



Glass Frame Work

API

Index

1. Stations Metadata (EPOS GNSS)

- 1.1. Endpoint `/stations/v2/station/{size}/{format}`
- 1.2. Endpoint `/stations/v2/marker/{marker}/{size}/{format}`
- 1.3. Endpoint `/stations/v2/coordinates/{area}/{size}/{format}`
- 1.4. Endpoint `/stations/v2/location/{type}/{instance}/{size}/{format}`
- 1.5. Endpoint `/stations/v2/network/{network}/{size}/{format}`
- 1.6. Endpoint `/stations/v2/combination/{size}/{format}`
- 1.7. Error Explanation

2. Station Metadata (EPOS-ICS)

- 2.1. Endpoint `/stations/v2/bbox/{minLon}/{minLat}/{maxLon}/{maxLat}`
- 2.2. Endpoint `/stations/v2/geojson/{format}/{geojson}`
- 2.3. Error Explanation

3. LogFiles

- 3.1. Endpoint `/log/{marker}`
- 3.2. Endpoint `/log/geodesyml/{marker}`
- 3.3. Error Explanation

4. List of Search Parameters

- 4.1. Endpoint `/stations/v2/list/station`
- 4.2. Endpoint `/stations/v2/list/{data}`
- 4.3. Error Explanation

5. Rinex File Metadata and URLs

- 5.1. Endpoint `/files/{type}/{instance}/{format}`
- 5.2. Endpoint `/files/station-marker/{instance}/{format}`
- 5.3. Endpoint `/files/combination/{instance}/{format}/{page}/{perpage}`
- 5.4. Error Explanation
- 5.5. Query Explanation

6. Rinex File Quality Check Information

- 6.1. Endpoint `/files/combinationT3/{instance}/{format}/{page}/{perpage}`

6.2. Endpoint /files/info

6.3. Error Explanation

7. High Rate Rinex File Metadata and URLs

7.1. Endpoint /files/highrate/station-marker/{instance}/{format}

7.2. Endpoint /stations/v2/highrate/bbox/{minLon}/{minLat}/{maxLon}/{maxLat}

8. Installation Guide

1. Stations Metadata (EPOS GNSS)

1.1. Endpoint /stations/v2/station/{size}/{format}

This endpoint returns information about all stations present in database. Multiple values can be placed for the network, the results of one network or another are shown.

All parameters are mandatory.

Examples

format

The format of the output

- json
- xml
- csv

size

The amount of information provided about each station

- full
- Short

Example Links

- ◆ <https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/station/short/geojson>

1.2. Endpoint /stations/v2/marker/{marker}/{size}/{format}

This endpoint returns information about one or several stations. It is possible to search for stations using the 9Marker.

All parameters are mandatory.

Examples

marker

The marker of station

- ABD000GLP
- ABD000GLP,CASC
- AMEL00NLD,BRAE00GBR

format

The format of the output

- json
- xml
- csv

size

The amount of information provided about each station

- full
- Short

Example Links

- ◆ <https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/marker/CASC00PRT/short/json>
- ◆ <https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/marker/CASC00PRT/full/json>

1.3. Endpoint `/stations/v2/coordinates/{area}/{size}/{format}`

This endpoint returns information about one or several stations. In this endpoint it is possible to search for stations using coordinates of geometric figures.

It is mandatory to fill in the format, size and the area parameter that decides what type of figure we will use.

Examples

area

The type of geometric figure

- circle
- centerlat:32.953368
- centerlon:-17.484741
- radius:147.8

polygon

- polygon=33.146750!-18.116455,33.559707!-16.226807,32.990236!-15.468750,32.166313!-15.106201,31.259770!-15.974121,31.980123!-18.489990

Rectangle

- maxlat:33.385586
- maxlon:-15.567627
- minlat:31.998759
- minlon:-18.105469

format

The format of the output

- json
- xml
- csv

size

The amount of information provided about each station

- full
- Short

Example Links

<https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/coordinates/circle/short/json?centerlat=38.685510¢erlon=-5.449219&radius=928.6>

<https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/coordinates/rectangle/short/json?maxlat=42.163403&maxlon=3.867187&minlat=37.996163&minlon=-19.511719>

<https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/coordinates/polygon/short/json?polygon=44.465151%2C-16.347656%3B40.580585%2C-0.175781%3B32.842674%2C-7.910156%3B36.173357%2C-18.105469%3B41.640078%2C-25.136719>

1.4. Endpoint

/stations/v2/location/{type}/{instance}/{size}/{format}

This endpoint returns information about one or several stations. This endpoint allows you to search for stations based on your location, city, country can be used. state or name.

This endpoint already contains pagination, so its parameters are mandatory except for the pagination parameters that may or may not be inserted.

Example of Station Search by City

type

- city

instance

- Bantry

Example of Station Search by Country

type

- country

instance

- Portugal

Example of Station Search by State

type

- state

instance

- Zamora

Example of Station Search by Name

type

- name

instance

- Benafim

format

The format of the output

- json
- xml
- csv

size

The amount of information provided about each station

- full
- short

page

Number of the page you want to view depends on the number of elements per page chosen.

- 1
- 2
- 3
- 4
- 5
- ...etc
- 0 to view all records

perpage

Number of elements per page.

- 5
- 10
- 20
- 50
- 100
- ...etc

Example Links

- ◆ <https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/location/country/Portugal/short/json?page=1&perpage=50>
- ◆ <https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/location/state/Leiria/short/json>
- ◆ <https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/location/city/Lisboa/short/json?page=1&perpage=50>

1.5. Endpoint `/stations/v2/agency/{agency}/{size}/{format}`

This endpoint returns information about one or several stations. It is possible to search for stations using the agency to which they belong.

All parameters are mandatory.

Examples

agency

The name of the agency

-
- C4G
- OCA
- LGIA
- NIEP
- RGA

format

The format of the output

- json
- xml
- csv

size

The amount of information provided about each station

- full
- Short

Example Links

- ◆ <https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/combination/short/json?city=Cascais&installedDateMax=2007-06-24&page=1&perpage=50>
- ◆ https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/combination/short/json?antenna=TWIVP6050_CONE&page=1&perpage=50
- ◆ <https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/combination/short/json?centerLat=43.834527¢erLon=-22.500000&radius=1126.1&page=1&perpage=50>

1.6. Endpoint `/stations/v2/network/{network}/{size}/{format}`

This endpoint returns information about one or several stations. It is possible to search for stations using the network of the station.

All parameters are mandatory.

Examples

network

The network of station belong

- ReNEP

format

The format of the output

- json
- xml
- csv

size

The amount of information provided about each station

- full
- Short

Example Links

- ◆ <https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/network/ReNEP/short/xml>

1.7. Endpoint /stations/v2/combination/{size}/{format}

This endpoint returns information about one or several stations. This endpoint allows a large combination of elements on one or several stations, which is why there are several parameters in it.

Only the size and format parameters are mandatory, if none of the others are filled in then all stations are listed.

Any of these can be filled in the parameters but look the results are an intersection of the parameters, but each parameter may contain multiple values.

Examples

Below are all the parameters that can be filled in this endpoint, you can use just one or use a combination of several.

- agency
- altitude
- antenna
- centerLat -> For this parameter, you have to fill in the 3 circle parameters: centerLat, centerLon, radius.
- centerLon -> For this parameter, you have to fill in the 3 circle parameters: centerLat, centerLon, radius.
- city
- constellation

- coordinates
- country
- dataAvailability
- dateRange
- date_from
- date_to
- elevangle
- fileType
- frequency
- geojson
- installedDateMax
- installedDateMin
- invertedNetworks
- latitude
- lifetime
- longitude
- marker
- maxAlt
- maxLat -> For this parameter, you have to fill in the 4 rectangle parameters: maxLat,maxLon,minLat,minLon.
- maxLon -> For this parameter, you have to fill in the 4 rectangle parameters: maxLat,maxLon,minLat,minLon.
- minAlt
- minLat -> For this parameter, you have to fill in the 4 rectangle parameters: maxLat,maxLon,minLat,minLon.
- minLon -> For this parameter, you have to fill in the 4 rectangle parameters: maxLat,maxLon,minLat,minLon.
- minimumObservationDays
- minimumObservationYears
- multipathvalue
- nbclockjumps
- nbcycleslips
- network
- observationtype
- polygon
- radius -> For this parameter, you have to fill in the 3 circle parameters: centerLat, centerLon, radius.
- radome
- ratioepoch
- receiver
- removedDateMax
- removedDateMin
- samplingFrequency
- samplingWindow
- satelliteSystem
- site
- state
- statusfile

format

The format of the output

- json
- xml
- csv

size

The amount of information provided about each station

- full
- short

page

Number of the page you want to view depends on the number of elements per page chosen.

- 1
- 2
- 3
- 4
- 5
- 0 to view all records
- Etc

perpage

Number of elements per page.

- 5
- 10
- 20
- 50
- 100
- ...etc

Example Links

- ◆ <https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/combination/short/json?city=Cascais&installedDateMax=2007-06-24&page=1&perpage=50>
- ◆ https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/combination/short/json?antenna=TWIVP6050_CONE&page=1&perpage=50
- ◆ <https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/combination/short/json?centerLat=43.834527¢erLon=-22.500000&radius=1126.1&page=1&perpage=50>

1.8. Error Explanation

Code 200

Ok

- × Everything went well and results returned

Code 400

Bad Request

- × Check the entered parameters;
- × Verifies that the API server link is correct;
- × Checks the entered parameters;
- × If everything above is fine, then the API may not be working.

Wrong full output type.

- × If you are using the wrong answer size, please enter short or full in the size field.

IOException

- × It was not possible to create the file to send the response, possibly an error in the functions to create the files for the response.

SQLException

- × An error in the SQL queries, probably some information that you wanted to retrieve has an error or there is no connection in a table.

ClassNotFoundException

- × The Java class could not be found, this implies a serious error in the API, probably a compilation/installation error.

ValidatorException

- × Something failed to validate the parameters entered, please check the size and characters used

2. Station MetaData (EPOS-ICS Portal)

2.1. Endpoint

/stations/v2/bbox/{minLon}/{minLat}/{maxLon}/{maxLat}

This endpoint returns all the stations inside the bounding box. The output format is GeoJson.

The parameters are the dimensions of the bounding box so they are all mandatory.

The coordinates shown in the example form a rectangle around Portugal.

Examples

maxLat

The maximum Latitude of the bounding box

- 44.08

maxLon

The maximum Longitude of the bounding box

- -5.53

minLat

The minimum Latitude of the bounding box

- 35.10

minLon

The minimum Longitude of the bounding box

- -12.21

Example Links

- ◆ <https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/station/bbox/-32.5/29.22/14.58/49.03>

2.2. Endpoint **/stations/v2/geojson/{format}/{geojson}**

This endpoint returns all stations that are inside the geometry of geoJSON.

All parameters are mandatory.

format

The format of the output

- json
- geojson

geojson

The geojson input that must be made on the following website: <https://geojson.io/>

You need to copy all the json code after drawing the figure on the map, after that you can use the following page to minimize the geojson to ensure that it is correctly read by the API: <https://codebeautify.org/jsonminifier>

Then you just need to place the geojson in the respective field

Example Links

- ◆ <https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/station/bbox/-32.5/29.22/14.58/49.03>

2.3. Error Explanation

Code 200

Ok

- × Everything went well and results returned

Code 204

- × The response to your request is empty, please check the request made or check the information in the database.

Code 400

Bad Request

- × Check the entered parameters;
- × Verifies that the API server link is correct;
- × Checks the entered parameters;
- × If everything above is fine, then the API may not be working.

SocketException

- × The connection was closed without the results being returned, something may be wrong in the connection with the database.

SQLException

- × An error in the SQL queries, probably some information that you wanted to retrieve has an error or there is no connection in a table.

ClassNotFoundException

- × The Java class could not be found, this implies a serious error in the API, probably a compilation/installation error.

3. Log Files

3.1. Endpoint /log/{marker}

This endpoint returns the Log File for a station. This endpoint only works on DWG Node.

The parameter is mandatory, 9Marker can be used, and you can use multiple markers on same time.

marker

The marker of the station

- FUNC00PRT
- MIRA00PRT
- FUNC00PRT,MIRA00PRT

Example Links

- ◆ <https://gnssdata-epos.oca.eu/GlassFramework/webresources/log/ABDZ00GLP>
- ◆ <https://gnssdata-epos.oca.eu/GlassFramework/webresources/log/FUNC00PRT,MIRA00PRT>

3.2. Endpoint /log/geodesyml/{marker}

This endpoint returns the Geodesy File for a station. This endpoint only works on DWG Node.

The parameter is mandatory, 9Marker can be used.

marker

The marker of the station

- FUNC00PRT
- MIRA00PRT
- FUNC00PRT,MIRA00PRT

Example Links

- ◆ <https://gnssdata-epos.oca.eu/GlassFramework/webresources/log/geodesyml/ANAY00FRA%20CAND200ESP>
- ◆ <https://gnssdata-epos.oca.eu/GlassFramework/webresources/log/geodesyml/ANAY00FRA>

3.3. Section Error Explanation

Code 200

Ok

- ✘ Everything went well and results returned

Code 204

The request made has a response without content, in this case as the response is a file it may not exist on the machine on which the request is made.

Something went wrong

- ✘ A Java error in the construction of the file name. Try placing the order again or something may be wrong with the code.

Couldn't determine file name for this station -> Unable to get the name of the station, check if it really exists, try it with 4Marker and 9Marker

Code 500

Missing constant in config file.

- ✘ Your Glass.conf file is not configured correctly. Please check the same.

IOException

- ✘ It was not possible to create the file to send the response, possibly an error in the functions to create the files for the response.

ZipException

- ✘ It was not possible to compress the various requested files into a single zip file, try requesting the files individually, otherwise something might be wrong with the API.

4. List of Search Parameters

4.1 Endpoint /stations/v2/list/{data}

This endpoint returns information to the user in the database about one of the criteria chosen in the “data” parameter. Only this parameter is mandatory, if the format parameter is not filled in then the default is json. Furthermore, there is another station_data parameter that should only be used when choosing data = station, which is used to return stations that contain at least one rinex file.

data

The type of data required

- agency
- antenna
- city
- country
- files_type
- network
- radome
- receiver
- state
- station -> station_data:true

format

The format of the response

- json
- list

Example Links

- ◆ https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/list/radome?format=json&station_data=false
- ◆ https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/list/state?format=json&station_data=false
- ◆ https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/list/station?format=list&station_data=true

4.2. Endpoint /stations/v2/list/station

This endpoint simply returns the stations in the database and has some optional parameters that allow filtering of the stations presented. All parameters are optional and when several are filled in, these parameters are intersected.

epos

Whether the station belongs to the EPOS network or not

- epos

- non-epos

marker

The 4 marker of the station

- HERO
- ABBS

markerlongname

The 9 marker of the station

- HERO00GBR
- CASC00PRT

name

The name of the station

- Alcanices
- BOROWIEC

Example Links

- <https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/list/station>
- <https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/list/station?markerlongname=CASC00PRT>

4.3. Error Explanation

Code 200

Ok

- × Everything went well and results returned

Code 204

- × The response to your request is empty, please check the request made or check the information in the database.

Code 400

Bad Request

- × Check the entered parameters;
- × Verifies that the API server link is correct;
- × Checks the entered parameters;
- × If everything above is fine, then the API may not be working.

SocketException

- × The connection was closed without the results being returned, something may be wrong in the connection with the database.

SQLException

- ✘ An error in the SQL queries, probably some information that you wanted to retrieve has an error or there is no connection in a table.

ClassNotFoundException

- ✘ The Java class could not be found, this implies a serious error in the API, probably a compilation/installation error.

5. Rinex File Metadata and URLs

5.1. Endpoint `/files/{type}/{instance}/{format}`

This endpoint returns files from stations that meet the criteria entered in the fields.

Using a specific type then the instance must be a direct match to the type.

In addition to these main fields, there is a field to enter the desired output format. Two required fields for pagination, where you can enter the number of elements per page to be displayed and the page you want to be displayed. In addition to a parameter that allows you to exclude files with a status less than zero.

Examples

Example of File Search by Agency

type

- agency

instance

- Ordnance Survey

Example of File Search by Coordinates

type

- coordinates

instance

- minLat=35&maxLat=40&minLon=0&maxLon=10

Example of File Search by Coordinates Data

type

- coordinates_data

instance

- coordinates=rectangle&maxLat=59.60&minLat=55.42&maxLon=32.43&minLon=21.31

Example of File Search by Country

type

- country

instance

- Portugal

Example of File Search by Date Range

type

- date_range

instance

- dateRange=2023-01-01,2023-01-10

Example of File Search by Network

type

- network

instance

- C4G

Example of File Search by Published Date of File

type

- published_date

instance

- date_from=2023-01-01&date_to=2023-01-02
-

Example of File Search by Type of Data Center Acronym

type

- data_center_acronym

instance

- EPOSGNSS

You can find all Acronimos Data Centers at this link: <https://gnssdata-epos.oca.eu/GlassFramework/webresources/t0-manager/getDataCenter>

format

The format of the output

- csv
- json
- script
- xml

filtervalidatedfiles

Whether to exclude files with status less than zero. Value 1 to exclude files, or 0 to view all files regardless of status.

- 1
- 0

Examples Links

- ◆ <https://gnssdata-epos.oca.eu/GlassFramework/webresources/files/marker/CASC/xml?filtervalidatedfiles=0&page=1&perpage=50>
- ◆ <https://gnssdata-epos.oca.eu/GlassFramework/webresources/files/combination/marker%3DCASC%26dateRange%3D2023-01-01%2C2023-01-10/json?filtervalidatedfiles=0&page=1&perpage=50>

5.2. Endpoint /files/station-marker/{instance}/{format}

This endpoint returns files from stations that meet the station-marker inserted.

To use this endpoint it is only necessary to insert a station nine-character marker in the station-marker field.

In addition to these main fields, there is a field to enter the desired output format. Two optional fields for pagination, where you can enter the number of elements per page to be displayed and the page you want to be displayed. In addition to a parameter that allows you to exclude files with a status less than zero.

station-marker

The station-marker of the station. Can be nine marker. One or more markers can be combined as long as they are separated by a comma.

- AJAC00FRA
- CASC00PRT
- AJAC00FRA,CASC00PRT

page

Number of the page you want to view depends on the number of elements per page chosen.

- 1
- 2
- 3
- 4
- 5
- 0 to view all records
- ... etc

perpage

Number of elements per page.

- 5
- 10
- 20
- 50
- 100
- ...etc

format

The format of the output

- csv
- json
- script
- xml

filtervalidatedfiles

Whether to exclude files with status less than zero. Value 1 to exclude files, or 0 to view all files regardless of status.

- 1
- 0

5.3. Endpoint

/files/combination/{instance}/{format}/{page}/{perpage}

This endpoint returns files from files that meet the criteria entered in the fields.

This endpoint is used to search files using a combination of several different criteria, allowing you to reach impossible combinations or some very specific ones.

In addition to these main fields, there is a field to enter the desired output format. Two required fields for pagination, where you can enter the number of elements per page to be displayed and the page you want to be displayed. In addition to a parameter that allows you to exclude files with a status less than zero.

All possible parameters to combine

- agency
- antenna
- coordinates[°]
- coordinates_data
- coordinates_data_list
- country
- date_range^{°°}
- file_name
- format
- marker or station-marker
- md5
- network
- published_date^{°°}
- radome
- receiver
- sampling_frequency
- station_dates^{°°}
- data_center_acronym^{°°}

[°] – this object uses “minLat=X&maxLat=X&minLon=X&maxLon=X” as it’s object instance.

^{°°} – this objects use “date_from=X&date_to=X” as it’s object instance.

°°° – this objects use “date_range=2020-01-01,2022-01-01 or published_date=2020-01-01,2022-01-01” as it’s object instance.

To get all Data Center Acronym:

<https://gnssdata-epos.oca.eu/GlassFramework/webresources/t0-manager/getDataCenter>

Instance

The combination of parameters. Combinations can be made with all the parameters presented as long as they are separated by &..

- marker=CASC00PRT&dateRange=2023-01-01,2023-01-20
- agency=Ordnance Survey&city=Aberporth
- published_date=2020-01-01,2022-01-01

page

Number of the page you want to view depends on the number of elements per page chosen.

- 1
- 2
- 3
- 4
- 5
- 0 to view all records
- ... etc

perpage

Number of elements per page.

- 5
- 10
- 20
- 50
- 100
- ...etc

format

The format of the output

- csv
- json
- script
- xml

filtervalidatedfiles

Whether to exclude files with status less than zero. Value 1 to exclude files, or 0 to view all files regardless of status.

- 1

- 0

5.4. Error Explanation

Code 200

Ok

- Everything went well and results returned

Ok but no result

- Nothing was found in the database that matches the request made

SocketException

- The connection was closed without the results being returned, something may be wrong in the connection with the database.

SQLException

- An error in the SQL queries, probably some information that you wanted to retrieve has an error or there is no connection in a table.

ClassNotFoundException

- The Java class could not be found, this implies a serious error in the API, probably a compilation/installation error.

Error: "format" Format Unknown

- The response format entered in the parameters is invalid, you must only use json, xml, csv or script.

Code 204

- The response to your request is empty, please check the request made or check the information in the database.

Code 400

Bad Request

- Check the entered parameters;
- Verifies that the API server link is correct;
- Checks the entered parameters;
- If everything above is fine, then the API may not be working.

ValidatorException

- Something failed to validate the parameters entered, please check the size and characters used

Invalid object type

- The parameter entered in type is not valid, please confirm that it is correct.

Invalid parameters [1]

- The number of parameters entered is incorrect.

Invalid parameters [2]

- Parameter names are incorrect or not entered correctly

Code 500

Internal Server Error

- Something serious has happened, the response may contain too much information and the server cannot respond or an internal server error prevents the response.

5.5. Query Explanation

This section shows the queries made to the database, so you can understand what needs to be filled in each field to find the desired files.

Type	Table	Field	Criterion
name	station	name	like%
marker	station	marker	=
network	network	name	=
type	station_type	name	ilike
antenna	antenna_type	name	ilike
receiver	receiver_type	name	ilike
radome	radome_type	name	=
country	country	name	=
state	state	name	=
city	city	name	ilike
agency	agency	name	ilike
sampling_frequency	file_type	sampling_frequency	=
length	file_type	sampling_window	=
coordinates_data	coordinates	lat,lon	<>=
format	file_type	format	%ilike
coordinates	coordinates	Lat,lon	<>=
date_range	rinex_file	reference_date	<>=
station_dates	station	date_from, date_to	<>=
published_date	rinex_file	published_date	<>=
md5	rinex_file	md5checksum	=
file_name	rinex_file	name	=
data_center_acronym	data_center	acronym	=

6. Rinex File Quality Check Information

6.1. Endpoint

/files/combinationT3/{instance}/{format}/{page}/{perpage}

This endpoint returns file information, including information about tables that are part of T3 from files that meet the criteria entered in the fields.

This endpoint is used to search files using a combination of several different criteria, allowing you to reach impossible combinations or some very specific ones.

In addition to these main fields, there is a field to enter the desired output format. Two required fields for pagination, where you can enter the number of elements per page to be displayed and the page you want to be displayed. In addition to a parameter that allows you to exclude files with a status less than zero.

All possible parameters to combine

- agency
- antenna
- coordinates[°]
- coordinates_data
- coordinates_data_list
- country
- date_range^{°°°}
- file_name
- format
- marker or station-marker
- md5
- network
- published_date^{°°°}
- radome
- receiver
- sampling_frequency
- station_dates^{°°}
- data_center_acronym^{°°°}

[°] – this object uses “minLat=X&maxLat=X&minLon=X&maxLon=X” as it’s object instance.

^{°°} – this objects use “date_from=X&date_to=X” as it’s object instance.

^{°°°} – this objects use “date_range=2020-01-01,2022-01-01 or published_date=2020-01-01,2022-01-01” as it’s object instance.

To get all Data Center Acronym:

<https://gnssdata-epos.oca.eu/GlassFramework/webresources/t0-manager/getDataCenter>

Instance

The combination of parameters. Combinations can be made with all the parameters presented as long as they are separated by &..

- marker=CASC00PRT&dateRange=2023-01-01,2023-01-20
- agency=Ordnance Survey&city=Aberporth
- published_date=2020-01-01,2022-01-01

page

Number of the page you want to view depends on the number of elements per page chosen.

- 1
- 2
- 3
- 4
- 5
- 0 to view all records
- ... etc

perpage

Number of elements per page.

- 5
- 10
- 20
- 50
- 100
- ...etc

format

The format of the output

- csv
- json
- script
- xml

filtervalidatedfiles

Whether to exclude files with status less than zero. Value 1 to exclude files, or 0 to view all files regardless of status.

- 1
- 0

6.2. Endpoint /files/info

This endpoint is used to obtain information about files. Being a simpler endpoint than the main /files endpoint. In this case, the information is more general. Allows you to search by station where the file belongs, by file status, by date_from or date_to of the file or the acronym of the data center. The response format is always json.

In this endpoint all parameters are optional, just fill in the ones you want.

marker_long_name

The marker_long_name of the station where the file belongs.

- CASC00PRT
- ABAN00ESP
- ACER00ITA

status

The status of the files

- -3
- -2
- -1
- 0
- 1
- 2
- 3

date_from

The date_from of the files

- 2024-08-23

date_to

The date_to of the files

- 2020-12-31

datacenter_acronym

The acronym of the data center

- C4G

page

Number of the page you want to view depends on the number of elements per page chosen.

- 1,2,3,4,5...etc

0 to view all records

perpage

Number of elements per page.

- 5,10,20,50,100...etc

Examples Links

- ◆ https://gnssdata-epos.oca.eu/GlassFramework/webresources/files/info?marker_long_name=CASC00PRT&status=1&datacenter_acronym=C4G&page=1&perpage=50

- ◆ https://gnssdata-epos.oca.eu/GlassFramework/webresources/files/info?status=1&date_from=2002-01-01&date_to=2006-01-01&page=1&perpage=50

6.3. Error Explanation

Code 200

Ok

- Everything went well and results returned

Ok but no result

- Nothing was found in the database that matches the request made

SocketException

- The connection was closed without the results being returned, something may be wrong in the connection with the database.

SQLException

- An error in the SQL queries, probably some information that you wanted to retrieve has an error or there is no connection in a table.

ClassNotFoundException

- The Java class could not be found, this implies a serious error in the API, probably a compilation/installation error.

Error: "format" Format Unknown

- The response format entered in the parameters is invalid, you must only use json, xml, csv or script.

Code 204

- The response to your request is empty, please check the request made or check the information in the database.

Code 400

Bad Request

- Check the entered parameters;
- Verifies that the API server link is correct;
- Checks the entered parameters;
- If everything above is fine, then the API may not be working.

ValidatorException

- Something failed to validate the parameters entered, please check the size and characters used

Invalid object type

- The parameter entered in type is not valid, please confirm that it is correct.

Invalid parameters [1]

- The number of parameters entered is incorrect.

Invalid parameters [2]

- Parameter names are incorrect or not entered correctly

Code 500

Internal Server Error

- Something serious has happened, the response may contain too much information and the server cannot respond or an internal server error prevents the response.

7. High Rate Rinex File Metadata and URLs

7.1. Endpoint `/files/highrate/station-marker/{instance}/{format}`

This endpoint returns the highrate rinex files from stations that meet the station-marker inserted.

To use this endpoint it is only necessary to insert a station marker in the station-marker field, it can be a four-character marker or a nine-character marker.

In addition to these main fields, there is a field to enter the desired output format. Two optional fields for pagination, where you can enter the number of elements per page to be displayed and the page you want to be displayed. In addition to a parameter that allows you to exclude files with a status less than zero.

station-marker

The station-marker of the station. Can be four marker or nine marker. One or more markers can be combined as long as they are separated by a comma.

- AJAC00FRA
- AJAC00FRA,CASC00PRT

page

Number of the page you want to view depends on the number of elements per page chosen.

- 1
- 2
- 3
- 4
- 5
- 0 to view all records
- ... etc

perpage

Number of elements per page.

- 5
- 10
- 20
- 50
- 100
- ...etc

format

The format of the output

- csv
- json

- script
- xml

filtervalidatedfiles

Whether to exclude files with status less than zero. Value 1 to exclude files, or 0 to view all files regardless of status.

- 1
- 0

Example links

- <https://gnssdata-epos.oca.eu/GlassFramework/webresources/files/highrate/station-marker/CASC00PRT/json?page=1&perpage=50>
- <https://gnssdata-epos.oca.eu/GlassFramework/webresources/files/highrate/station-marker/CASC00PRT,ABBS00GBR/json?page=1&perpage=50>

7.2. Endpoint

/stations/v2/highrate/bbox/{minLon}/{minLat}/{maxLon}/{maxLat}

This endpoint returns all the stations inside the bounding box that contain highrate files associated. The output format is GeoJson.

The parameters are the dimensions of the bounding box so they are all mandatory.

The coordinates shown in the example form a rectangle around Portugal.

Examples

maxLat

The maximum Latitude of the bounding box

- 44.08

maxLon

The maximum Longitude of the bounding box

- -5.53

minLat

The minimum Latitude of the bounding box

- 35.10

minLon

The minimum Longitude of the bounding box

- -12.21

Example Links

- ◆ <https://glass.c4g-pt.eu/GlassFramework/webresources/stations/v2/highrate/bbox/-32.5/29.22/14.58/49.03>

8. Installation Guide

8.1. Requirements

Version 1.7.x

- Java Open JDK 8
- Payara 5 or Glassfish 4
- Connection to Database gnns-europe version 1.3.0

Version 2.0 or above

- Java Open JDK 11 or above
- Payara 6
- Connection to Database gnns-europe version 1.3.0

8.2. Installation Java

Choose one of the options taking into account the version of the Glass Framework API used.

Version 1.7.x of GlassFramework API – Java 8

The first thing to do on your system is to install Java

Java 8: <https://www.oracle.com/pt/java/technologies/javase/javase8-archive-downloads.html>

Access one of the previous links and download it according to your operating system and install it on your system.

Otherwise, you have a system with a GUI so you have to follow some commands to carry out the installation, we leave some commands that can serve as a reference below.

For linux distros with yum package manager.

- `sudo yum -y install java-1.8.0-openjdk`
- `sudo yum -y install java-1.8.0-openjdk-devel`

For linux distros with aptitude package manager.

- `sudo apt-get install openjdk-8-jre`
- `sudo apt-get install java-1.8.0-openjdk-devel`

Version 2.0 or higher of GlassFramework API – Java 11

The first thing to do on your system is to install Java.

Java 11: <https://www.oracle.com/pt/java/technologies/javase/jdk11-archive-downloads.html>

Access one of the previous links and download it according to your operating system and install it on your system.

Otherwise you have a system with a GUI so you have to follow some commands to carry out the installation, we leave some commands that can serve as a reference below.

For linux distros with aptitude package manager, for Java 11.

- sudo apt-get install openjdk-11-jdk
- sudo apt-get install openjdk-11-jre

8.3. Installation Payara

The second fundamental part of installing the API comes down to installing Payara. This is an open-source server and will be used to run the Glass Framework API.

Choose one of the options taking into account the version of the Glass Framework API used.

Version 1.7.x of GlassFramework API - Payara 5

To install Payara 5, simply access the link below and the software will automatically download. Payara 5 will only work with Java JDK 8.

<https://repo1.maven.org/maven2/fish/payara/distributions/payara/5.2022.5/payara-5.2022.5.zip>

After downloading, extract the application to a folder of your choice and the server is ready for use.

Version 2.0 or higher of GlassFramework API – Payara 6

To install Payara 6, simply access the link below and the software will automatically download. Payara 6 will only work with Java JDK 11 or higher.

To install Payara 6, access the following link and download the version "Payara Server 6.xxx.x (Full)".

<https://www.payara.fish/downloads/payara-platform-community-edition/>

After downloading, extract the application to a folder of your choice and the server is ready for use.

8.4. Database

In this part it is important to check that the API installation machine is possible to connect to PostegreeSQL where the database is installed.

Make sure you have a user created in the database with "select, insert, update, delete, trigger" requirements to use in the API.

8.5. Glass API Configuration in Payara

In this part of the installation we will configure the GlassAPI files to work with Payara.

Go to GitLab from the following link and choose the version desired from the GlassFrameWork API

https://gitlab.com/gpseurope/EPOS_GLASS_Framework/-/releases

Use the previous link to download the GlassFramework.war file from swagger.

At the next link, download the GLASS.conf file.

https://gitlab.com/gpseurope/EPOS_GLASS_Framework/-/blob/1.7.3/dist/GLASS.conf?ref_type=tags

Let's start by configuring the GLASS.conf file. Open it and edit the following fields to match your connection to PostegreeSQL.

```
# DATABASE
dbip=[DATABASE_SERVER_IP]
dbport=[DATABASE_PORT]
database=[DATABASE_NAME]
dbuser=[DATABASE_USERNAME]
dbpassword=[DATABASE_PASSWORD]
```

To enable GLASS-API to send emails to users the same GLASS.conf file should also include the following:

```
# EMAIL
username=[EMAIL_ADDRESS]
password=[EMAIL_PASSWORD]
smtpserver=[SMTP_SERVER]
port=587
debug=false
```

The Log files part currently only requires configuration of the DGW Node.

After configuring the file, save and place it in the respective Payara folder.

Path: /path_to_payara/payara/glassfish/domains/ [YOUR_DOMAIN_(usually domain1)]
/config/GLASS.conf

After this step we will use the GlassFrameWork.war file, let's start by placing the file in the right Payara folder.

Path: /path_to_payara/payara5/glassfish/domains/ [YOUR_DOMAIN_(usually domain1)] /autodeploy/GlassFramework.war

After placing the . war file in the right folder, we will start payara. Using the following command:

- /path_to_payara/payara/glassfish/bin/asadmin start-domain domain1

If the entire installation is correct then payara will run correctly and create a GlassFramework application in the folder:

/path_to_payara/payara6/glassfish/domains/domain1/applications/GlassFramework/

Go to this folder and open the swagger.json file, let's change its servers. Eliminating all existing servers and creating one with the link to your machine that the API should use.

After successful start of payara, go to your browser and navigate to the link below where you should now see the node management platform.

http://<SERVER_IP>:8080/GlassFramework