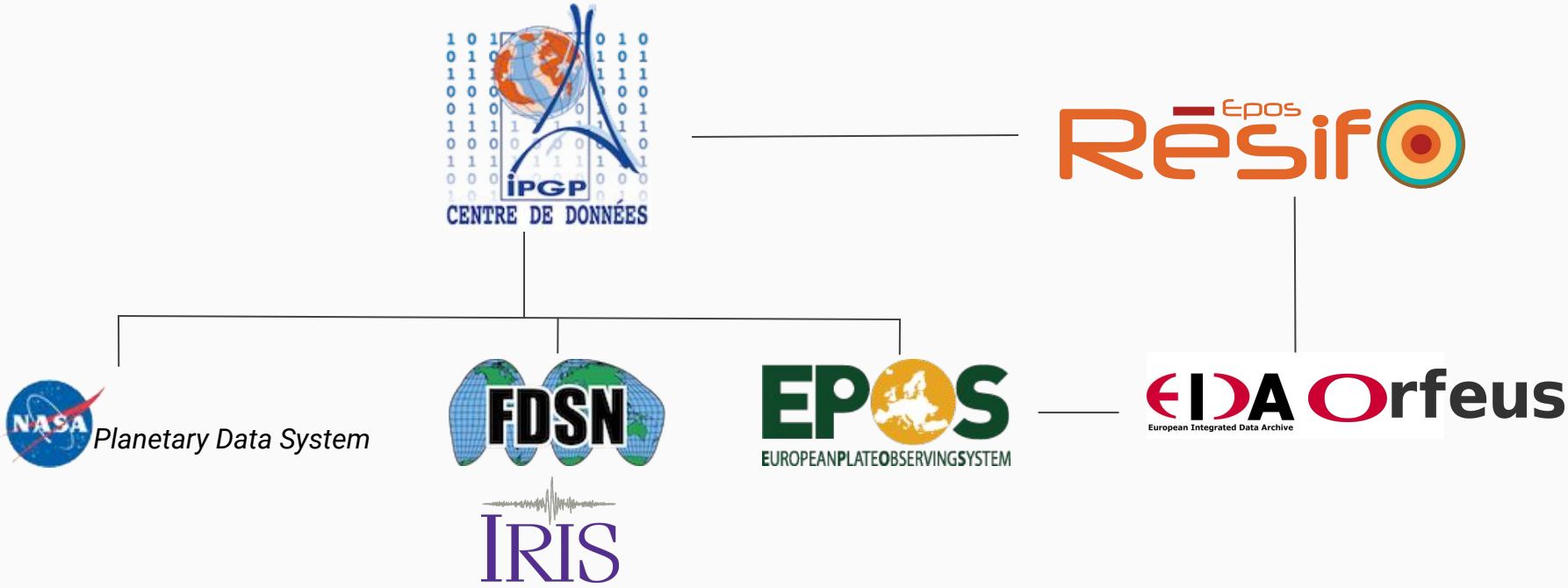
https://citation.crosscite.org

https://www.doi2bib.org



CC BY

French / European / World Data Centers



CC BY

Part 1 : Seismological and volcanological data

Data Formats

SEED Data Format

1987 : International Federation of Digital Seismograph Networks (FDSN)

The **SEED**, **S**tandard for the **E**xchange of **E**arthquake **D**ata adopted

http://www.fdsn.org/pdf/SEEDManual_V2.4.pdf

Data is separately managed or used:

- Metadata: station and instrument information
- Time series (data only records) with “some metadata” (network, station, location, channel, encoding, clock information, etc….)
- Continuous and event data

A dataset is associated to a Network (FDSN) and a station and refers to different channels (streams)

CC BY

SEED Channel Naming

LOC (2 letters) CHANNEL (3 letters)

00 **BHE**

"Acquisition" Band Instrument Orientation

Bande Codes (*)

E	Extremely Short Period	>= 80 to < 250
S	Short Period	>= 10 to < 80
H	High Broad Band	>= 80 to < 250
B	Broad Band	>= 10 to < 80
L	Long Period	= 1
V	Very Long Period	~ 0.1
.....		
Q	Greater than 10 days	

(*) non exhaustive

Orientation Codes (*)

Z N E Traditional
(Vertical, North-South, East-West)
I O (Inside/Outside)
NONE

Instrument (*)	Code
High Gain and Short Period Seismometers	H
Low Gain Seismometer	L
Mass Position Seismometer	M
Strong Motion	N
Temperature	K
Pressure	D
Magnetometer	F
Wind	W
Tilt Meter	A
Non-specific instruments	Y

CC BY

The StationXML Metadata Format

Metadata:

- Station information: network, station, coordinates, remarks
- Instrument responses

The purpose of the FDSN StationXML schema is to define an XML representation of the most important and commonly used structures of SEED 2.4 metadata with enhancements (http://www.fdsn.org/xml/station/)

CC BY

Web Services

Data access before web services

- GEOSCOPE: A CD-ROM sent on demand (before 2000)
- NetDC - Networked Data Centers (1999-2016)
  - request sent by mail, data access by mail and ftp protocols.
  - SEED data, inventory and RESP files
  - Data available in networked data centers
- Arclink (GFZ/EIDA) (2010 - 2019)
  - Request and access to data by TCP/IP
  - Dataless SEED, miniSEED
  - Standalone and EIDA
- FDSN Web Service specifications Version 1.0 (2013)

CC BY

What is a web service?

A **web service** is an **application** or **data source** that is accessible via a **standard web protocol** (HTTP or HTTPS). Unlike web applications, web services are **designed to communicate with other programs**, rather than directly with users.

While web services can provide data in a number of different formats, XML and JSON are the most common. These standard text-based formats can be easily recognized and parsed by another program that receives the data.

(source: https://techterms.com/definition/web_service)

Example:
http://www.geoplugin.net/extras/location.gp?lat=48.8448&lon=2.3565

CC BY

What is a web service?

A web service is not a web application (or a web site)!

It is a way of retrieving data from a server using the HTTP protocol.

CC BY

Why use a webservice?

Advantages:

- Uses standard web ports (typically 80, 8080 or 443) which are normally open by institute/corporate firewalls;
- Platform agnostic: only needs a web browser (or widely available command line tools like **wget** or **curl**) to access the service;
- Scriptable: can be used as basis for more complex workflows/applications.

Drawbacks:

- Difficult to grasp for beginners (that's why we're here 😉)
- Not always a "URL builder" is provided

CC BY

Web services for data science

- Easy way to discover and access Big Data
- Query interface. Data discovery and selection based on:
  - Geographic bounds
  - Time span
  - Recording sensor name or type
  - …
- Discover data on the server, before downloading it!

CC BY

Anatomy of a web service

https://ws.ipgp.fr	/fdsnws	/dataselect	/1	/query?
Base URL	Service Family	Service Name	Service Version	Method

Examples:

https://ws.ipgp.fr/fdsnws/dataselect/1/query?starttime=2020-01-01T00:00:00&endtime=2020-01-02T00:00:00&network=G&station=SSB&location=00&channel=BHZ

https://ws.resif.fr/resifws/timeseries/1/query?net=RA&station=PYTO&cha=HN2&loc=02&demean&correct&start=2017-11-02T13:35:00&end=2017-11-02T13:40:00&format=plot

Note that query parameters are separated by "&"

CC BY

FDSN web services

Web services to access common data types in Seismology:

- fdsnws-station – For access to station metadata in FDSN StationXML format
- fdsnws-dataselect – For access to time series data in miniSEED format
- fdsnws-availability – For access to time series data availability
- fdsnws-event – For access to event parameters in QuakeML format

Standardized by FDSN

References:

- Official site and specification: https://www.fdsn.org/webservices/
- IRIS implementation: https://service.iris.edu/fdsnws/

CC BY

FDSN web services @ IPGP-DC

Currently available at: https://ws.ipgp.fr

- fdsnws-station – For access to station metadata in FDSN StationXML format
- fdsnws-dataselect – For access to time series data in miniSEED format
- fdsnws-event – For access to event parameters in QuakeML format
- fdsnws-availability – For access to time series data availability (soon)

CC BY

fdsnws-station

https://ws.ipgp.fr	/fdsnws	/station	/1	/query?

Parameter	Alias	Support	Default	Allowed Values	Type	Unit
starttime	**start**	**required**	**[Any]**	**Any valid time**	**time**	**UTC**
Limit results to time series samples on or after the specified start time.						
endtime	**end**	**required**	**[Any]**	**Any valid time**	**time**	**UTC**
Limit results to time series samples on or before the specified end time.						
startbefore		**optional**	**[Any]**	**Any valid time**	**time**	**UTC**
Limit to metadata epochs starting before specified time.						
startafter		**optional**	**[Any]**	**Any valid time**	**time**	**UTC**
Limit to metadata epochs starting after specified time.						
endbefore		**optional**	**[Any]**	**Any valid time**	**time**	**UTC**
Limit to metadata epochs ending before specified time.						
endafter		**optional**	**[Any]**	**Any valid time**	**time**	**UTC**
Limit to metadata epochs ending after specified time.						
network	**net**	**required**	**[Any]**	**Valid network code or wildcard**	**string**	
Select one or more network codes. Can be FDSN network codes or data center defined codes. Multiple codes are comma-separated.						
station	**sta**	**required**	**[Any]**	**Valid station code or wildcard**	**string**	
Select one or more FDSN station codes. Multiple codes are comma-separated.						
location	**loc**	**required**	**[Any]**	**Valid location code or wildcard**	**string**	
Select one or more FDSN location identifiers. Multiple identifiers are comma-separated. As a special case '--' (two dashes) will be translated to a string of two space characters to match blank location IDs.						
channel	**cha**	**required**	**[Any]**	**Valid channel code or wildcard**	**string**	
Select one or more FDSN channel codes. Multiple codes are comma-separated.						

Source: https://www.fdsn.org/webservices/

CC BY

fdsnws-station

https://ws.ipgp.fr	/fdsnws	/station	/1	/query?

Parameter	Alias	Support	Default	Allowed Values	Type	Unit
minlatitude	**minlat**	**required**	**-90.0**	**-90.0 to 90.0**	**float**	**degrees**
	Limit to stations with a latitude larger than or equal to the specified minimum.					
maxlatitude	**maxlat**	**required**	**90.0**	**-90.0 to 90.0**	**float**	**degrees**
	Limit to stations with a latitude smaller than or equal to the specified maximum.					
minlongitude	**minlon**	**required**	**-180.0**	**-180.0 to 180.0**	**float**	**degrees**
	Limit to stations with a longitude larger than or equal to the specified minimum.					
maxlongitude	**maxlon**	**required**	**108.0**	**-180.0 to 180.0**	**float**	**degrees**
	Limit to stations with a longitude smaller than or equal to the specified maximum.					
latitude	**lat**	**optional**	**0.0**	**-90.0 to 90.0**	**float**	**degrees**
	Specify the latitude to be used for a radius search.					
longitude	**lon**	**optional**	**0.0**	**-180.0 to 180.0**	**float**	**degrees**
	Specify the longitude to be used for a radius search.					
minradius		**optional**	**0.0**	**0 to 180.0**	**float**	**degrees**
	Limit results to stations within the specified minimum number of degrees from the geographic point defined by the latitude and longitude parameters.					
maxradius		**optional**	**180.0**	**0 to 180.0**	**float**	**degrees**
	Limit results to stations within the specified maximum number of degrees from the geographic point defined by the latitude and longitude parameters.					
level		**required**	**station**	**network, station, channel, response**	**string**	
	Specify the level of detail for the results.					
includerestricted		**optional**	**TRUE**	**TRUE or FALSE**	**boolean**	
	Specify if results should include information for restricted stations.					

Source: https://www.fdsn.org/webservices/

CC BY

fdsnws-station

https://ws.ipgp.fr	/fdsnws	/station	/1	/query?

Parameter	Alias	Support	Default	Allowed Values	Type	Unit
includeavailability		**optional**	**FALSE**	**TRUE or FALSE**	**boolean**	
	Specify if results should include information about time series data availability.					
updatedafter		**optional***	**[Any]**	**Any valid time**	**time**	**UTC**
	Limit to metadata updated after specified time; updates are data center specific. * While this option is not required it is highly recommended due to usefulness.					
matchtimeseries		**optional**	**FALSE**	**TRUE or FALSE**	**boolean**	
	Limit to metadata where selection criteria matches time series data availability.					
format		**optional**	**text**	**xml, text**	**string**	
	Specify format of result, either ***xml*** (default) or ***text*** (defined below). If this parameter is not specified the service must return StationXML.					
nodata		**optional**	**204**	**204 or 404**	**string**	
	Select status code for "no data", either ***204*** (default) or ***404***.					

Source: https://www.fdsn.org/webservices/

CC BY

fdsnws-station

Note that query parameters are separated by "&"

Examples

https://ws.ipgp.fr/fdsnws/station/1/query?starttime=2020-01-01&endtime=2020-01-02&network=G&format=text

Non sicuro — ws.ipgp.fr/fdsnws/station/1/query?starttime=2020-01-01&

```
#Network|Station|Latitude|Longitude|Elevation|SiteName|StartTime|EndTime
G|AIS|-37.79635|77.569186|35.9|Nouvelle-Amsterdam - TAAF, France|1993-12-25T00:00:00|
G|ATD|11.530725|42.846634|610.0|Arta Cave - Arta, Republic of Djibouti|1993-07-07T00:00:00|
G|CAN|-35.318715|148.996325|700.0|Canberra, Australia|1987-11-27T00:00:00|
G|CCD|-75.1065|123.305|3240.0|Concordia Base, Dome C, Antartica|2007-01-01T00:00:00|
G|CLF|48.02579|2.26|145.0|Chambon la Foret Observatory, France|2008-07-10T00:00:00|
G|COYC|-45.57299|-72.08139|235.0|Coyhaique, Chile|2004-12-17T00:00:00|
G|CRZF|-46.43096|51.855308|140.0|Port Alfred - Ile de la Possession - Crozet Islands, France|1986-02-01T00:00:00|
G|DRV|-66.664908|140.002069|40.0|Dumont d'Urville - Terre Adelie, Antarctica|1986-02-01T00:00:00|
G|DZM|-22.07164|166.44377|878.0|Dzumac - New Caledonia, France|2003-09-01T00:00:00|
G|ECH|48.216313|7.158961|580.0|Echery - Sainte Marie aux Mines, France|1990-11-08T00:00:00|
G|EDA|3.778868|10.153427|140.0|Edea, Cameroon|2019-05-04T00:00:00|
G|FDFM|14.73493|-61.159728|320.0|Morne la Rosette, Martinique, France|2019-10-24T00:00:00|
G|FOMA|-24.97565|46.978877|28.0|Nahampoana Reserve - Fort Dauphin, Madagascar|2008-09-01T00:00:00|
G|FUTU|-14.307673|-178.12107|97.0|Maopo'opo, Futuna|2016-06-26T00:00:00|
G|HDC|10.00201|-84.11142|1186.0|Heredia, Costa Rica|1997-03-08T00:00:00|
G|HYB|17.41867|78.55213|510.0|Hyderabad, India|1989-01-15T00:00:00|
G|INU|35.35|137.029|132.0|Inuyama, Japan|1987-03-04T00:00:00|
G|IVI|61.205808|-48.171178|15.0|Ivittuut, Greenland|2011-09-12T00:00:00|
G|KIP|21.42|-158.0112|110.0|Kipapa, Hawaii, USA|1986-04-17T00:00:00|
G|MBO|14.392024|-16.955467|3.0|M'Bour, Senegal|1985-09-01T00:00:00|
G|MPG|5.11011|-52.644483|147.0|Montagne des Peres - French Guiana, France|2000-07-03T00:00:00|
G|NOUC|-22.09863|166.30665|137.0|Port Laguerre - New Caledonia, France|1988-03-21T00:00:00|
G|PAF|-49.351|70.210708|17.0|Port aux Francais - Kerguelen - TAAF, France|1983-01-01T00:00:00|
G|PEL|-33.143607|-70.67494|700.0|Peldehue, Chile|1995-10-04T00:00:00|
G|PPTF|-17.5896|-149.56525|705.0|Pamatai - Papeete - Tahiti island - French Polynesia, France|2009-04-25T00:00:00|
G|RER|-21.1712|55.73986|834.0|Riviere de l'Est - Sainte Rose - La Reunion island, France|1986-02-10T00:00:00|
G|ROCAM|-19.75553|63.37014|52.0|Leguat Reserve, Rodrigues Island, Republic of Mauritius|2012-12-15T00:00:00|
G|SANVU|-15.447148|167.203231|56.0|Espiritu Santo, Vanuatu|2011-11-01T00:00:00|
G|SPB|-23.592691|-47.426979|647.0|Sao Paulo, Brazil|1996-06-17T00:00:00|
G|SSB|45.279|4.542|700.0|Tunnel de Badole - Saint Sauveur en Rue, France|1982-05-02T00:00:00|
G|TAM|22.79149|5.52838|1410.0|Tamanrasset, Algeria|1983-11-16T00:00:00|
G|TAOE|-8.85492|-140.14775|800.0|Taiohae - Marquesas islands, France|2004-11-01T20:00:00|
G|TRIS|-37.0681|-12.3152|60.0|Tristan da Cunha|2004-03-03T00:00:00|
G|UNM|19.329662|-99.178065|2280.0|Unam - Mexico, Mexico|1990-06-06T00:00:00|
G|WUS|41.200716|79.216498|1468.0|Wushi - Xinjiang Uygur, China|1988-10-31T00:00:00|
```

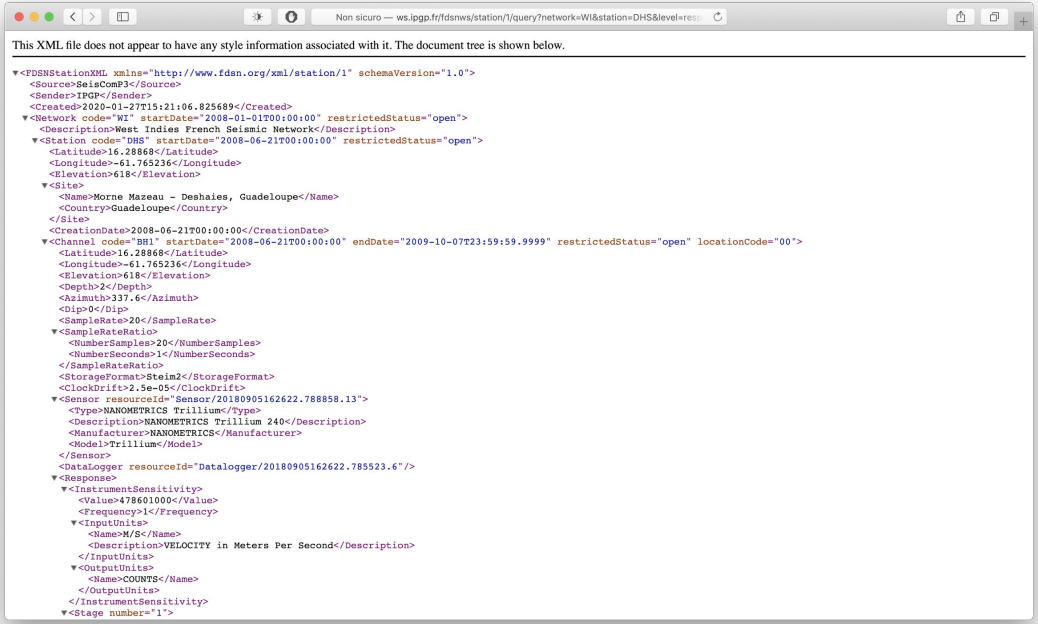
CC BY

fdsnws-station

Note that query parameters are separated by "&"

Examples

https://ws.ipgp.fr/fdsnws/station/1/query?network=WI&station=DHS&level=response



CC BY

fdsnws-station

Note that query parameters are separated by "&"

Examples

https://ws.ipgp.fr/fdsnws/station/1/query?network=WI&station=DHS&level=response

Or, in terminal, using wget:

Note that special characters are escaped by "\"

```
$ wget https://ws.ipgp.fr/fdsnws/station/1/query\?network\=WI\
&station\=DHS\&level\=response -O WI.DHS.xml
```

...or curl:

```
$ curl https://ws.ipgp.fr/fdsnws/station/1/query\?network\=WI\
&station\=DHS\&level\=response > WI.DHS.xml
```

CC BY

fdsnws-station

Using ObsPy www.obspy.org

ObsPy
A Python Framework for Seismology

```
$ python
>>> from obspy.clients.fdsn import Client
>>> cl = Client('https://ws.ipgp.fr')
>>> inv = cl.get_stations(network='WI', station='DHS', level='response')
>>> inv.write('WI.DHS.xml', format='StationXML')
```

Documentation: https://docs.obspy.org/packages/obspy.clients.fdsn.html

CC BY

fdsnws-station

Using FetchMetadata

https://seiscode.iris.washington.edu/projects/ws-fetch-scripts

```
$ export SERVICEBASE=http://ws.ipgp.fr
$ FetchMetadata -resp -N WI -S DHS -X WI.DHS.xml
```

Documentation: https://seiscode.iris.washington.edu/projects/ws-fetch-scripts/wiki

CC BY

Manipulating a stationXML file

Let's check the content of stationXML file using ObsPy

```
$ python
>>> from obspy import read_inventory
>>> inv = read_inventory('WI.DHS.xml')
>>> print(inv)
Inventory created at 2020-01-29T14:40:05.256924Z
    Created by: None
            None
    Sending institution: SeisComP3 (IPGP)
    Contains:
        Networks (1):
            WI
        Stations (1):
            WI.DHS (Morne Mazeau - Deshaies, Guadeloupe)
        Channels (42):
            WI.DHS.00.BHZ (4x), WI.DHS.00.BHN, WI.DHS.00.BHE,
            WI.DHS.00.BH1 (3x), WI.DHS.00.BH2 (3x), WI.DHS.00.HHZ (4x),
            WI.DHS.00.HHN, WI.DHS.00.HHE, WI.DHS.00.HH1 (3x), ...
```

Manipulating a stationXML file

Let's check the content of stationXML file using ObsPy

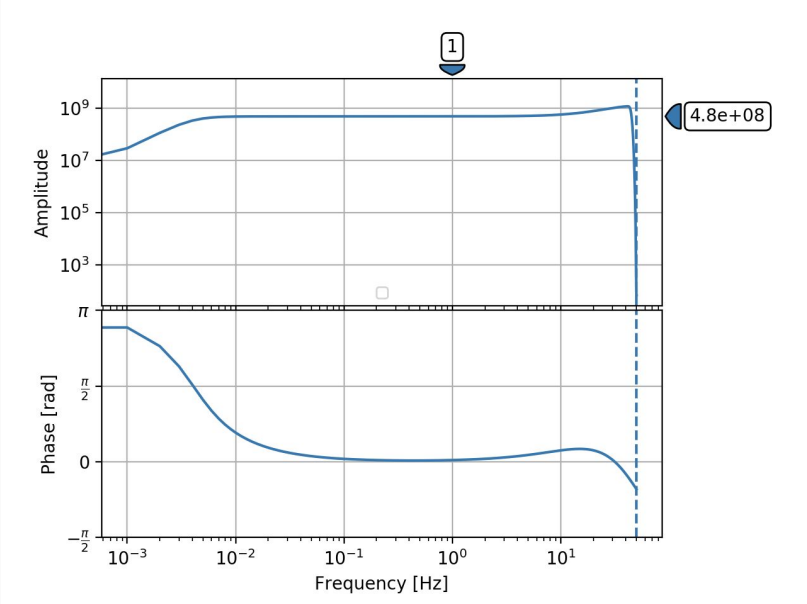
```
>>> from obspy import UTCDateTime
>>> t0 = UTCDateTime('20200101')
>>> print(inv.get_response('WI.DHS.00.HHZ', datetime=t0))
Channel Response
    From M/S (VELOCITY in Meters Per Second) to COUNTS ()
    Overall Sensitivity: 4.78601e+08 defined at 1.000 Hz
    7 stages:
        Stage 1: PolesZerosResponseStage from M/S to V, gain: 1196.5
        Stage 2: PolesZerosResponseStage from V to V, gain: 1
        Stage 3: PolesZerosResponseStage from V to V, gain: 0.4
        Stage 4: CoefficientsTypeResponseStage from V to COUNTS, gain: 1e+06
        Stage 5: FIRResponseStage from COUNTS to COUNTS, gain: 1
        Stage 6: FIRResponseStage from COUNTS to COUNTS, gain: 1
        Stage 7: FIRResponseStage from COUNTS to COUNTS, gain: 1
```

CC BY

Manipulating a stationXML file

Let's check the content of stationXML file using ObsPy

```
>>> inv.get_response('WI.DHS.00.HHZ', datetime=t0).plot(min_freq=0.001)
```



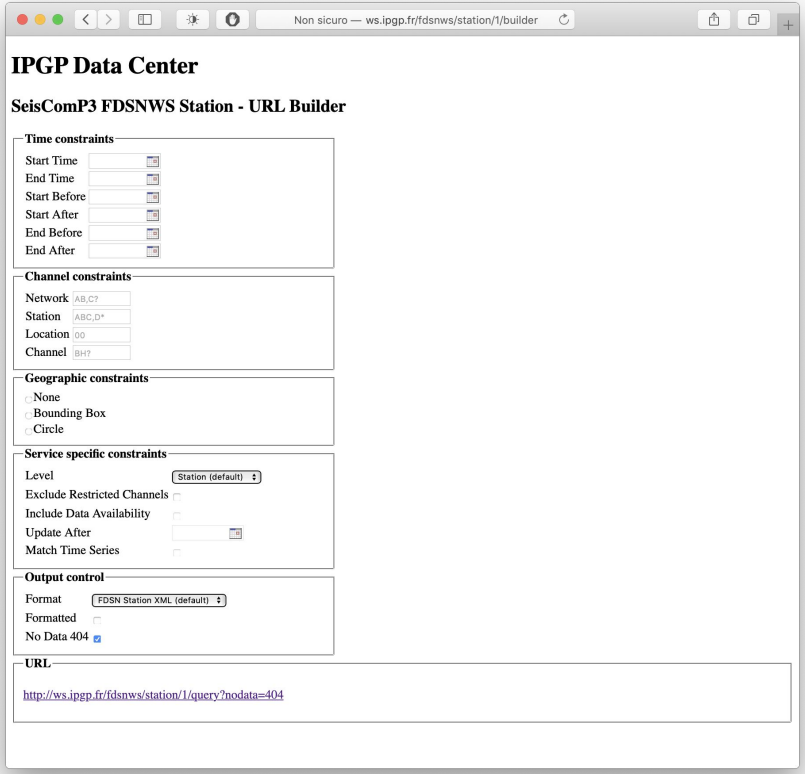
CC BY

fdsnws-**station**: query builder

https://ws.ipgp.fr/fdsnws/station/1/builder

A web interface to build the web service URL for a given request.

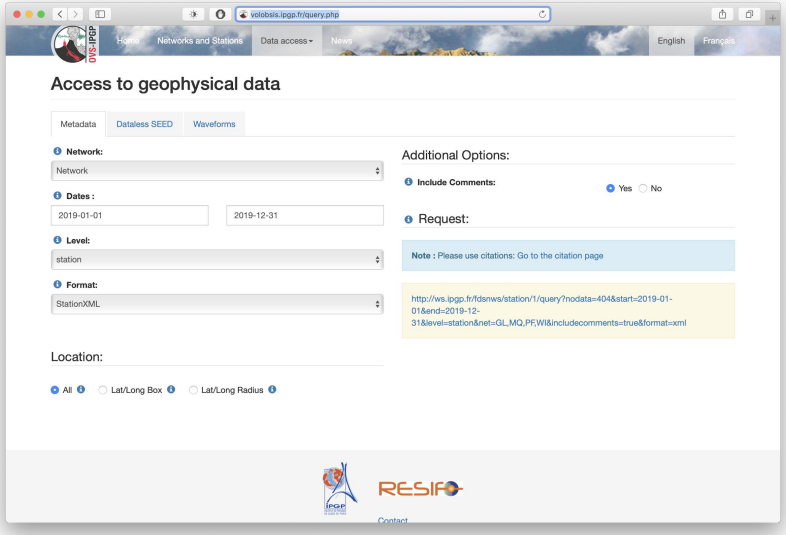
Provided by SeisComP3 software.



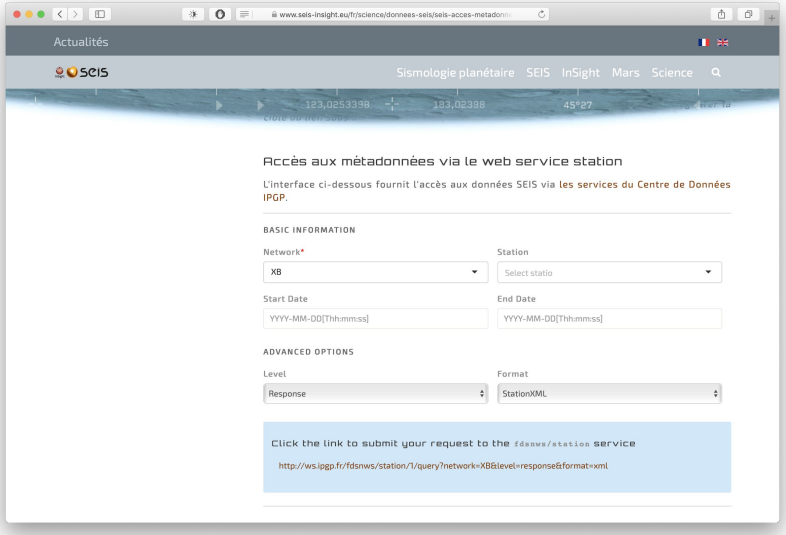
CC BY

fdsnws-**station**: query builder

Other builders provided by IPGP-DC:



http://volobsis.ipgp.fr/query.php



https://www.seis-insight.eu/fr/science/donnees-seis/seis-acces-metadonnees

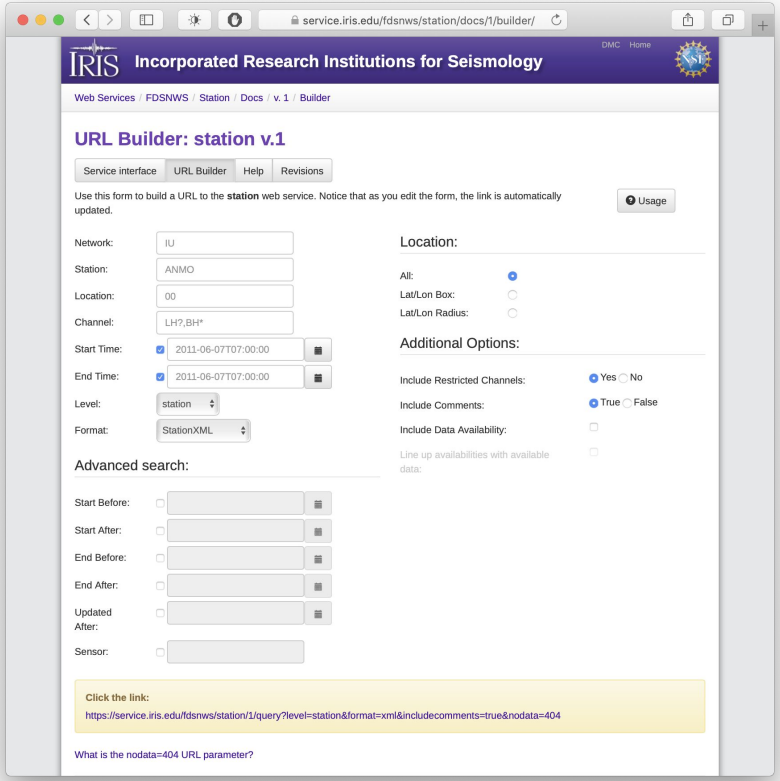
CC BY

fdsnws-**station**: query builder

You can always use the builder provided by IRIS-DMC:

https://service.iris.edu/fdsnws/station/docs/1/builder/

(And, if needed, change the base URL)



CC BY

fdsnws-dataselect

https://ws.ipgp.fr	/fdsnws	/dataselect	/1	/query?

Parameter	Alias	Support	Default	Allowed Values	Type	Unit
starttime	**start**	**required**	**[Any]**	**Any valid time**	**time**	**UTC**
Limit results to time series samples on or after the specified start time						
endtime	**end**	**required**	**[Any]**	**Any valid time**	**time**	**UTC**
Limit results to time series samples on or before the specified end time						
network	**net**	**required**	**[Any]**	**Valid network code or wildcard**	**string**	
Select one or more network codes. Can be SEED network codes or data center defined codes. Multiple codes are comma-separated.						
station	**sta**	**required**	**[Any]**	**Valid station code or wildcard**	**string**	
Select one or more SEED station codes. Multiple codes are comma-separated.						
location	**loc**	**required**	**[Any]**	**Valid location code or wildcard**	**string**	
Select one or more SEED location identifiers. Multiple identifiers are comma-separated. As a special case '--' (two dashes) will be translated to a string of two space characters to match blank location IDs.						
channel	**cha**	**required**	**[Any]**	**Valid channel code or wildcard**	**string**	
Select one or more SEED channel codes. Multiple codes are comma-separated.						
quality		**optional**	**B**	**D, R, Q, M or B**	**string**	
Select a specific SEED quality indicator, handling is data center dependent.						
minimumlength		**optional**	**0.0**	**>= 0.0**	**float**	
Limit results to continuous data segments of a minimum length specified in seconds.						
longestonly		**optional**	**FALSE**	**TRUE or FALSE**	**boolean**	
Limit results to the longest continuous segment per channel.						
format		**optional**	**miniseed**	**miniseed**	**string**	
Specify format of result, the default value is ***miniseed***. If this parameter is not specified the service must return miniSEED.						
nodata		**optional**	**204**	**204 or 404**	**string**	
Select status code for "no data", either ***204*** (default) or ***404***.						

Source: https://www.fdsn.org/webservices/

CC BY

fdsnws-dataselect

Note that query parameters are separated by "&"

Examples

https://ws.ipgp.fr/fdsnws/dataselect/1/query?starttime=2019-11-26T02:00:00&endtime=2019-11-26T05:00:00&network=G&station=SSB&channel=HH?

dataselect returns by default a miniSEED file!

A data center may choose to return the results in alternate formats as specified by the "format" parameter (e.g., iris-dmc provides: SAC, GeoCSV)

CC BY

fdsnws-dataselect

Or, in terminal, using wget:

Note that special characters are escaped by "\"

```
$ wget https://ws.ipgp.fr/fdsnws/dataselect/1/query\?
starttime\=2019-11-26T02:00:00\&endtime\=2019-11-26T05:00:00\
&network\=G\&station\=SSB\&channel\=HH\? -O G.SSB.HH.mseed
```

...or curl:

```
$ curl https://ws.ipgp.fr/fdsnws/dataselect/1/query\?
starttime\=2019-11-26T02:00:00\&endtime\=2019-11-26T05:00:00\
&network\=G\&station\=SSB\&channel\=HH\? > G.SSB.HH.mseed
```

CC BY

fdsnws-dataselect

Using ObsPy
www.obspy.org

ObsPy
A Python Framework for Seismology

```
$ python
>>> from obspy.clients.fdsn import Client
>>> from obspy import UTCDateTime
>>> cl = Client('https://ws.ipgp.fr')
>>> st = cl.get_waveforms(
            starttime=UTCDateTime('2019-11-26T02:00:00'),
            endtime=UTCDateTime('2019-11-26T05:00:00'),
            network='G', station='SSB', location='*', channel='HH?')
>>> st.write('G.SSB.HH.mseed', format='MSEED')
```

Documentation: https://docs.obspy.org/packages/obspy.clients.fdsn.html

CC BY

fdsnws-dataselect

Using FetchData

https://seiscode.iris.washington.edu/projects/ws-fetch-scripts

```
$ export SERVICEBASE=http://ws.ipgp.fr
$ FetchData -s 2019-11-26T02:00:00 -e 2019-11-26T05:00:00 -N G -S SSB -C HH\?
  -O G.SSB.HH.mseed
```

Documentation: https://seiscode.iris.washington.edu/projects/ws-fetch-scripts/wiki

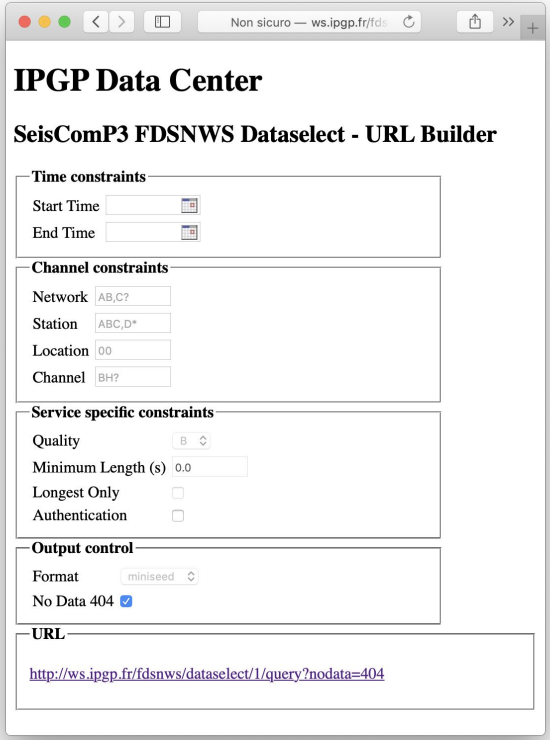
CC BY

fdsnws-**dataselect**: query builder

https://ws.ipgp.fr/fdsnws/dataselect/1/builder

A web interface to build the web service URL for a given request.

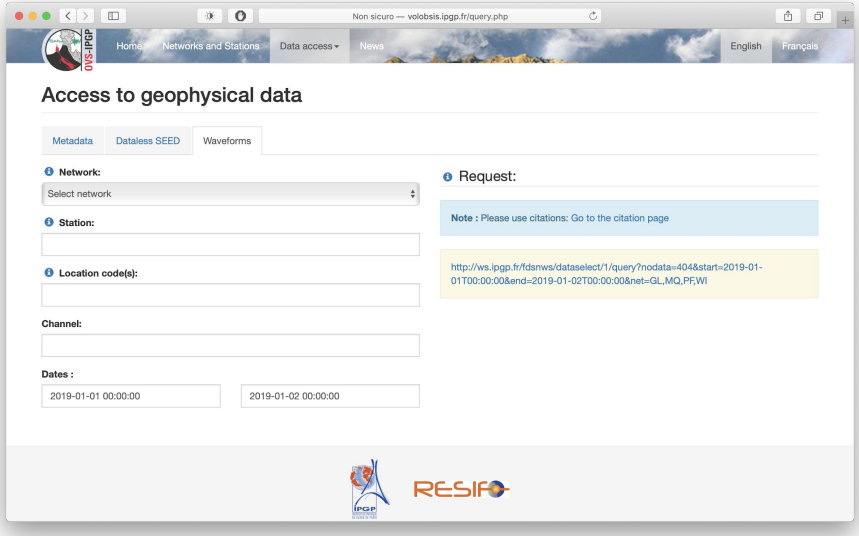
Provided by SeisComP3 software.

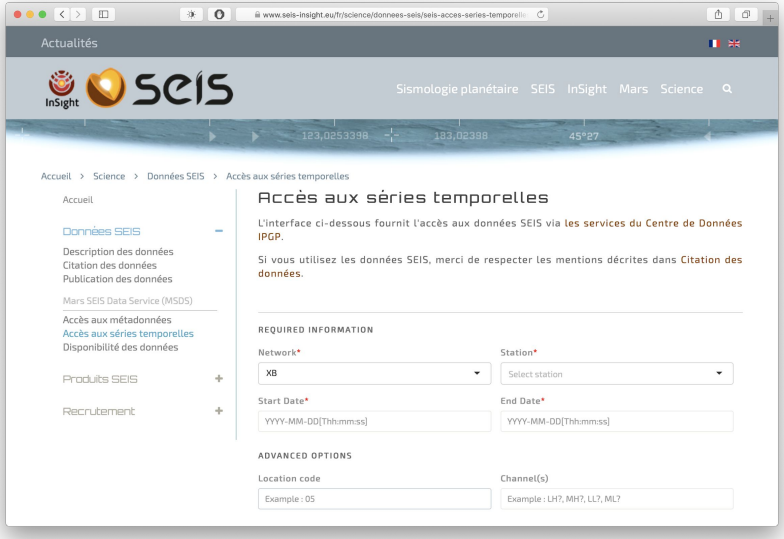


CC BY

fdsnws-**dataselect**: query builder

Other builders provided by IPGP-DC:





http://volobsis.ipgp.fr/query.php

https://www.seis-insight.eu/fr/science/donnees-seis/seis-acces-series-temporelles

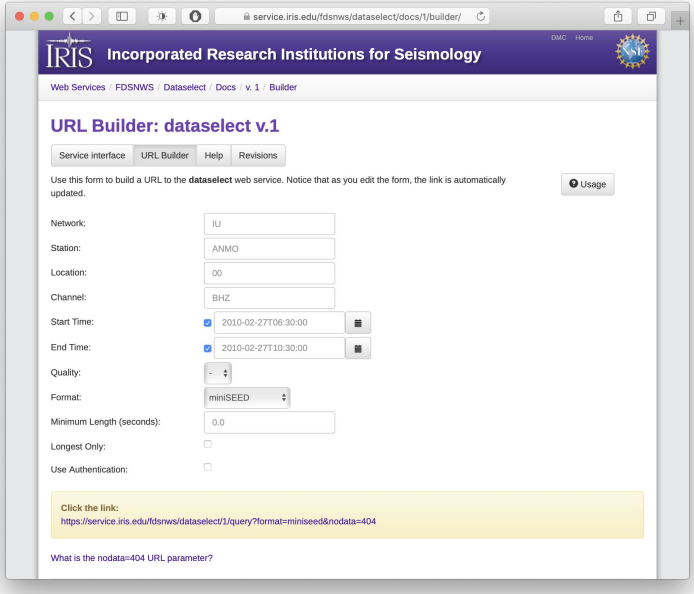
CC BY

fdsnws-**dataselect**: query builder

You can always use the builder provided by IRIS-DMC:

https://service.iris.edu/fdsnws/dataselect/docs/1/builder/

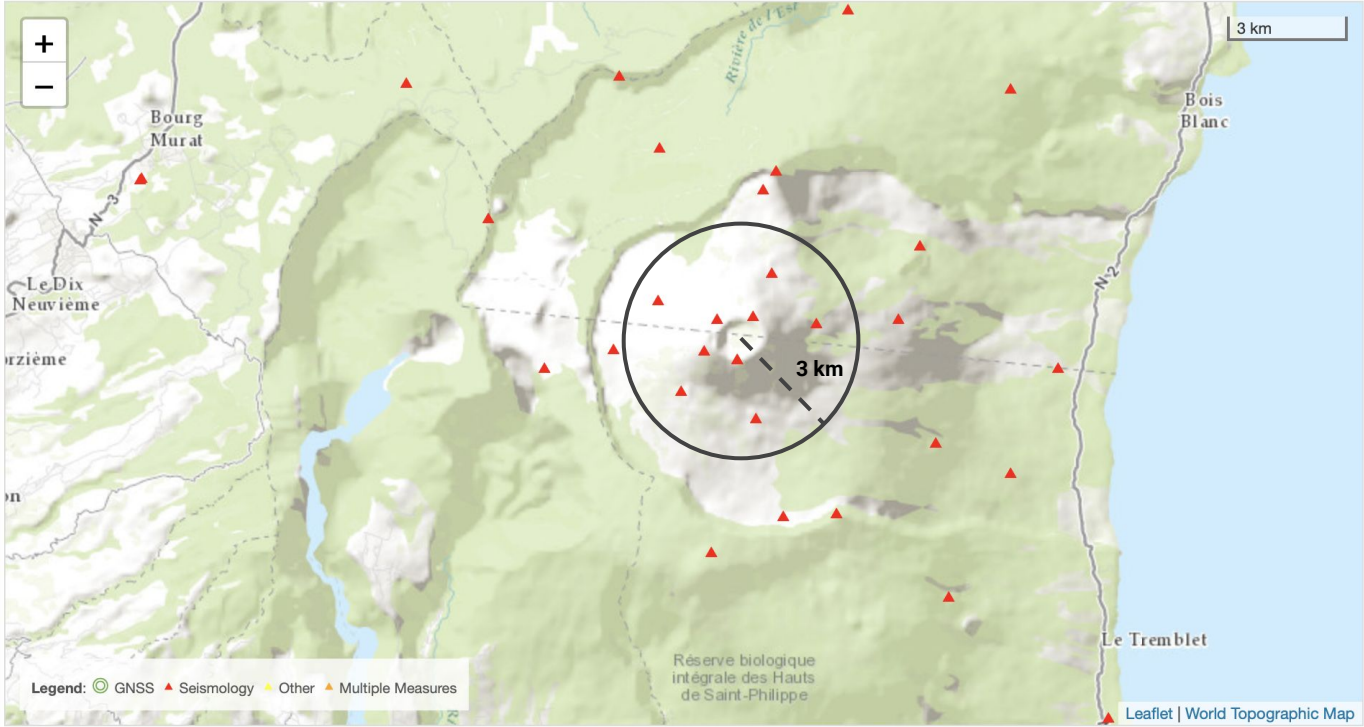
(And, if needed, change the base URL. But pay attention to options unsupported by your data center)



CC BY

Combining **station** and **dataselect**

Advanced example: Download data from seismological stations (HH and EH channels) within 3 km of the Piton de la Fournaise main crater.



CC BY

Combining **station** and **dataselect**

First part of the script: we use the **station** webservice to get a list of stations within 3 km from the crater.

```
#!/usr/bin/env bash

# Webservices URL
WSURL=https://ws.ipgp.fr/fdsnws

# Center coordinates to compute max radius
lat0=-21.244384
lon0=55.714093
# Max radius (in degrees, 1 degree = 111 km)
maxradius=0.027  # 3 km

# starttime and endtime
starttime=2019-12-01T00:00:00
endtime=2019-12-01T01:00:00

# Query station list
BASEURL="$WSURL/station/1/query?"
BASEURL+="latitude=$lat0&longitude=$lon0"
BASEURL+="&minradius=0&maxradius=$maxradius"
URL="$BASEURL&level=channel&format=text"

wget -nv $URL -O stations.txt
```

Combining **station** and **dataselect**

The output list is rich of information…

```
$ cat stations.txt
#Network|Station|Location|Channel|Latitude|Longitude|Elevation|Depth|Azimuth|Dip|SensorDescription|Sca
le|ScaleFreq|ScaleUnits|SampleRate|StartTime|EndTime
PF|BON|00|HHE|-21.23971|55.70921|2552.0|0.0|90.0|0.0|GURALP CMG-3ESPCDE
100S|733831000.0|1.0|M/S|100.0|2011-05-09T06:38:00|2011-12-21T12:45:00
PF|BON|00|HHE|-21.23971|55.70921|2552.0|0.0|90.0|0.0|GURALP CMG-3ESPCDE
100S|1838660000.0|1.0|M/S|100.0|2013-08-12T12:50:00|2015-06-17T04:47:59.9999
PF|BON|00|HHE|-21.23971|55.70921|2552.0|0.0|90.0|0.0|CMG40T, 60 s, 800 V/m/s-Taurus, 40 Vpp (gain
0.4),|319199000.0|1.0|M/S|100.0|2015-06-17T04:48:00|2016-04-20T07:25:00
PF|BON|00|HHE|-21.23971|55.70921|2552.0|0.0|90.0|0.0|GURALP CMG-3ESPCDE
100S|618895000.0|1.0|M/S|100.0|2016-04-20T08:10:00|
PF|BON|00|HHN|-21.23971|55.70921|2552.0|0.0|0.0|0.0|GURALP CMG-3ESPCDE
100S|734455000.0|1.0|M/S|100.0|2011-05-09T06:38:00|2011-12-21T12:45:00
PF|BON|00|HHN|-21.23971|55.70921|2552.0|0.0|0.0|0.0|GURALP CMG-3ESPCDE
100S|1829720000.0|1.0|M/S|100.0|2013-08-12T12:50:00|2015-06-17T04:47:59.9999
PF|BON|00|HHN|-21.23971|55.70921|2552.0|0.0|0.0|0.0|CMG40T, 60 s, 800 V/m/s-Taurus, 40 Vpp (gain
0.4),|319999000.0|1.0|M/S|100.0|2015-06-17T04:48:00|2016-04-20T07:25:00
PF|BON|00|HHN|-21.23971|55.70921|2552.0|0.0|0.0|0.0|GURALP CMG-3ESPCDE
100S|613212000.0|1.0|M/S|100.0|2016-04-20T08:10:00|
PF|BON|00|HHZ|-21.23971|55.70921|2552.0|0.0|0.0|-90.0|GURALP CMG-3ESPCDE
100S|738819000.0|1.0|M/S|100.0|2011-05-09T06:38:00|2011-12-21T12:45:00
...
```

CC BY

Combining **station** and **dataselect**

We use **grep** to select only "EH" and "HH" channels, and **awk** to print only the trace ids:

```
$ grep -e "|EH" -e "|HH" stations.txt | awk -v FS="|" '{print $1, $2, $3, $4}'
PF BON 00 HHE
PF BON 00 HHE
PF BON 00 HHE
PF BON 00 HHE
PF BON 00 HHN
PF BON 00 HHN
PF BON 00 HHN
PF BON 00 HHN
PF BON 00 HHZ
PF BON 00 HHZ
PF BON 00 HHZ
PF BON 00 HHZ
PF BOR 00 EH1
PF BOR 00 EH2
PF BOR 00 EHE
PF BOR 00 EHE
PF BOR 00 EHE
PF BOR 00 EHE
PF BOR 00 EHE
PF BOR 00 EHN
...
```

CC BY

Combining **station** and **dataselect**

Second part of the script: we loop over trace ids and use **dataselect** to retrieve miniSEED data.

```
# Loop over EH? and HH? channels
grep -e "|EH" -e "|HH" stations.txt | awk -v FS="|" '{print $1, $2, $3, $4}' |\
while read net sta loc chan
do
  echo $net $sta $loc $chan

  outfile=$net.$sta.$loc.$chan.$starttime.$endtime.mseed

  # Query data. We use "nodata=404" parameter to generate an error if no data is available
  URL="$WSURL/dataselect/1/query?"
  URL+="network=$net&station=$sta&location=$loc&channel=$chan"
  URL+="&starttime=$starttime&endtime=$endtime&nodata=404"
  # If no data is available, wget will generate an error and outfile is removed (it is empty)
  wget -nv $URL -O $outfile || rm -f $outfile
done
```

CC BY

Combining **station** and **dataselect**

Output files should look like this:

```
$ ls *mseed
PF.BON.00.HHE.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
PF.ENO.90.EHZ.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
PF.BON.00.HHN.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
PF.FJS.00.HH1.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
PF.BON.00.HHZ.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
PF.FJS.00.HH2.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
PF.BOR.00.EH1.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
PF.FJS.00.HHE.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
PF.BOR.00.EH2.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
PF.FJS.00.HHN.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
PF.BOR.00.EHE.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
PF.FJS.00.HHZ.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
PF.BOR.00.EHN.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
PF.FLR.00.HHE.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
PF.BOR.00.EHZ.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
PF.FLR.00.HHN.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
PF.BOR.90.EHE.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
PF.FLR.00.HHZ.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
PF.BOR.90.EHN.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
PF.FOR.00.HHE.2019-12-01T00:00:00.2019-12-01T01:00:00.mseed
...
```

Combining **station** and **dataselect**

Same script in Python, using ObsPy (1/2):

```
#!/usr/bin/env python
from obspy.clients.fdsn import Client
from obspy import UTCDateTime

# Webservices URL
WSURL = 'https://ws.ipgp.fr'
# Init the client
cl = Client(WSURL)
# ObsPy has shortcuts for major data centers:
# cl = Client('IPGP')

# Center coordinates to compute max radius
lat0 = -21.244384
lon0 = 55.714093
# Max radius (in degrees, 1 degree = 111 km)
maxradius = 0.027  # 3 km

# starttime and endtime
starttime = UTCDateTime('2019-12-01T00:00:00')
endtime = UTCDateTime('2019-12-01T01:00:00')
```

CC BY

Combining **station** and **dataselect**

Same script in Python, using ObsPy (2/2):

```
# Query station list
inv = cl.get_stations(latitude=lat0, longitude=lon0, minradius=0, maxradius=maxradius,level='channel')
channels = inv.get_contents()['channels']

# select 'EH?' and 'HH?' channels
channels_select = [c for c in channels if c.split('.')[3][:2] in ('EH', 'HH')]

# Loop over EH? and HH? channels
for chan_id in channels_select:
    print(chan_id)
    net, sta, loc, chan = chan_id.split('.')

    outfile = '{}.{}.{}.{}.{}.{}.mseed'.format(net, sta, loc, chan, starttime, endtime)

    # Query data
    try:
        st = cl.get_waveforms(
            network=net, station=sta, location=loc, channel=chan,starttime=starttime, endtime=endtime)
        st.write(outfile, format='MSEED')
    except Exception:
        pass
```

CC BY

fdsnws-event

https://ws.ipgp.fr	/fdsnws	/event	/1	/query?

Parameter	Alias	Support	Default	Allowed Values	Type	Unit
starttime	**start**	**required**	**[Any]**	**Any valid time**	**time**	**UTC**
Limit to events on or after the specified start time.						
endtime	**end**	**required**	**[Any]**	**Any valid time**	**time**	**UTC**
Limit to events on or before the specified end time.						
minlatitude	**minlat**	**required**	**-90.0**	**-90.0 to 90.0**	**float**	**degrees**
Limit to events with a latitude larger than or equal to the specified minimum.						
maxlatitude	**maxlat**	**required**	**90.0**	**-90.0 to 90.0**	**float**	**degrees**
Limit to events with a latitude smaller than or equal to the specified maximum.						
minlongitude	**minlon**	**required**	**-180.0**	**-180.0 to 180.0**	**float**	**degrees**
Limit to events with a longitude larger than or equal to the specified minimum.						
maxlongitude	**maxlon**	**required**	**108.0**	**-180.0 to 180.0**	**float**	**degrees**
Limit to events with a longitude smaller than or equal to the specified maximum.						
latitude	**lat**	**optional**	**0.0**	**-90.0 to 90.0**	**float**	**degrees**
Specify the latitude to be used for a radius search.						
longitude	**lon**	**optional**	**0.0**	**-180.0 to 180.0**	**float**	**degrees**
Specify the longitude to be used for a radius search.						
minradius		**optional**	**0.0**	**0.0 to 180.0**	**float**	**degrees**
Limit to events within the specified minimum number of degrees from the geographic point defined by the latitude and longitude parameters.						
maxradius		**optional**	**180.0**	**0.0 to 180.0**	**float**	**degrees**
Limit to events within the specified maximum number of degrees from the geographic point defined by the latitude and longitude parameters.						
mindepth		**required**	**[Any]**	**Any valid depth**	**float**	**kilometers**
Limit to events with depth more than the specified minimum.						
maxdepth		**required**	**[Any]**	**Any valid depth**	**float**	**kilometers**
Limit to events with depth less than the specified maximum.						

CC BY

fdsnws-event

https://ws.ipgp.fr	/fdsnws	/event	/1	/query?

Parameter	Alias	Support	Default	Allowed Values	Type	Unit
minmagnitude	**minmag**	**required**	**[Any]**	**Any valid magnitude**	**float**	*defined by*
Limit to events with a magnitude larger than the specified minimum.						
maxmagnitude	**maxmag**	**required**	**[Any]**	**Any valid magnitude**	**float**	*mag type*
Limit to events with a magnitude smaller than the specified maximum.						
magnitudetype	**magtype**	**optional**	**[Any]**	**Any valid magnitude type**	**string**	
Specify a magnitude type to use for testing the minimum and maximum limits.						
eventtype		**optional**	**[Any]**	**Any valid event type**	**string**	
Limit to events with a specified eventType. The parameter value can be a single item, a comma-separated list of items. Allowed values are from QuakeML or ***unknown*** if eventType is not given.						
includeallorigins		**optional**	**FALSE**	**TRUE or FALSE**	**boolean**	
Specify if all origins for the event should be included, default is data center dependent but is suggested to be the preferred origin only.						
includeallmagnitudes		**optional**	**FALSE**	**TRUE or FALSE**	**boolean**	
Specify if all magnitudes for the event should be included, default is data center dependent but is suggested to be the preferred magnitude only.						
includearrivals		**optional**	**FALSE**	**TRUE or FALSE**	**boolean**	
Specify if phase arrivals should be included.						
eventid		**optional**	**[Any]**	**Any valid ID**	**string**	
Select a specific event by ID; event identifiers are data center specific.						

CC BY

fdsnws-event

https://ws.ipgp.fr	/fdsnws	/event	/1	/query?

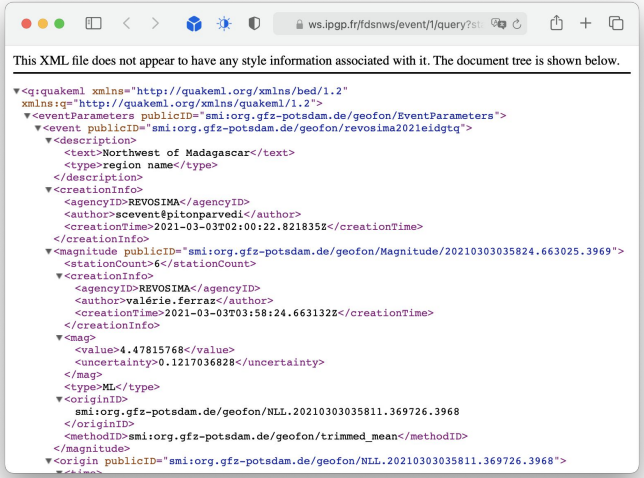
Parameter	Alias	Support	Default	Allowed Values	Type	Unit
limit		**optional**	**[Any]**	**>= 1**	**integer**	
Limit the results to the specified number of events.						
offset		**optional**	**1**	**>= 1**	**integer**	
Return results starting at the event count specified, starting at 1.						
orderby		**required**	***time***	**Valid sort value**	**string**	
Order the result by time or magnitude with the following possibilities: ***time***: order by origin descending time ***time-asc***: order by origin ascending time ***magnitude***: order by descending magnitude ***magnitude-asc***: order by ascending magnitude						
catalog		**optional**	**[Any]**	**Valid catalog identifier**	**string**	
Limit to events from a specified catalog.						
contributor		**optional**	**[Any]**	**Valid contributor identifier**	**string**	
Limit to events contributed by a specified contributor.						
updatedafter		**optional***	**[Any]**	**Any valid time**	**time**	**UTC**
Limit to events updated after the specified time. * While this option is not required it is highly recommended due to usefulness.						
format		**optional**	**text**	**xml, text**	**string**	
Specify format of result, either ***xml*** (default) or ***text*** (defined below). If this parameter is not specified the service must return QuakeML.						
nodata		**optional**	**204**	**204 or 404**	**string**	
Select status code for “no data”, either ‘204’ (default) or ‘404’.						

CC BY

fdsnws-event

Examples

https://ws.ipgp.fr/fdsnws/event/1/query?starttime=2021-01-01T00:00:00&endtime=2021-06-30&minlatitude=-20&maxlatitude=20&minlongitude=30&maxlongitude=50&minmagnitude=1&orderby=magnitude&nodata=404



Default return format is QuakeML

CC BY

fdsnws-**event**

Examples

https://ws.ipgp.fr/fdsnws/event/1/query?starttime=2021-06-27T00:00:00&endtime=2021-06-30&minlatitude=-30&maxlatitude=20&minlongitude=30&maxlongitude=50&minmagnitude=1&orderby=magnitude&format=text&nodata=404

A simple text format can be requested

ws.ipgp.fr/fdsnws/event/1/query?starttime=2021-06-27T00:00:00&endtime=202

```
#EventID|Time|Latitude|Longitude|Depth/km|Author|Catalog|Contributor|ContributorID|MagType|Magnitude|MagAuthor|EventLocationName
revosima2021mpuaex|2021-06-28T13:27:08.33|-12.798700|45.435500|20.67|PhaseNet||REVOSIMA|revosima2021mpuaex|ML|2.985665978|scmag@pitonparvedi|Northwest of Madagascar
revosima2021mmznrg|2021-06-27T01:00:12.700301|-12.829578|45.630809|37.23046875|nicolas.desfete||REVOSIMA|revosima2021mmznrg|ML|2.981918063|nicolas.desfete|Northwest of Madagascar
revosima2021mnslob|2021-06-27T10:31:01.073565|-12.824315|45.609730|40.96875|philippe.catherine||REVOSIMA|revosima2021mnslob|ML|2.642405118|philippe.catherine|Northwest of Madagascar
revosima2021morvxx|2021-06-27T23:17:21.993236|-12.740697|45.520465|47.80078125|philippe.catherine||REVOSIMA|revosima2021morvxx|ML|2.30362119|philippe.catherine|Northwest of Madagascar
revosima2021mqdzru|2021-06-28T18:28:16.539999|-12.851800|45.639000|33.49|PhaseNet||REVOSIMA|revosima2021mqdzru|ML|2.244480111|scmag@pitonparvedi|Northwest of Madagascar
revosima2021morout|2021-06-27T23:09:00.434608|-12.856117|45.613030|39.1640625|philippe.catherine||REVOSIMA|revosima2021morout|ML|1.880722507|philippe.catherine|
revosima2021mpkcmd|2021-06-28T08:28:03.69|-12.775800|45.437800|22.32|PhaseNet||REVOSIMA|revosima2021mpkcmd|ML|1.707847862|scmag@pitonparvedi|Northwest of Madagascar
revosima2021mnaxav|2021-06-27T01:41:13.393518|-12.835952|45.624338|42.38671875|nicolas.desfete||REVOSIMA|revosima2021mnaxav|ML|1.657253923|nicolas.desfete|
revosima2021mozkax|2021-06-28T03:04:57.333714|-12.783228|45.426543|21.84765625|philippe.catherine||REVOSIMA|revosima2021mozkax|ML|1.589937442|philippe.catherine|
revosima2021mqeppb|2021-06-28T18:46:43.24|-12.789700|45.342300|40.15|PhaseNet||REVOSIMA|revosima2021mqeppb|ML|1.557691704|scmag@pitonparvedi|Northwest of Madagascar
revosima2021mopmrd|2021-06-27T22:06:27.314865|-12.783292|45.361725|36.37109375|philippe.catherine||REVOSIMA|revosima2021mopmrd|ML|1.121810874|philippe.catherine|
revosima2021mqlxjx|2021-06-28T22:26:58.719999|-12.832000|45.416800|31.13|PhaseNet||REVOSIMA|revosima2021mqlxjx|ML|1.012394326|scmag@pitonparvedi|Northwest of Madagascar
```

CC BY

fdsnws-event

Examples

https://ws.ipgp.fr/fdsnws/event/1/query?eventid=ovsg2021lempgx

ws.ipgp.fr/fdsnws/event/1/query?eve

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
▼<q:quakeml xmlns="http://quakeml.org/xmlns/bed/1.2"
 xmlns:q="http://quakeml.org/xmlns/quakeml/1.2">
  ▼<eventParameters publicID="smi:org.gfz-potsdam.de/geofon/EventParameters">
    ▼<event publicID="smi:org.gfz-potsdam.de/geofon/ovsg2021lempgx">
      <typeCertainty>known</typeCertainty>
      ▼<creationInfo>
        <agencyID>OVSG</agencyID>
        <author>scevent@aulart</author>
        <creationTime>2021-06-08T11:12:08.65567Z</creationTime>
      </creationInfo>
      ▼<magnitude publicID="smi:org.gfz-potsdam.de/geofon/Magnitude#20210727171655.527534.8921">
        <stationCount>6</stationCount>
        ▼<creationInfo>
          <agencyID>OVSG</agencyID>
          <author>pierre</author>
          <creationTime>2021-07-27T17:17:41.466717Z</creationTime>
        </creationInfo>
        ▼<mag>
          <value>4.619070076</value>
          <uncertainty>0.1755876006</uncertainty>
        </mag>
        <type>ML</type>
        ▼<originID>
          smi:org.gfz-potsdam.de/geofon/NLL.20210727171638.438657.8915
        </originID>
        <methodID>smi:org.gfz-potsdam.de/geofon/trimmed_mean</methodID>
        <evaluationStatus>confirmed</evaluationStatus>
      </magnitude>
      ▼<origin publicID="smi:org.gfz-potsdam.de/geofon/NLL.20210727171638.438657.8915">
        ▼<time>
          <value>2021-06-08T05:57:15.732406Z</value>
        </time>
        ▼<longitude>
          <value>-62.6341444</value>
          <uncertainty>51.58838262</uncertainty>
        </longitude>
        ▼<latitude>
          <value>19.78901147</value>
          <uncertainty>20.15058526</uncertainty>
        </latitude>
```


CC BY

fdsnws-event

In terminal, using wget:

Note that special characters are escaped by "\"

```
$ wget
https://ws.ipgp.fr/fdsnws/event/1/query\?starttime\=2021-05-14T00:00:00\&endtime\=2021-
05-14T23:59:59\&minlatitude\=12\&maxlatitude\=20\&minlongitude\=-70\&maxlongitude\=-50\
&minmagnitude\=2\&orderby\=magnitude\&nodata=404 -O events.xml
```

...or curl:

```
$ curl
https://ws.ipgp.fr/fdsnws/event/1/query\?starttime\=2021-05-14T00:00:00\&endtime\=2021
-05-14T23:59:59\&minlatitude\=12\&maxlatitude\=20\&minlongitude\=-70\&maxlongitude\=-5
0\&minmagnitude\=2\&orderby\=magnitude\&nodata=404 -O events.xml
```

CC BY

fdsnws-events

Using ObsPy
www.obspy.org

ObsPy
A Python Framework for Seismology

```
$ python
>>> from obspy.clients.fdsn import Client
>>> from obspy import UTCDateTime
>>> cl = Client('IRIS')
>>> cat = cl.get_events(starttime=UTCDateTime('2018-05-14T00:00:00'),
        endtime=UTCDateTime('2018-05-14T23:59:59'), minlatitude=-14, maxlatitude=-12,
        minlongitude=45, maxlongitude=46, minmagnitude=4)
>>> cat.write('events.xml', format='QUAKEML')
```

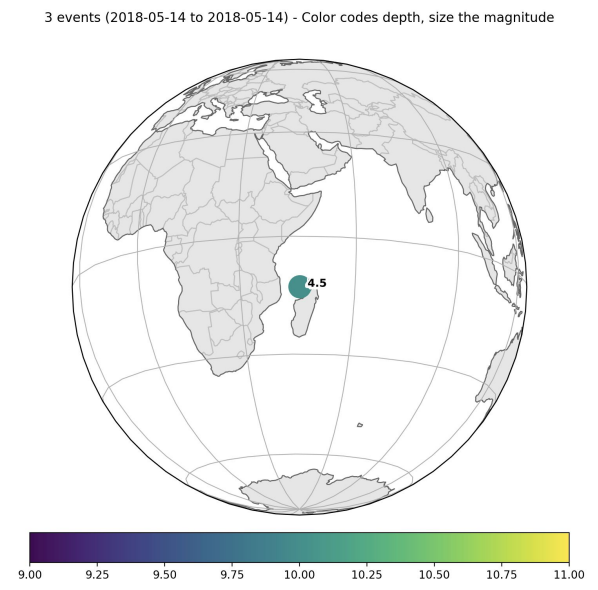
Documentation: https://docs.obspy.org/packages/obspy.clients.fdsn.html

CC BY

fdsnws-events

ObsPy bonus: plot catalog

```
>>> cat.plot(projection='ortho')
```



CC BY

fdsnws-events

Using FetchEvent
https://seiscode.iris.washington.edu/projects/ws-fetch-scripts

```
$ FetchEvent -s 2018-05-14T00:00:00 -e 2018-05-14T23:59:59 --lat -14:-12 --lon 45:46
--mag 4 --orderbymag -X events.xml
```

```
$ FetchEvent --evid 10803428
```

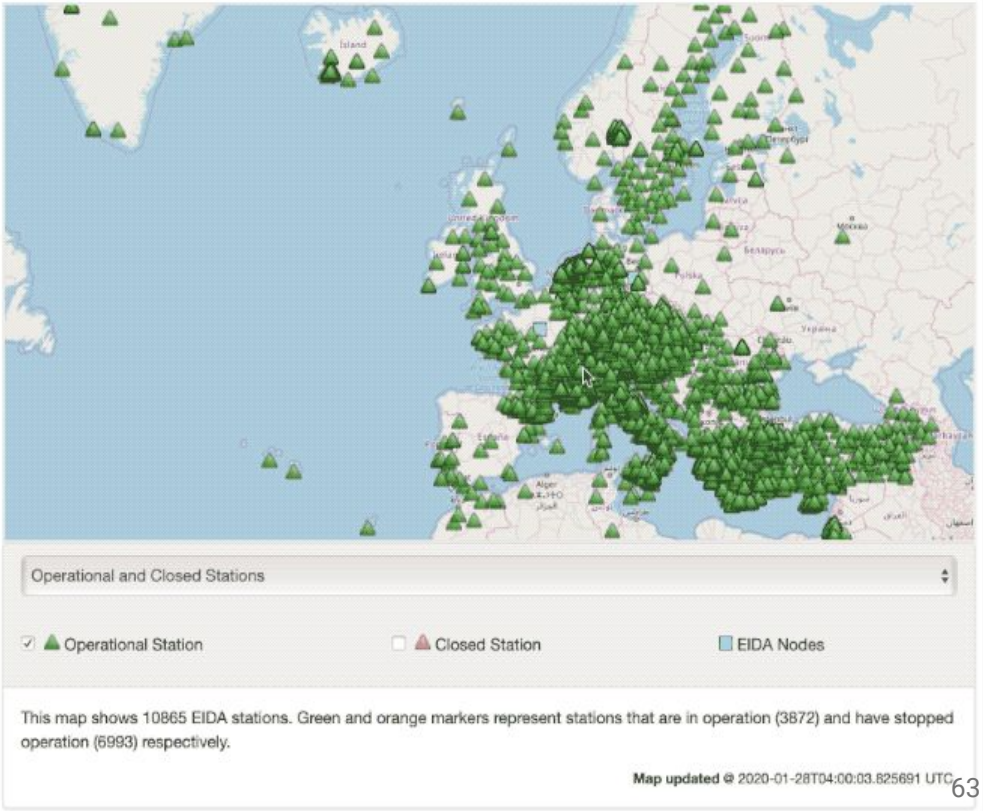
Documentation: https://seiscode.iris.washington.edu/projects/ws-fetch-scripts/wiki

CC BY

Federated/routing web services

Interesting seismic data can cross nation and datacenter borders.

How to know which data center has the data I'm interested in?



CC BY

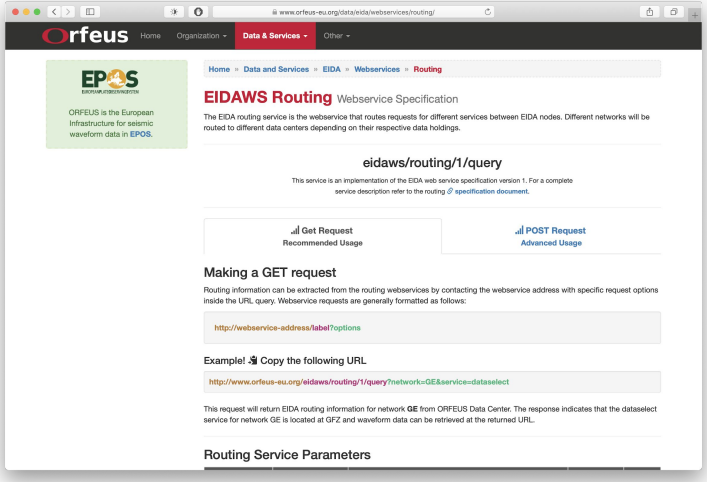
Federated/routing web services

How to know which data center has the data I'm interested in?

The answer is: "Use a federated/routing webservice!"

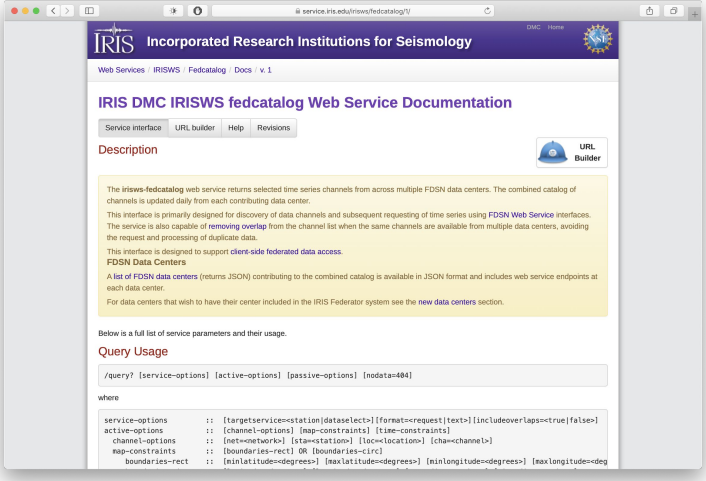
Currently, two possibilities (not yet a FDSN standard…):

EIDAWS Routing



https://www.orfeus-eu.org/data/eida/webservices/routing/

IRISWS fedcatalog



https://service.iris.edu/irisws/fedcatalog/1/

CC BY

eidaws-routing

http://www.orfeus-eu.org	/eidaws	/routing	/1	/query?

Routing Service Parameters

Parameter	Examples	Description	Default	Type
Temporal Constraints				
starttime (start)	`2016-01-01T00:00:00`	Starttime of request		ISO-8601
endtime (end)	`2016-01-02T00:00:00`	Endtime of request		ISO-8601
SEED Identifiers				
network (net)	`NL`	SEED Network code		String
station (sta)	`HGN`	SEED Station code		String
location (loc)	`00`, `--`	SEED Location code		String
channel (cha)	`BHZ`, `HH*`	SEED Channel code		String
Request Options				
minlatitude (minlat)	`20`	Limit to stations with a latitude larger than or equal to the specified minimum.		Float
maxlatitude (maxlat)	`65`	Limit to stations with a latitude smaller than or equal to the specified maximum.		Float
minlongitude (minlon)	`-15`	Limit to stations with a longitude larger than or equal to the specified minimum.		Float
maxlongitude (maxlon)	`70`	Limit to stations with a longitude smaller than or equal to the specified maximum.		Float
service	`dataselect`	Service (`dataselect`, `station`, `wfcatalog`)	`dataselect`	String
format	`json`	Response format (`xml`, `json`, `get`, `post`)	`xml`	String
alternative	`true`	Include alternative routes	`false`	Boolean

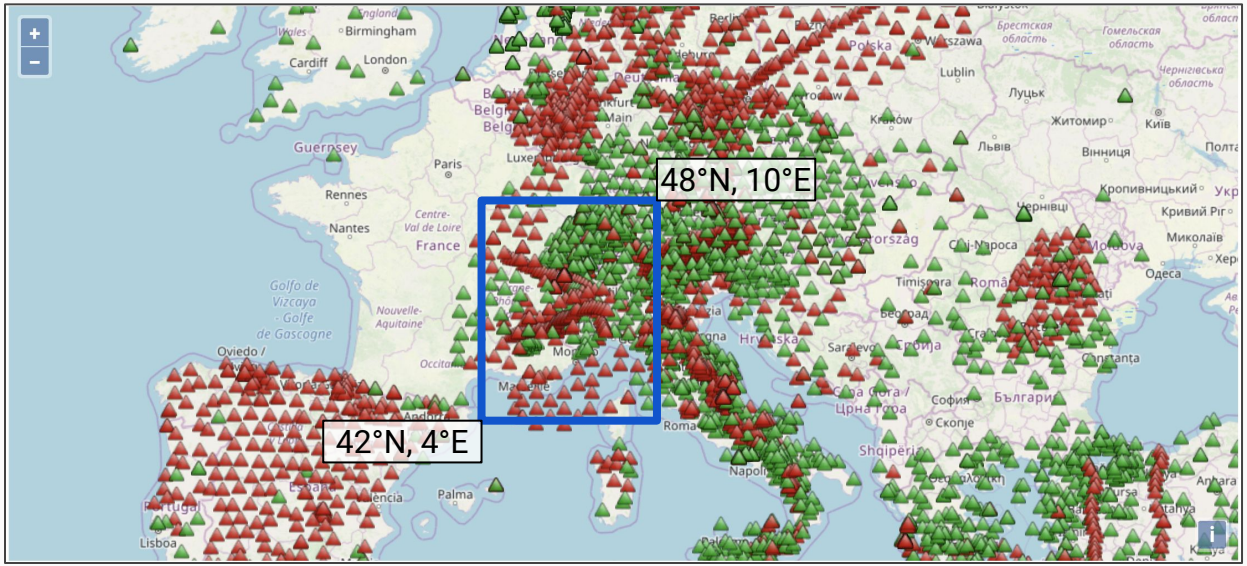
Source: https://www.orfeus-eu.org/data/eida/webservices/routing/

CC BY

eidaws-routing

Example

Let's query all the stations within a geographic bounding box which crosses several European countries.



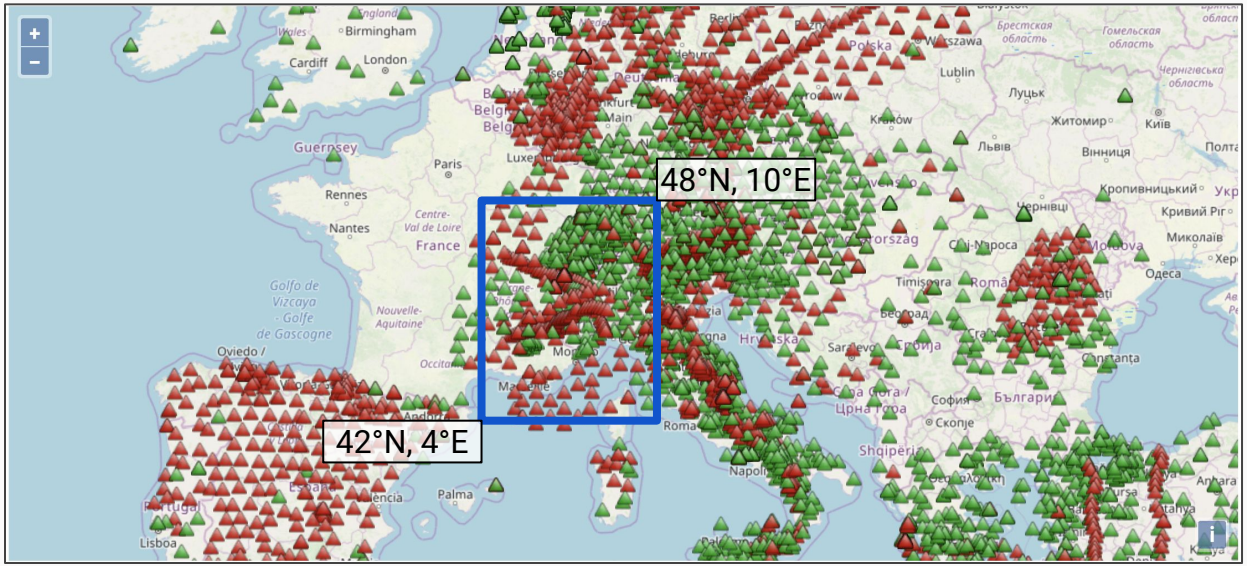
Unfortunately, eidaws-routing does not support circular selection

CC BY

eidaws-routing

Example (fdsnws-station routing)

http://www.orfeus-eu.org/eidaws/routing/1/query?service=station&minlatitude=42&maxlatitude=48&minlongitude=4&maxlongitude=10



CC BY

eidaws-routing

Example (fdsnws-station routing)

http://www.orfeus-eu.org/eidaws/routing/1/query?service=station&minlatitude=42&maxlatitude=48&minlongitude=4&maxlongitude=10

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<service>
 ▼<datacenter>
   <url>http://eida.ethz.ch/fdsnws/station/1/query</url>
   <name>station</name>
  ▼<params>
     <loc>*</loc>
     <cha>*</cha>
     <end/>
     <start>2008-07-03 00:00:00</start>
     <net>S</net>
     <priority>2</priority>
     <sta>EBELL</sta>
   </params>
  ▼<params>
     <loc>*</loc>
     <cha>*</cha>
     <end/>
     <start>2008-07-03 00:00:00</start>
     <net>S</net>
     <priority>2</priority>
     <sta>EBUEL</sta>
   </params>
  ▼<params>
     <loc>*</loc>
     <cha>*</cha>
     <end/>
     <start>2008-07-03 00:00:00</start>
     <net>S</net>
     <priority>2</priority>
     <sta>ELAUS</sta>
   </params>
  ▼<params>
     <loc>*</loc>
     <cha>*</cha>
```


The default reply is an XML file.

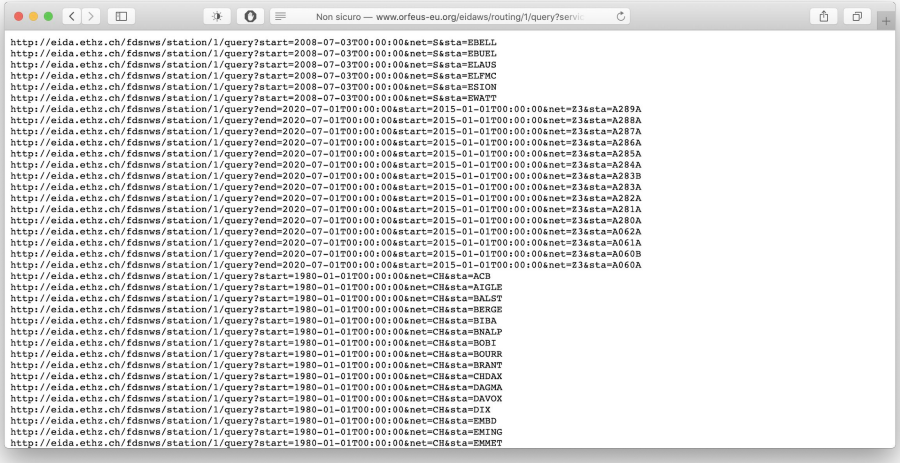
But there is a more interesting output format...

CC BY

eidaws-routing

Example (fdsnws-station routing)

http://www.orfeus-eu.org/eidaws/routing/1/query?service=station&minlatitude=42&maxlatitude=48&minlongitude=4&maxlongitude=10&format=GET



This is a station-by-station list of basic **fdsnws-station** requests to be made to get station metadata, with the correct data center.

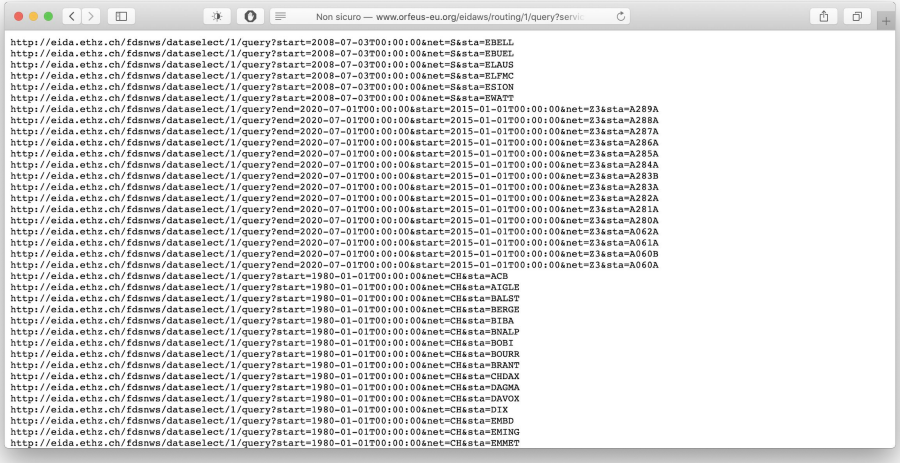
You can use this list into a script and append to these basic requests other parameters, specific to the **fdsnws-station** webservice.

CC BY

eidaws-routing

Example (fdsnws-dataselect routing)

http://www.orfeus-eu.org/eidaws/routing/1/query?service=dataselect&minlatitude=42&maxlatitude=48&minlongitude=4&maxlongitude=10&format=GET



This is a station-by-station list of basic **fdsnws-dataselect** requests to be made to get station data, with the correct data center.

You can use this list into a script and append to these basic requests other parameters, specific to the **fdsnws-dataselect** webservice.

CC BY

irisws-fedcatalog

https://service.iris.edu	/irisws	/fedcatalog	/1	/query?

Detailed Descriptions of each Query Parameter

parameter	examples	discussion	default	type
Service options				
format	text	Specify output format. Accepted values are `request` (the default) and `text`, more details.	request	string
targetservice	station	By default all known service endpoints are returned in the results. Specify `station` or `dataselect` to only return endpoints for one of those services.	***all***	string
includeoverlaps	true	Control whether overlapping channel entries are included in the response (`true` or `false`). Overlapping entries will occur when the same data are available from multiple data centers.	false	boolean
level	station	Specify level of detail (for an **fdsnws-station** service) using `network`, `station`, `channel`, or `response`.	channel	string
datacenter	IRIS	Limit the results to metadata held at an FDSN data center of interest. Use data center codes available at the *datacenters* endpoint. Accepts wildcards, lists, and negation.	***all***	string
Channel constraints				
net[work]	IU	Select one or more network codes. Can be SEED codes or data center defined codes. Accepts wildcards and lists.	***any***	string
sta[tion]	ANMO	Select one or more SEED station codes. Accepts wildcards and lists.	***any***	string
loc[ation]	00	Select one or more SEED location identifier. Use `--` for “Blank” location IDs (ID’s containing 2 spaces). Accepts wildcards and lists.	***any***	string
cha[nnel]	BH1	Select one or more SEED channel codes. Accepts wildcards and lists.	***any***	string

Source: https://service.iris.edu/irisws/fedcatalog/1/

CC BY

irisws-fedcatalog

https://service.iris.edu	/irisws	/fedcatalog	/1	/query?

Detailed Descriptions of each Query Parameter

parameter	examples	discussion	default	type
Time constraints				
start[time]	2001-12-31	Limit to metadata listed on or after the specified start time.	***any***	day/time
end[time]	2012-12-31	Limit to metadata listed on or before the specified end time.	***any***	day/time
startbefore	2001-12-31	Limit to metadata epochs listed before specified time. Applied to channel epochs.	***any***	day/time
startafter	2005-12-31	Limit to metadata epochs listed after specified time. Applied to channel epochs.	***any***	day/time
endbefore	2005-12-31	Limit to metadata epochs ending before specified time. Applied to channel epochs.	***any***	day/time
endafter	2005-12-31	Limit to metadata epochs ending after specified time. Applied to channel epochs.	***any***	day/time

Source: https://service.iris.edu/irisws/fedcatalog/1/

CC BY

irisws-fedcatalog

https://service.iris.edu	/irisws	/fedcatalog	/1	/query?

Detailed Descriptions of each Query Parameter

parameter	examples	discussion	default	type
Rectangular geographic constraints				
minlat[itude]	34.5	Specify the southern boundary.	-90°	degrees
maxlat[itude]	36.0	Specify the northern boundary.	90°	degrees
minlon[gitude]	-108.0	Specify the western boundary.	-180°	degrees
maxlon[gitude]	-105.0	Specify the eastern boundary.	180°	degrees
Radial geographic constraints				
lat[itude]	32.0	Specify the central latitude point.		degrees
lon[gitude]	-106.5	Specify the central longitude point.		degrees
maxradius	10	Specify maximum distance from the geographic point defined by `latitude` and `longitude`.	N/A	degrees
minradius	5	Specify minimum distance from the geographic point defined by `latitude` and `longitude`.	0	degrees

Source: https://service.iris.edu/irisws/fedcatalog/1/

CC BY

irisws-fedcatalog

https://service.iris.edu	/irisws	/fedcatalog	/1	/query?

Detailed Descriptions of each Query Parameter

parameter	examples	discussion	default	type
Waveform options *these are passive options, passed through in the response but otherwise unused*				
quality	B	Select data based on miniSEED data quality indicator. `D`, `R`, `Q`, `M`, `B`. `M` and `B` (default) are treated the same and indicate *best* available. If `M` or `B` are selected, the output data records will be stamped with an `M`.	B	quality
minimumlength	0.0	Limit results to continuous data segments of a minimum length specified in seconds.	0.0	Float
longestonly	false	Limit results to the longest continuous segment per channel.	false	boolean
Other Options *these are passive options, passed through in the response but otherwise unused*				
includerestricted	true	Specify whether results should include information relating to restricted channels.	true	boolean
includeavailability	true	Specify if results should include information about time series data availability at the channel level.	false	boolean
updatedafter	2012-01-01	Limit to metadata updated after specified UTC date; updates are data center specific.	***any***	day/time
matchtimeseries	true	Retrieve only metadata that is likely to match available time series data	false	boolean
nodata	404	Specify which HTTP status code is returned when no data is found.[1]	204	204 or 404

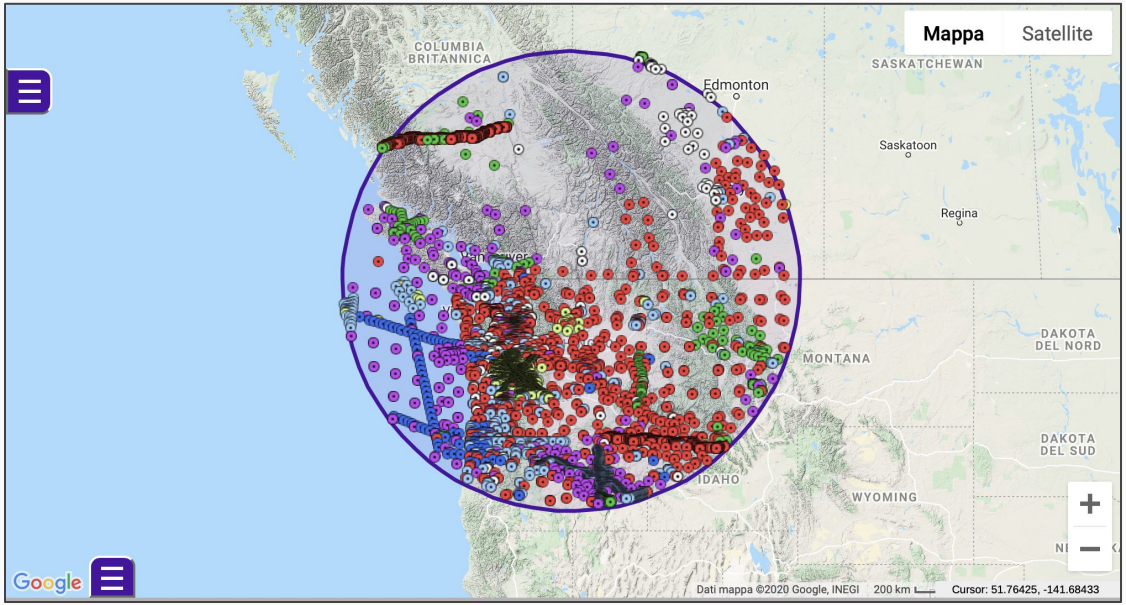
Source: https://service.iris.edu/irisws/fedcatalog/1/

CC BY

irisws-fedcatalog

Example

Let's query all the stations within a 6 degrees radius from a point in North America, across US-Canada border.



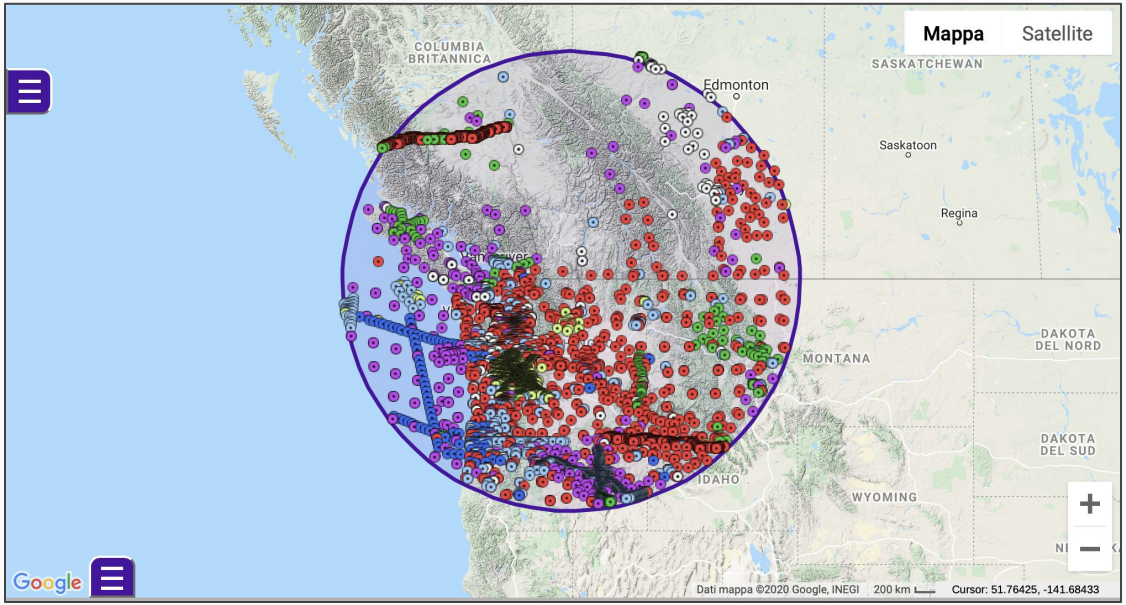
CC BY

irisws-fedcatalog

The "net" parameter is mandatory!

Example

https://service.iris.edu/irisws/fedcatalog/1/query?net=*&lat=49.000&lon=-120.000&minradius=0&maxradius=6.000



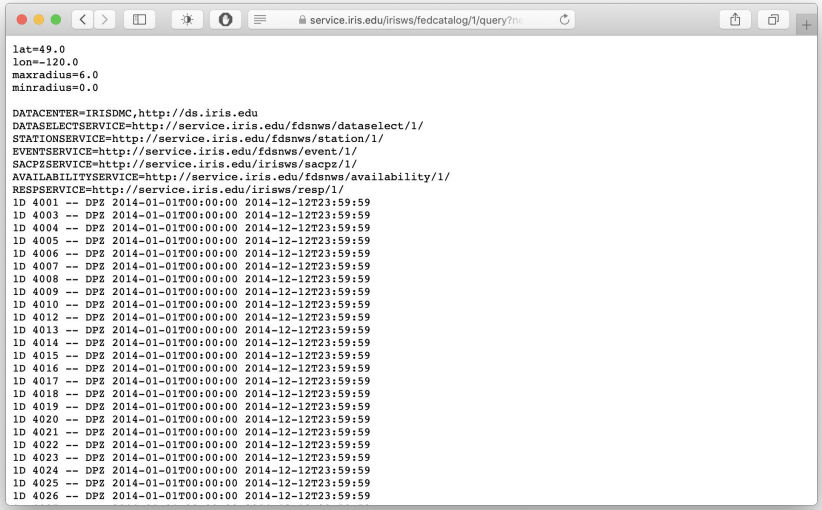
CC BY

irisws-fedcatalog

The "net" parameter is mandatory!

Example

https://service.iris.edu/irisws/fedcatalog/1/query?net=*&lat=49.000&lon=-120.000&minradius=0&maxradius=6.000



irisws-fedcatalog output is a text file with:

- The selection parameters provided by the user
- A list of datacenters and services
- A list of available channels

This file can be read and used by a script, but this is beyond the scope of this class...

CC BY

Using federated/routing WS with ObsPy

ObsPy provides a simple and consistent interface to the two federated/routing webservices.

Example: use **eidaws-routing** to retrieve station metadata

```
from obspy.clients.fdsn import RoutingClient
from obspy import UTCDateTime

cl = RoutingClient('eida-routing')

# Note: ObsPy requires to specify starttime and endtime
t0 = UTCDateTime('20190101')
t1 = UTCDateTime('20200101')
inv = cl.get_stations(
    starttime=t0, endtime=t1,
    minlatitude=42, maxlatitude=48, minlongitude=4, maxlongitude=10)
inv.write('inventory.xml', format='StationXML')

# ObsPy even allows circular selection for eidaws-routing
inv = cl.get_stations(
     starttime=t0, endtime=t1, latitude=45, longitude=7, maxradius=6)
inv.write('inventory.circular.xml', format='StationXML')
```

CC BY

Using federated/routing WS with ObsPy

ObsPy provides a simple and consistent interface to the two federated/routing webservices.

Example: use **eidaws-routing** to retrieve station metadata and data

```
from obspy.clients.fdsn import RoutingClient
from obspy import UTCDateTime

cl = RoutingClient('eida-routing')
t0 = UTCDateTime('2019-01-01T00:00:00')
t1 = UTCDateTime('2019-01-01T00:10:00')
inv = cl.get_stations(starttime=t0, endtime=t1, minlatitude=42, maxlatitude=48,
                      minlongitude=4, maxlongitude=10, level='channel')

for channel in inv.get_contents()['channels']:
    print(channel)
    net, sta, loc, chan = channel.split('.')
    try:
        st = cl.get_waveforms(network=net, station=sta, location=loc, channel=chan,
                              starttime=t0, endtime=t1)
    except Exception:
        continue
    outfile = channel + '.mseed'
    st.write(outfile, format='MSEED')
```

CC BY

Using federated/routing WS with ObsPy

ObsPy provides a simple and consistent interface to the two federated/routing webservices.

Example: use **fdsnws-fedcatalog** to retrieve station metadata and data

```
from obspy.clients.fdsn import RoutingClient
from obspy import UTCDateTime

cl = RoutingClient('iris-federator')
t0 = UTCDateTime('2019-01-01T00:00:00')
t1 = UTCDateTime('2019-01-01T00:10:00')
inv = cl.get_stations(starttime=t0, endtime=t1, latitude=49, longitude=-120,
                      maxradius=6, level='channel', includerestricted=False)

for channel in inv.get_contents()['channels']:
    print(channel)
    net, sta, loc, chan = channel.split('.')
    try:
        st = cl.get_waveforms(network=net, station=sta, location=loc, channel=chan,
                              starttime=t0, endtime=t1)
    except Exception:
        continue
    outfile = channel + '.mseed'
    st.write(outfile, format='MSEED')
```

CC BY

Using federated/routing WS with ObsPy

Documentation about ObsPy FDSN clients:

https://docs.obspy.org/packages/obspy.clients.fdsn.html

ObsPy
A Python Framework for Seismology

CC BY

fdsnws-availability

https://ws.resif.fr	/fdsnws	/availability	/1	/extent?

Parameter	Alias	Support	Default	Allowed Values	Type	Unit
starttime	**start**	**required**	**[Any]**	**Any valid time]**	**time**	**UTC**
	Limit results to time series samples on or after the specified start time.					
endtime	**end**	**required**	**[Any]**	**Any valid time**	**time**	**UTC**
	Limit results to time series samples on or before the specified end time.					
network	**net**	**required**	**[Any]**	**Valid network code or wildcard**	**string**	
	Select one or more network codes. Can be FDSN network codes or data center defined codes. Multiple codes are comma-separated.					
station	**sta**	**required**	**[Any]**	**Valid network code or wildcard**	**string**	
	Select one or more FDSN station codes. Multiple codes are comma-separated.					
location	**loc**	**required**	**[Any]**	**Valid station code or wildcard**	**string**	
	Select one or more FDSN location identifiers. Multiple identifiers are comma-separated. As a special case '--' (two dashes) will be translated to a string of two space characters to match blank location IDs.					
channel	**cha**	**required**	**[Any]**	**Valid channel code or wildcard**	**string**	
	Select one or more FDSN channel codes. Multiple codes are comma-separated.					
quality		**optional**	**[Any]**	**Valid quality code or wildcard**	**string**	
	Select a specific FDSN quality indicator, handling is data center dependent.					
merge		**optional**	**[None]**	**Valid merge flag**	**string**	
	If set to one or more of the following values, time spans are merged as described. Multiple values may be specified as a comma-delimited list, e.g. *merge=samplerate,quality*. ***samplerate***: time spans from data with differing sample rates will be grouped together. If specified this field will be omitted from the result. ***quality***: time spans from data with differing quality codes will be grouped together. If specified this field will be omitted from the result. ***overlap***: time spans from data that overlap will be merged together. This option does not apply to the *extent* method.					

The **/extent** method produces a list of available time extents (earliest to latest) for selected channels (network, station, location, channel and quality) and time ranges.

Source: https://www.fdsn.org/webservices/

CC BY

fdsnws-availability

https://ws.resif.fr	/fdsnws	/availability	/1	/extent?

Parameter	Alias	Support	Default	Allowed Values	Type	Unit
orderby		**optional**	**nslc_time_quality_samplerate**		**string**	
	Sort results by one of the following values in the order specified: ***nslc_time_quality_samplerate***: network, station, location, channel, time-range, quality, sample-rate (**default**) ***latestupdate***: update-date (past to present), network, station, location, channel, time-range, quality, sample-rate ***latestupdate_desc***: update-date (present to past), network, station, location, channel, time-range, quality, sample-rate ***timespancount***: number of timespans (small to large), network, station, location, channel, time-range, quality, sample-rate ***timespancount_desc***: number of timespans (large to small), network, station, location, channel, time-range, quality, sample-rate					
limit		**optional**	**[Unlimited]**	**Valid merge flag**	**integer**	
	Limit results to the specified number of timespans.					
includerestricted		**optional**	**FALSE**	**TRUE or FALSE**	**boolean**	
	If true, all data are reported. If false, only data that can be openly accessed.					
format		**optional**	**text**	**text, geocsv, request**	**string**	
	Specify format of result, either ***text*** (the default) or ***geocsv*** or ***json*** or ***request***.					
nodata		**optional**	**204**	**204 or 404**	**string**	
	Select HTTP status code for "no data", either ***204*** (default) or ***404***.					

The ***/extent*** method produces a list of available time extents (earliest to latest) for selected channels (network, station, location, channel and quality) and time ranges.

Source: https://www.fdsn.org/webservices/

CC BY

fdsnws-availability

https://ws.resif.fr	/fdsnws	/availability	/1	/query?

Table 2. Additional *query* method parameters:

Parameter	Alias	Support	Default	Allowed Values	Type	Unit
mergegaps		**optional**	**0.0**	**>= 0.0**	**float**	
	If set, merge time spans that are separated by the specified tolerance in seconds.					
show		**optional**	**[None]**	**latestupdate**	**string**	
	If set to ***latestupdate***, the latest times at which data contributing to the returned time spans were loaded into the repository are included in the result. This option applies to all formats except ***request***.					

The **/query** method produces a list of contiguous time spans for selected channels (network, station, location, channel and quality) and time ranges.

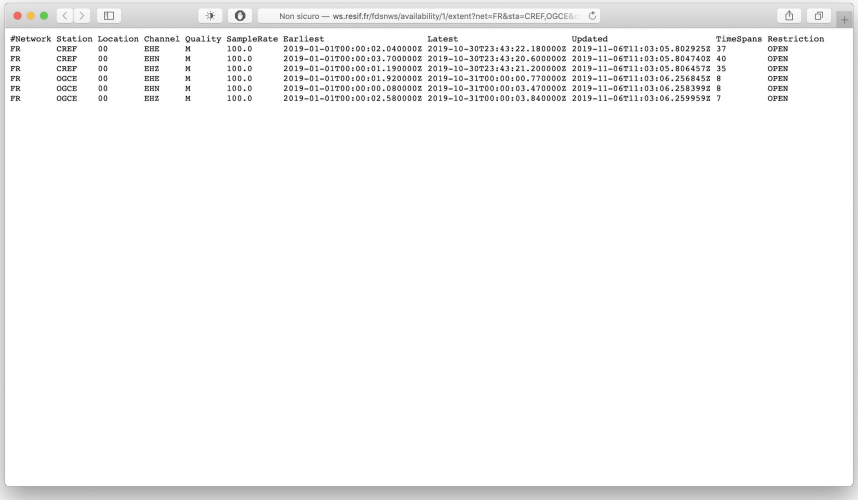
Source: https://www.fdsn.org/webservices/

CC BY

fdsnws-**availability**

Examples

https://ws.resif.fr/fdsnws/availability/1/extent?net=FR&sta=CREF,OGCE&cha=EH?&start=2018-01-01



```
#Network Station Location Channel Quality SampleRate Earliest                   Latest                   Updated                  TimeSpans Restriction
FR      CREF    00       EHE     M       100.0      2019-01-01T00:00:02.040000Z 2019-10-30T23:43:22.180000Z 2019-11-06T11:03:05.802925Z 37        OPEN
FR      CREF    00       EHN     M       100.0      2019-01-01T00:00:03.700000Z 2019-10-30T23:43:20.600000Z 2019-11-06T11:03:05.804740Z 40        OPEN
FR      CREF    00       EHZ     M       100.0      2019-01-01T00:00:01.190000Z 2019-10-30T23:43:21.200000Z 2019-11-06T11:03:05.806457Z 35        OPEN
FR      OGCE    00       EHE     M       100.0      2019-01-01T00:00:01.920000Z 2019-10-31T00:00:00.770000Z 2019-11-06T11:03:06.256845Z 8         OPEN
FR      OGCE    00       EHN     M       100.0      2019-01-01T00:00:00.080000Z 2019-10-31T00:00:03.470000Z 2019-11-06T11:03:06.258399Z 8         OPEN
FR      OGCE    00       EHZ     M       100.0      2019-01-01T00:00:02.580000Z 2019-10-31T00:00:03.840000Z 2019-11-06T11:03:06.259959Z 7         OPEN
```

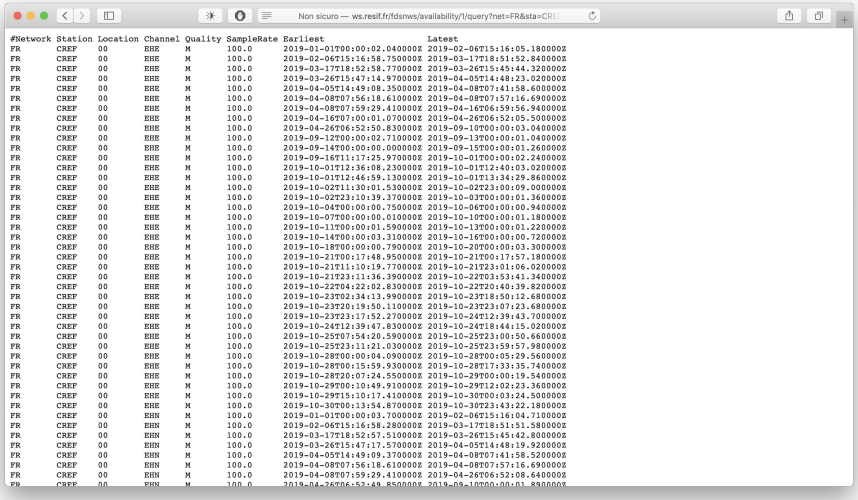
The **/extent** method produces a list of available time extents (earliest to latest) for selected channels (network, station, location, channel and quality) and time ranges.

CC BY

fdsnws-**availability**

Examples

https://ws.resif.fr/fdsnws/availability/1/query?net=FR&sta=CREF,OGCE&cha=EH?&start=2018-01-01

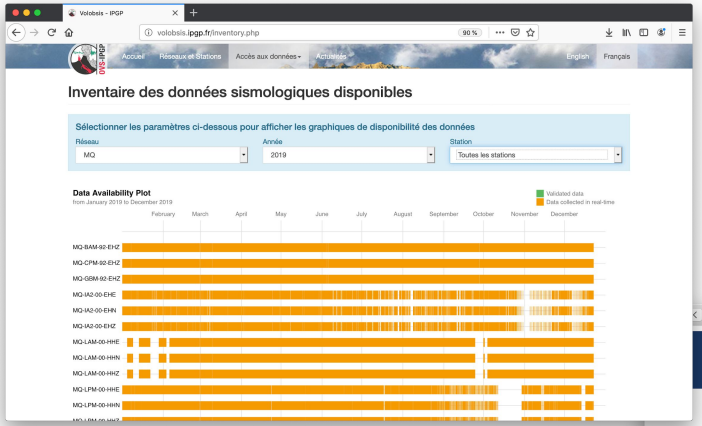



```
#Network Station Location Channel Quality SampleRate Earliest                  Latest
FR      CREF    00       EHE     M       100.0      2019-01-01T00:00:02.040000Z 2019-02-06T15:16:05.180000Z
FR      CREF    00       EHE     M       100.0      2019-02-06T15:16:58.750000Z 2019-03-17T18:51:52.840000Z
FR      CREF    00       EHE     M       100.0      2019-03-17T18:52:58.770000Z 2019-03-26T15:45:44.320000Z
FR      CREF    00       EHE     M       100.0      2019-03-26T15:47:14.970000Z 2019-04-05T14:48:23.020000Z
FR      CREF    00       EHE     M       100.0      2019-04-05T14:49:08.350000Z 2019-04-08T07:41:58.600000Z
FR      CREF    00       EHE     M       100.0      2019-04-08T07:56:18.610000Z 2019-04-08T07:57:16.690000Z
FR      CREF    00       EHE     M       100.0      2019-04-08T07:59:29.410000Z 2019-04-16T06:59:56.940000Z
FR      CREF    00       EHE     M       100.0      2019-04-16T07:00:01.070000Z 2019-04-26T06:52:05.500000Z
FR      CREF    00       EHE     M       100.0      2019-04-26T06:52:50.830000Z 2019-09-10T00:00:03.040000Z
FR      CREF    00       EHE     M       100.0      2019-09-12T00:00:02.710000Z 2019-09-13T00:00:01.040000Z
FR      CREF    00       EHE     M       100.0      2019-09-14T00:00:00.000000Z 2019-09-15T00:00:01.260000Z
FR      CREF    00       EHE     M       100.0      2019-09-16T11:17:25.970000Z 2019-10-01T00:00:02.240000Z
FR      CREF    00       EHE     M       100.0      2019-10-01T12:36:08.230000Z 2019-10-01T12:40:03.020000Z
FR      CREF    00       EHE     M       100.0      2019-10-01T12:46:59.130000Z 2019-10-01T13:34:29.860000Z
FR      CREF    00       EHE     M       100.0      2019-10-02T11:30:01.530000Z 2019-10-02T23:00:09.000000Z
FR      CREF    00       EHE     M       100.0      2019-10-02T23:10:39.370000Z 2019-10-03T00:00:01.360000Z
FR      CREF    00       EHE     M       100.0      2019-10-04T00:00:00.750000Z 2019-10-06T00:00:00.940000Z
FR      CREF    00       EHE     M       100.0      2019-10-07T00:00:00.010000Z 2019-10-10T00:00:01.180000Z
FR      CREF    00       EHE     M       100.0      2019-10-11T00:00:01.590000Z 2019-10-13T00:00:01.220000Z
FR      CREF    00       EHE     M       100.0      2019-10-14T00:00:03.310000Z 2019-10-16T00:00:00.720000Z
FR      CREF    00       EHE     M       100.0      2019-10-18T00:00:00.790000Z 2019-10-20T00:00:03.300000Z
FR      CREF    00       EHE     M       100.0      2019-10-21T00:17:48.950000Z 2019-10-21T00:17:57.180000Z
FR      CREF    00       EHE     M       100.0      2019-10-21T11:10:19.770000Z 2019-10-21T23:01:06.020000Z
FR      CREF    00       EHE     M       100.0      2019-10-21T23:11:36.390000Z 2019-10-22T03:53:41.340000Z
FR      CREF    00       EHE     M       100.0      2019-10-22T04:22:02.830000Z 2019-10-22T20:40:39.820000Z
FR      CREF    00       EHE     M       100.0      2019-10-23T02:34:13.990000Z 2019-10-23T18:50:12.680000Z
FR      CREF    00       EHE     M       100.0      2019-10-23T20:19:50.110000Z 2019-10-23T23:07:23.680000Z
FR      CREF    00       EHE     M       100.0      2019-10-23T23:17:52.270000Z 2019-10-24T12:39:43.700000Z
FR      CREF    00       EHE     M       100.0      2019-10-24T12:39:47.830000Z 2019-10-24T18:44:15.020000Z
FR      CREF    00       EHE     M       100.0      2019-10-25T07:54:20.590000Z 2019-10-25T23:00:50.660000Z
FR      CREF    00       EHE     M       100.0      2019-10-25T23:11:21.030000Z 2019-10-25T23:59:57.980000Z
FR      CREF    00       EHE     M       100.0      2019-10-28T00:00:04.090000Z 2019-10-28T00:05:29.560000Z
FR      CREF    00       EHE     M       100.0      2019-10-28T00:15:59.930000Z 2019-10-28T17:33:35.740000Z
FR      CREF    00       EHE     M       100.0      2019-10-28T20:07:24.550000Z 2019-10-29T00:00:19.540000Z
FR      CREF    00       EHE     M       100.0      2019-10-29T00:10:49.910000Z 2019-10-29T12:02:23.360000Z
FR      CREF    00       EHE     M       100.0      2019-10-29T15:10:17.410000Z 2019-10-30T00:03:24.500000Z
FR      CREF    00       EHE     M       100.0      2019-10-30T00:13:54.870000Z 2019-10-30T23:43:22.180000Z
FR      CREF    00       EHN     M       100.0      2019-01-01T00:00:03.700000Z 2019-02-06T15:16:04.710000Z
FR      CREF    00       EHN     M       100.0      2019-02-06T15:16:58.280000Z 2019-03-17T18:51:51.580000Z
FR      CREF    00       EHN     M       100.0      2019-03-17T18:52:57.510000Z 2019-03-26T15:45:42.800000Z
FR      CREF    00       EHN     M       100.0      2019-03-26T15:47:17.570000Z 2019-04-05T14:48:19.920000Z
FR      CREF    00       EHN     M       100.0      2019-04-05T14:49:09.370000Z 2019-04-08T07:41:58.520000Z
FR      CREF    00       EHN     M       100.0      2019-04-08T07:56:18.610000Z 2019-04-08T07:57:16.690000Z
FR      CREF    00       EHN     M       100.0      2019-04-08T07:59:29.410000Z 2019-04-26T06:52:08.640000Z
FR      CREF    00       EHN     M       100.0      2019-04-26T06:52:49.850000Z 2019-09-10T00:00:01.890000Z
```

The **/query** method produces a list of contiguous time spans for selected channels (network, station, location, channel and quality) and time ranges.

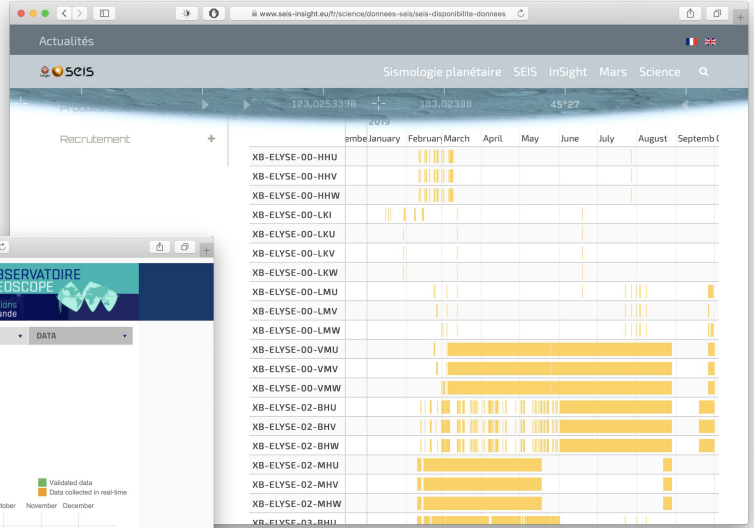
CC BY

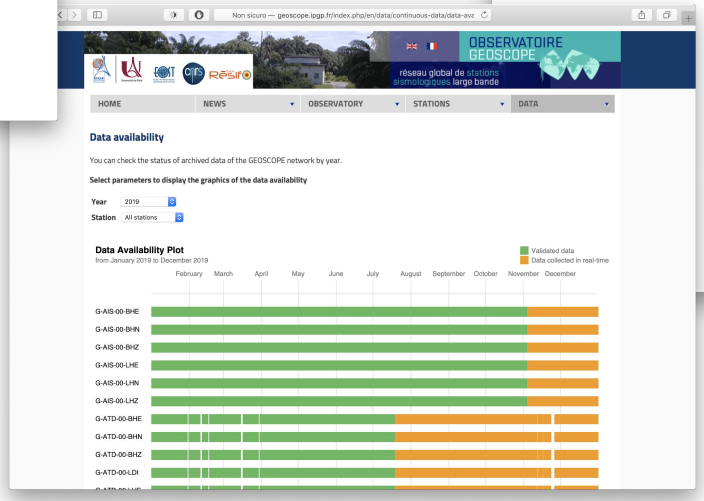
Data availability webpages @IPGP-DC



http://volobsis.ipgp.fr/inventory.php

https://www.seis-insight.eu/en/science/seis-data/seis-data-availability





http://geoscope.ipgp.fr/index.php/en/data/continuous-data/data-availability

CC BY

Other webservices

- resifws-timeseries and irisws-timeseries
  - Access time series data in different formats (e.g., SAC, ASCII, PNG, JPEG) and with optional signal processing (e.g., filtering, deconvolution).
- resifws-timeseriesplot and irisws-timeseriesplot
  - Plot time series data. Can make interactive plots (RESIFWS), helicorders (IRISWS).
- eidaws-wfcatalog (also available from RESIF)
  - Provides detailed information on the contents of waveform data including quality control parameters.
- the full family of IRISWS
  - IRIS provides **a lot** of custom-made web services (for data junkies 🤓).
- IRIS Mustang
  - Webservice for data quality metrics: noise-psd, noise-pdf, spectrogram (and much more… 😲).
- IRIS PH5WS
  - "FDSN-like" access to PH5 data and metadata (ex: large-N experiments using nodes).

CC BY

Volcano Reports web service @IPGP-DC

Available at:

https://ws.ipgp.fr/volcano-reports/

- Access to volcano activity reports
- Connected to EPOS - Volcano Observations

IPGP / EPOS - Volcano activity reports webservice

The Volcano Activity Reports webservice provides access to volcano activity reports metadata in EPOS activity report JSON format. The metadata may be filtered e.g. by time and agency. The *HTTP-GET* method is used to request data.

Available URLs

- http://ws.ipgp.fr/volcano-reports/report

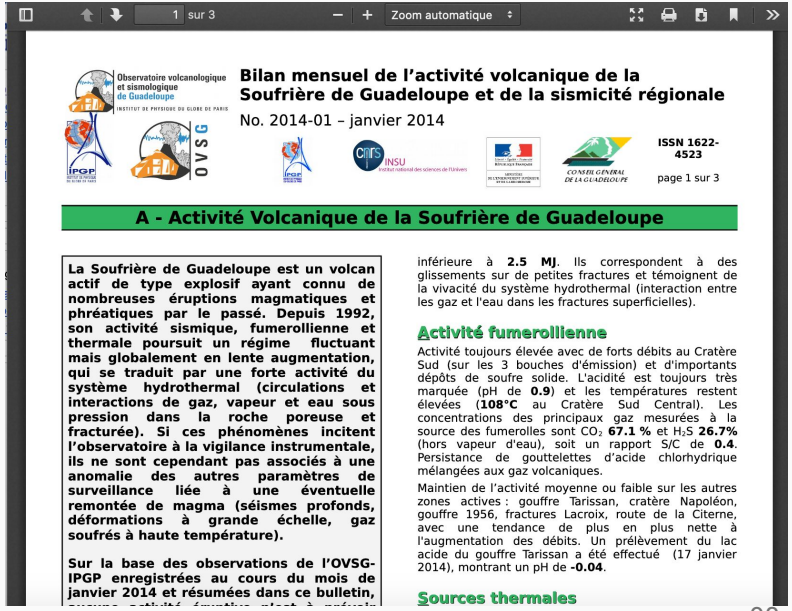
Service description :

- Default response :
  - If no parameters are provided, the webservice returns the last report added.
- Query example :
  - http://ws.ipgp.fr/volcano-reports/report?start_time=2014-01-01&end_time=2014-04-01&agency=OVSG&type=periodic
- Available request parameters :
  - *start_time*
    - Start date
    - required format : YYYY-MM-DD
    - default: none
  - *end_time*
    - End date
    - required format : YYYY-MM-DD
    - default: none
  - *agency*
    - Abbreviation of emmiting agencies
    - standard: [OVSM, OVSG, OVPF]
    - default: none
  - *volcano*
    - Volcano name in french, si, or volcano ICAO code.
    - IPGP values : [Soufriere Guadeloupe, Pelee, Piton de la Fournaise, Soufriere Hills]
    - default: none
  - *type*
    - Type of report
    - standard: [VONA, weekly, periodic, exceptional, eruption]

Volcano Reports web service @IPGP-DC

https://ws.ipgp.fr/volcano-reports/report?start_time=2014-01-01&end_time=2014-04-01&agency=OVSG&type=periodic


```
0:
  lineage:          "Ce rapport a été rédigé par les équipes des OVS de l'IPGP sur la base de différentes
                    observations de type sismiques, géodésiques, géochimiques."
  domain:
    level0:         "Geohazards"
    level1:         "Volcanic hazard product"
    level2:         "Volcano Notification"
    level3:         "Rapport périodique d'activité volcanique"
  language:         "fr-FR"
  title:            "Rapport périodique d'activité volcanique"
  byteSize:         435239
  abstract:         "Le Rapport périodique d'activité volcanique est un rapport périodique résumant l'activité
                    géophysique d'un ou plusieurs volcans sur la base de différentes observations de type
                    sismiques, géodésiques, géochimiques. Ce rapport synthétise ces observations sur la période
                    et indique l'état d'alerte des volcans concernés. "
  characterset:     "UTF-8"
  responsibleParty:
    affiliation:    "999597417"
    identifier:     "https://orcid.org/0000-0003-4468-2114"
    role:           "Directrice"
    name:           "Céline Dessert"
    email:          "dessert@ipgp.fr"
  uri:              "http://volobsis.ipgp.fr/volcano-reports/2014/01/OVSG_2014-01.pdf"
  accessRights:     "public"
  checksum:
    value:          "85e3467c7042710f03b609b98ac5bc00"
    algorithm:      "md5"
  licence:          ""
```

1 sur 3 — Zoom automatique

Observatoire volcanologique et sismologique de Guadeloupe — INSTITUT DE PHYSIQUE DU GLOBE DE PARIS

Bilan mensuel de l'activité volcanique de la Soufrière de Guadeloupe et de la sismicité régionale

No. 2014-01 – janvier 2014

ISSN 1622-4523

page 1 sur 3

A - Activité Volcanique de la Soufrière de Guadeloupe

La Soufrière de Guadeloupe est un volcan actif de type explosif ayant connu de nombreuses éruptions magmatiques et phréatiques par le passé. Depuis 1992, son activité sismique, fumerollienne et thermale poursuit un régime fluctuant mais globalement en lente augmentation, qui se traduit par une forte activité du système hydrothermal (circulations et interactions de gaz, vapeur et eau sous pression dans la roche poreuse et fracturée). Si ces phénomènes incitent l'observatoire à la vigilance instrumentale, ils ne sont cependant pas associés à une anomalie des autres paramètres de surveillance liée à une éventuelle remontée de magma (séismes profonds, déformations à grande échelle, gaz soufrés à haute température).

Sur la base des observations de l'OVSG-IPGP enregistrées au cours du mois de janvier 2014 et résumées dans ce bulletin,

inférieure à **2.5 MJ**. Ils correspondent à des glissements sur de petites fractures et témoignent de la vivacité du système hydrothermal (interaction entre les gaz et l'eau dans les fractures superficielles).

Activité fumerollienne

Activité toujours élevée avec de forts débits au Cratère Sud (sur les 3 bouches d'émission) et d'importants dépôts de soufre solide. L'acidité est toujours très marquée (pH de **0.9**) et les températures restent élevées (**108°C** au Cratère Sud Central). Les concentrations des principaux gaz mesurées à la source des fumerolles sont CO_2 **67.1 %** et H_2S **26.7%** (hors vapeur d'eau), soit un rapport S/C de **0.4**. Persistance de gouttelettes d'acide chlorhydrique mélangées aux gaz volcaniques.

Maintien de l'activité moyenne ou faible sur les autres zones actives : gouffre Tarissan, cratère Napoléon, gouffre 1956, fractures Lacroix, route de la Citerne, avec une tendance de plus en plus nette à l'augmentation des débits. Un prélèvement du lac acide du gouffre Tarissan a été effectué (17 janvier 2014), montrant un pH de **-0.04**.

Sources thermales

CC BY

Useful links

- SEED manual
  - The Bible (or kind of)
- FDSN
  - The standard body of Seismology
- FDSN StationXML
  - Everything you wanted to know about StationXML* (*but were afraid to ask)
- QuakeML
  - It's complicated... I mean, look at this diagram.
  - A better view of event metadata is provided in the ObsPy documentation.
- miniSEED Matlab: https://github.com/IPGP/mseed-matlab

CC BY