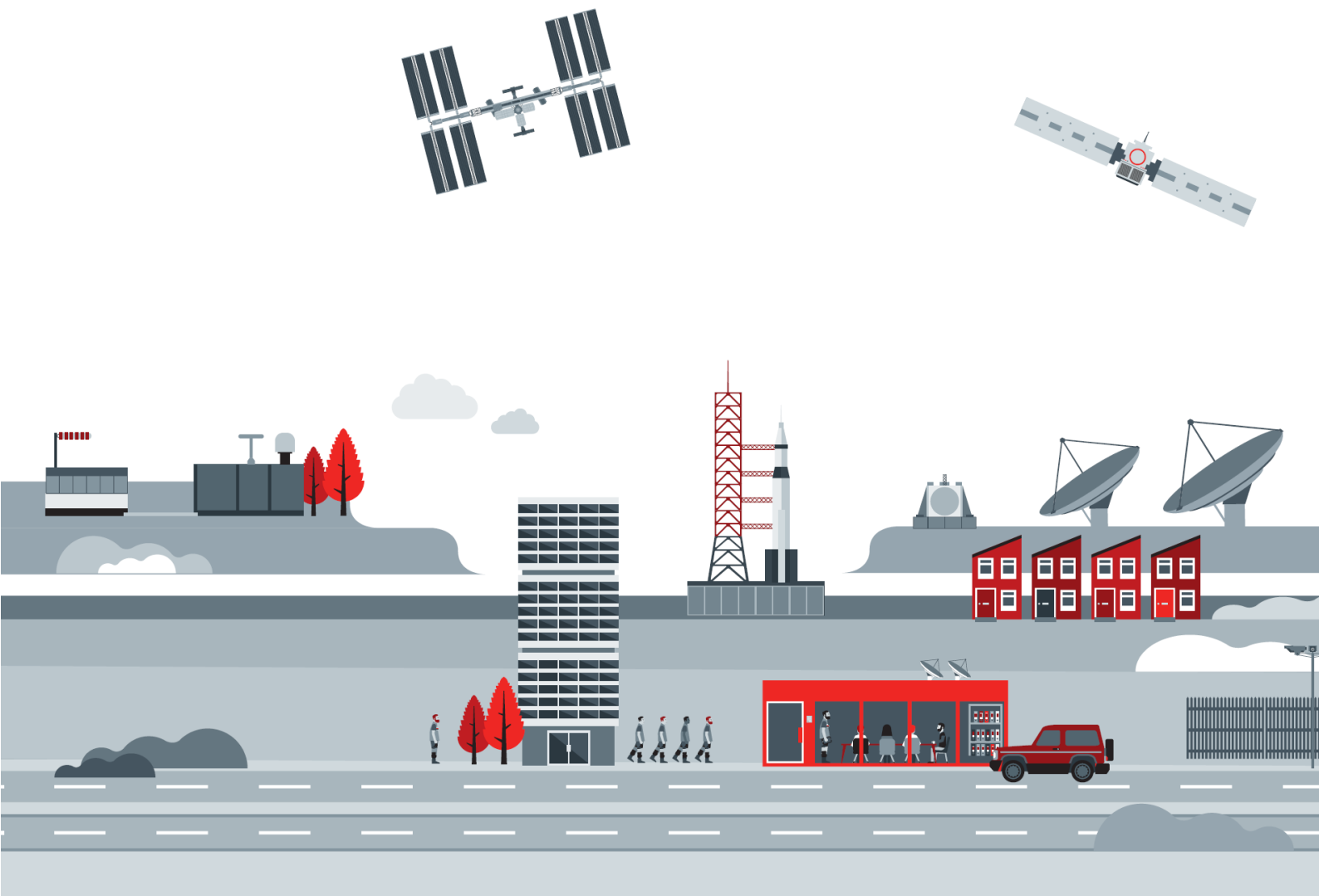


Collaborative Data Hub Software - Maintenance and Evolution Services - Ready for Digital Twin Earth

Semantic Framework Installation and Configuration Manual



| Role/Title | Name | Signature | Date |
|------------|------------------------|-----------|------------|
| Author | Alexandros Vassiliades | | 27/06/2024 |

Change register

| Version/Rev. | Date | Reason |
|--------------|------------|---|
| 1.0 | 08/04/2022 | First release of the document |
| 1.2 | 19/05/2022 | Second release of the document |
| 2.0 | 30/06/2022 | According to DHS #3 release |
| 2.1 | 13/07/2022 | Updates based on comments of DHS #3 release |
| 2.2 | 20/09/2022 | Keycloak integration |
| 2.3 | 24/10/2022 | Updates based on the comments of v2.2 |
| 2.4 | 25/10/2022 | Updates regarding SF-2.2 release |
| 2.5 | 31/10/2022 | Updates regarding SF-2.3 release |
| 2.6 | 8/6/2023 | Updates regarding the DHS #5 release |
| 2.7 | 27/7/2023 | Updates regarding the DHS #6 release |
| 3.4 | 24/05/2024 | Updates regarding SF-3.4 release |
| 3.5 | 27/06/2024 | Updates regarding SF-3.6 release |



Table of Contents

- 1. Introduction 4
 - 1.1 Scope 4
 - 1.2 Purpose 4
 - 1.3 Document applicability 4
 - 1.4 Document structure 4
 - 1.5 Applicable Documents 4
 - 1.6 Reference Documents 5
 - 1.7 Acronyms and Abbreviations 5
- 2. Semantic Framework/Natural Language Installation 7
 - 2.1 Docker compose setup 7
 - 2.2 Keycloak Integration 7
 - 2.3 Installation testing and first population of the KB 10
 - 2.4 Configuration file 10
- 3. Semantic Framework 14
 - 3.1 Service Installation Requirements 14
 - 3.2 Current Features 14
 - 3.3 Semantic Framework Examples 15
 - 3.4 Request for earthquake events on specific city 15
 - 3.5 Request for earthquake events having magnitude greater than specific value 16
 - 3.6 Logs and Outputs 18
 - 3.7 Error messages 19
- 4. Natural Language 24
 - 4.1 Service Installation Requirements 24
 - 4.2 Current Features 24
 - 4.3 Natural Language Examples 24
 - 4.4 Request for earthquake events using NL 24
 - 4.5 Logs and Outputs 27
 - 4.1 Error Messages 28

1. Introduction

1.1 Scope

The scope of this document is to provide instructions to the users to install, configure and use the Semantic Framework (SF), along with its Natural Language (NL) module, within the Collaborative Data Hub Software Maintenance and Evolution Services for Digital Twin Earth.

1.2 Purpose

The purpose of this document is to provide detailed instructions to the users that wish to install and use the Semantic Framework and the Natural Language module. Apart from the installation manual, the document includes some examples of usage for the framework and the expected results, focusing also on log files and error messages.

1.3 Document applicability

Please note that this document is referring to version 3.6, belonging to the DHS#6 release.

| Component | Component Release | DHS Release |
|--------------------|-------------------|-------------|
| Natural Language | V1.7 | DHS#6 |
| Semantic Framework | V3.6 | DHS#6 |

1.4 Document structure

The document is structured as follows:

- Section 1 introduces the SF and the NL module and provides the connection of the framework with the reference documents and the definitions/acronyms reported on this document.
- Section 2 describes the installation instructions for the SF and the NL module and instructions to test the framework, pre-populate the KB and change the configuration files.
- Section 3 provides details related to the requirements that are satisfied by this release of the SF and the currently supported features.
- Section 4 demonstrates some examples of requests towards the SF.

1.5 Applicable Documents

| Ref. | Title | Reference and Version |
|------|---|-----------------------------|
| AD-1 | [AD-SOW] Statement of Work: COLLABORATIVE DATA HUB SOFTWARE - MAINTENANCE AND EVOLUTION SERVICES - READY FOR DIGITAL TWIN | ESA-EOPG-EOPGC-SOW-12, v1.0 |

| Ref. | Title | Reference and Version |
|------|--|--|
| | EARTH | |
| AD-2 | DHS System Design Document | COPE-SERCO-TN-21-1171 |
| AD-3 | Collaborative Data Hub Software – Semantic Framework – ICD | COPE-SERCO-IF-21-1194, v3.5 |
| AD-4 | Semantic Framework SRS | COPE-SERCO-RD-21-1177 - Collaborative Data Hub Software - SRS v6.2 |

1.6 Reference Documents

| Ref. | Title | Reference and Version |
|--------|--|--|
| RD-1. | HTTP Status Codes | < https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html > |
| RD-2. | ONDA-DIAS | https://www.onda-dias.eu/ |
| RD-3. | Data Hub Service (DHuS) | https://colhub.copernicus.eu/ |
| RD-4. | INGV | http://terremoti.ingv.it/en/ |
| RD-5. | OpenSearch Protocol | https://www.opensearch.org/ |
| RD-6. | fdsnws-event API | https://github.com/INGV/openapi |
| RD-7. | Open Data Protocol (OData) | https://www.odata.org/ |
| RD-8. | Transformed and clean cities list file combined with country names | https://github.com/SercoSPA/SF-Natural-Language/blob/master/SF_service/cities_countries_clear.json |
| RD-9. | docker-compose.yml | https://github.com/SercoSPA/SF-Natural-Language/blob/master/SF_service/docker-compose.yml |
| RD-10. | configuration.json | https://github.com/SercoSPA/SF-Natural-Language/blob/master/SF_service/config_sf/configuration.json |
| RD-11. | keycloak_configuration.json | https://github.com/SercoSPA/SF-Natural-Language/blob/master/SF_service/config_nl/keycloak_configuration.json https://github.com/SercoSPA/SF-Natural-Language/blob/master/SF_service/config_sf/keycloak_configuration.json |
| RD-12. | nl_logger_config.json | https://github.com/SercoSPA/SF-Natural-Language/blob/master/SF_service/config_nl/nl_logger_config.json |

1.7 Acronyms and Abbreviations

| Acronym | Definition |
|---------|---------------------|
| AD | Applicable Document |

| | |
|-------|------------------------------------|
| API | Application Programming Interface |
| DHS | Data Hub Software |
| DHuS | Data Hub Service |
| DR | Data Receiver |
| ESA | European Space Agency |
| HTTP | Hypertext Transfer Protocol Secure |
| HTTPS | Hypertext Transfer Protocol Secure |
| ICD | Interface Control Document |
| KB | Knowledge Base |
| OData | Open Data Protocol |
| RD | Reference Document |
| SF | Semantic Framework |
| URL | Uniform Resource Locator |
| NL | Natural Language |

2. Semantic Framework/Natural Language Installation

The Semantic Framework is a component of the Copernicus Sentinels Collaborative Data Hub Software (DHS) that aims in the detection of earthquake events and Sentinel-1 products from various data providers corresponding to the time before and after the event. The criteria of each search are defined by the user.

2.1 Docker compose setup

The installation of the SF is achieved using the docker-compose. The following technologies should be installed in order to successfully run the framework:

- Docker version ($\geq 23.0.5$)
- Docker engine ($\geq 23.0.5$)
- docker-compose ($\geq 2.17.3$)

The docker image size is approximately 933 MB for the semantic framework, 123 MB for the KB and 1.98 GB for the NL module.

Moreover, the network "token-network" defined in the docker-compose.yml configuration file shall be created to allow a proper communication between different services.

To successfully install the framework:

- Download the "SF_service" folder from the github repository (https://github.com/DHS-SemanticFramework/SF-Natural-Language/tree/main/SF_service)
- Set up Keycloak on your side (see 2.2.) and run it as a service.
- Configure if wanted the "docker-compose.yml" file.
- Run the "docker-compose.yml" file with the command: **docker-compose up -d**

Note: In case, the "container_name" or the port of the "sf-api" changes, the environmental variable "SF_ENDPOINT" should also be changed accordingly.

2.2 Keycloak Integration

In order to allow only authenticated end users (with basic authentication) to interact with the Semantic Framework (i.e. perform queries), it is required to integrate the SF with the Keycloak interface. The steps to successfully achieve that are the following:

Step 1. Login to Keycloak online interface¹

- using the administration console and a valid set of credentials.

Step 2. Client Creation

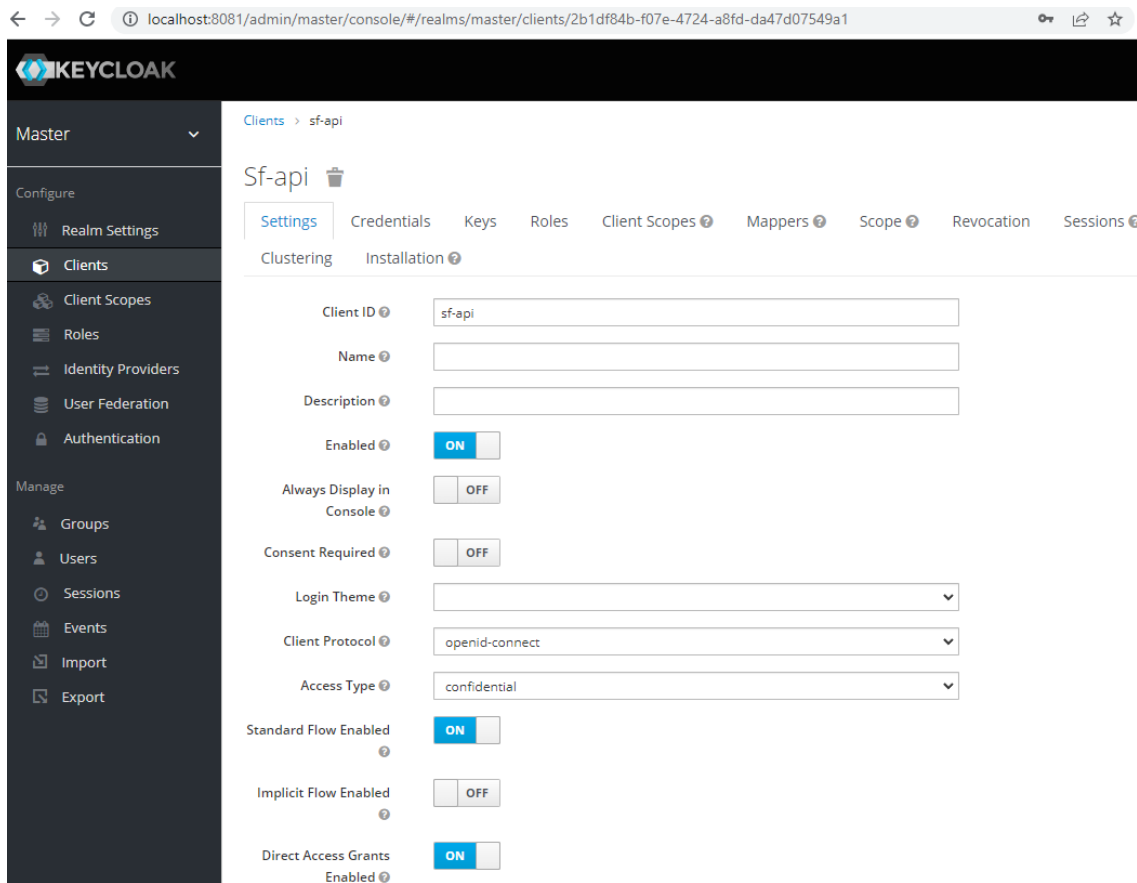
¹ <http://<IP>:<PORT>/auth/>

- Create a new client with the following requirements:


| Field | Value |
|----------------------------|-----------------------------|
| id | <client-name> (i.e. sf-api) |
| Access type | confidential |
| valid redirect URIs | http://<IP>:<PORT>/* |




And select "Save".


- Use the URL path /clients/<uuid> to get the UUID value (i.e. 2b1df84b-f07e-4724-a8fd-da47d07549a1) and add it to "client_container_id" field of the two "keycloak_configuration.json" files under the directories of "SF_service/config_sf/keycloak_configuration.json" and "SF_service/config_nl/keycloak_configuration.json"
- Both of those files require the same credentials based on your keycloak setting.





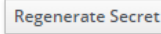
- Navigate to the tab "Credentials". Copy "Secret" and add it to "client_secret" field of the two "keycloak_configuration.json" files under the directories of "SF_service/config_sf/keycloak_configuration.json" and "SF_service/config_nl/keycloak_configuration.json "
- Both of those files require the same credentials based on your keycloak setting.



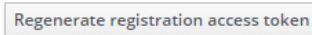
Sf-api 

Settings **Credentials** Keys Roles Client Scopes  Mappers  Scope  Revocation

Clustering Installation 

Client Authenticator  Client Id and Secret 

Secret LjJmpOjtsnt4GnbnWIZSQNYbgh5qcpX 

Registration access token   

- Navigate to the tab "Roles" and execute the following
 - Add Role>Role Name "Search">Save
 - Add Role>Role Name "Semantic Framework Manager">Save

Step 3. Create new users

- Navigate to "Manage">"Users"
- Add a user with specific username (e.g. searchuser) and select save
- Navigate to tab "Credentials". Set password (e.g. 1234) and Temporary value to "OFF".
- Navigate to tab "role mappings". Set Client roles>sf-api, select "Search" and "add selected".
- The same procedure should be followed to add a new Semantic Framework Manager user.

Step 4. Configure the two "keycloak_configuration.json" files (under the directories of "SF_service/config_sf/keycloak_configuration.json" and "SF_service/config_nl/keycloak_configuration.json")

- Add a valid set of administrator username and password. As administrator we consider a person that has the credentials to login to Keycloak online interface and execute the aforementioned steps (create a client, create roles and users).
- Add the URL to access Keycloak, the realm and the client name on the corresponding fields.
- The URL should remain unchanged if keycloak is in a docker container named "kc-main" and belongs to the network named "token-network" (i.e. leave it as "http://kc-main:8080/").
- An example of the expected structure for each "keycloak_configuration.json" file is the following ("test" values are to be replaced):

```
{
  "client_secret": "test",
  "client_container_id": " test ",
  "client_name": "sf-api",
  "URL": "http://kc-main:8080/",
  "realm": "master",
  "adminUsername":"test",
  "adminPassword":" test" }
```

2.3 Installation testing and first population of the KB

Calling the following URL will pre-populate the KB with location-related information and it will indicate a successful installation of SF:

```
http://<IP>:<PORT>/SemanticFramework/api/population
```

It is worth mentioning that this procedure should run only once, during the installation of the framework. Any user that corresponds to "Semantic Framework Manager" user role can execute this action, using the credentials (username/password) that have been generated on section 2.2 - Step 3. The request should be executed using POST type and Content-Type should be set to "application/json". The request body in this case, can be empty and the expected output is:

```
{
  "status": "Successfully added to KB."
}
```

Receiving this message indicates that the KB now contains cities, the countries that they belong to, and the coordinates of them. The default <PORT> is 8087 and localhost can be used in the <IP> field to test the framework locally. In case needed, the port can be changed from the "docker-compose.yml" by changing the sf-api ports:

```
sf-api:
  ...
  ports:
    - '8087:8080'
```

In this case the SF is exposed in port 8087. The KB is exposed in ports 8890 and 1111. The DR is exposed on port 8088. The NL module is exposed on port 8000. To change the exposed ports of the KB, change the left-part number on the "docker-compose.yml" file.

2.4 Configuration file

The expected structure of the configuration file is the following and is also defined in "config_sf/configuration.json":

```
{
  "resultsPerPage": "2",
  "sources": [
    {
      "dataSource": "https://catalogue.onda-dias.eu/dias-catalogue/Products?$search=%22footprint:%20%22Intersects(POINT([eventLat]%20[eventLong]))%"
```

```

22%20AND%20%20beginPosition:[[eventStartTime]T00:00:00.000Z%20TO%20[eventEndTime]T00:
00:00:00.000Z]%20AND%20platformShortName:S1%20AND%20productType:IW_SLC__1S%22&$forma
t=json",
  "username": "null",
  "password": "null",
  "type": "copernicus",
  "associatedId": "1"
},
{
  "DHuSAddress": "colhub.copernicus.eu/dhus",
  "dataSource": "https://[DHuSAddress]/search?start=0&rows=100&q=(platformname:Sentinel-
1%20AND%20producttype:IW_SLC__1S%20AND%20sensoroperationalmode:IW%20AND%20footpri
nt:%22Intersects(POINT([eventLat]%20[eventLong]))%22%20AND%20beginposition:[[eventStartTi
me]T00:00:00.000Z%20TO%20[eventEndTime]T00:00:00.000Z])&format=json",
  "username": "<username>",
  "password": "<password>",
  "type": "copernicus",
  "associatedId": "1"
},
{
  "dataSource":
"https://webservices.ingv.it/fdsnws/event/1/query?starttime=[SelectedDate]T00:00:00&endtime=[En
dDate]T23:59:59&minmagnitude=[magnitude]&format=text&lat=[CityLat]&lon=[CityLong]&maxradiu
skm=50&limit=[limit]&offset=[offset]",
  "username": "null",
  "password": "null",
  "eventType": "earthquake",
  "associatedId": "1"
},
{
  "DHuSAddress": "colhub.copernicus.eu/dhus",
  "dataSource": "https://[DHuSAddress]/search?start=0&rows=100&q=(platformname:Sentinel-
2%20AND%20producttype:S2MSI2A%20AND%20footprint:%22Intersects(POINT([eventLat]%20[eve
ntLong]))%22%20AND%20beginposition:[[eventStartTime]T00:00:00.000Z%20TO%20[eventEndTim
e]T00:00:00.000Z])&format=json",
  "username": "<username>",
  "password": "<password>",
  "type": "copernicus",
  "associatedId": "2"
},
{
  "dataSource": "https://data.seattle.gov/resource/kzjm-

```

```

xkqj.json?${limit}=${limit}&${offset}=${offset}&${where}=within_circle(report_location,%20[CityLat],%20[CityLong],%2050)%20and%20datetime%20between%20%27[SelectedDate]T00:00:00%27%20and%20%27[EndDate]T23:59:59%27",
  "username": "null",
  "password": "null",
  "additionalFields": ["incidentType", "eventId"],
  "eventType": "firecall",
  "associatedId": "2"
},
{
  "dataSource": "https://colhub.copernicus.eu/dhus/search?q=uuid:[productUUID]&format=json",
  "username": "<username>",
  "password": "<password>",
  "type": "orbitPass"
}
]
}

```

More specifically, the "resultsPerPage" field is defined to support pagination and more specifically the expected events number per page. The first two sources correspond to Copernicus data sources, while the third to event data source (i.e. earthquake). The next source corresponds to Copernicus data source and the last to event data source. In terms of Copernicus data sources, "type" should be set to "copernicus". The data source query should not be changed on the configuration file. In terms of earthquake events "eventType" should be set to "earthquake". Authentication is not needed in INGV and ONDA-DIAS data sources and in such cases username and password can be set to "null". In DHuS data sources, username and password are required. The field "associatedId" corresponds to matching Copernicus images levels to event types, which might be useful in future versions of the framework to allow the extensibility of this scenario. It is worth mentioning that for each associatedId, a new Copernicus data source should be added with the expected parameters (e.g. sentinel level, product type, etc).

Copernicus data sources can be omitted or added in the configuration file. In cases where a data source needs to be omitted, the whole data source should be deleted from the configuration file (i.e. rows 4-10 for removing ONDA-DIAS). At least one Copernicus data source and one event data source should be present on the configuration file to ensure the correct operation of the SF. In case that you want to include a new DHuS data source, you can change the "DHuSAddress" source including username and password if needed.

To efficiently extend the framework and include a new use case, additional data sources should be added on the configuration file. The corresponding Copernicus data sources and associatedId should be set, as mentioned above. On the event data sources, "additionalFields" might be included as a json array which should contain the additional fields that a user wants to add on the response, with the exactly same field names as those are returned by the Data Receiver (taking into consideration the Data Translation process).

Collaborative Data Hub Software - Maintenance and Evolution Services - Ready for Digital Twin Earth

In case that configuration file has been changed, restarting the framework is required, to ensure the successful execution of it. Restarting the framework can be achieved by executing: **docker-compose down** (on the path that "docker-compose.yml" is stored) and after that: **docker-compose up -d**.

It also worth mentioning the last entry of the configuration file, which has the "type" *orbitPass*, that allows us to access information about the orbit and pass direction of the products that are returned in a user's query.

3. Semantic Framework

The currently available version of the semantic framework is v3.6. On this section we focus on the features that are currently supported during that release of the framework.

3.1 Service Installation Requirements

The SF can easily be installed locally by any authorized user using the installation instructions reported in section 2. To interact directly with the service, at this release, a client should be generated (i.e. using POSTMAN) that executes "POST" requests over the specified URLs. The "Content-type" should be set to "application/json". The sf-api interacts with the dr-api, which communicates with the data sources (i.e. INGV, Data Hub Service (DHuS), ONDA-DIAS), and with the KB, which is exposed using the docker-compose and returns the results that correspond to the criteria defined by the user.

3.2 Current Features

The latest version of the SF contains a wide list of features that match the requirements reported in [AD-4]. Some of the most important are reported below:

- Ability to combine earthquake events with Sentinel-1 images before and after the event corresponding to a specific city and time frame using the available API.
- Ability to combine earthquake events with Sentinel-1 images before and after the event corresponding to a specific city and time frame using the available API, in order to achieve interferometry.
- Location expressed as city or country.
- Error messages in cases input is not provided as expected.
- Log file to monitor possible errors, exceptions and the responses to the user.
- Dockerization of the service and deployment using a docker-compose.
- Code publishing using GitHub and DockerHub.
- User access monitoring using Keycloak.
- Ability to include new data sources.
- Ability to include new scenarios and make associations of new types of events and products from different level and/or type with minor modifications or the development of a DR.
- Modular result schema, which is easily extensible by including the required fields on the configuration file.
- Communication mechanism between the Natural Language module and the SF, based on which the SF can receive the messages that arrive from the Natural Language mechanism.
- Communication processing mechanism based on which the SF can process the messages arriving from the Natural Language module and extract the crucial information.
- Handling error mechanism for errors coming from the Natural Language module. In other words the SF can provide error messages for errors arriving from the Natural Language module.

It is worth mentioning that the framework is licensed under the GNU AFFERO GENERAL PUBLIC LICENSE Version 3.

3.3 Semantic Framework Examples

In this section we provide a guideline with some examples for the users that want to access the Semantic Framework.

3.4 Request for earthquake events on specific city

After performing the initial step described in section 2.3, the users with "Search" user role are able to access the following URL in order to execute their queries:

```
http://<IP>:<PORT>/SemanticFramework/api/retrieve
```

using basic authentication on the authorization parameters and the credentials (username and password) that have been generated on section 2.2 - Step 3 for this user role.

At this point we would like to underline that the user can either directly address queries to the SF with abovementioned URL, and use an input as the one shown below, or (s)he can use the NL module using free language through the following endpoint: `http://<IP>:<PORT>/NaturalLanguage`.

As already mentioned, the request should be executed using POST type and Content-Type should be defined to "application/json". The request body in this case, should be a JSON having the following structure:

```
{
  'text': 'I search earthquake located in Bayugan, Philippines in 2024 with magnitude greater than 5.5',
  'page': '1',
  'nlp':
    {'event': 'earthquake', 'city': 'Bayugan', 'country': 'Philippines', 'year': 2024, 'month': 'null', 'day': 'null', 'magnitude': 5.5, 'comparative': '>', 'point': False, 'latitude': 'null', 'lognitude': 'null'}
}
```

The response in this case is the following:

```
[{"year": "2024", "city": "Bayugan", "country": "Philippines", "location_coordinates": "POINT (8.75611 125.7675)", "images_before1": {"link": "https://colhub.copernicus.eu/dhus/odata/v1/Products('e92537ab-d2f5-42f7-b2bb-c7c483b13486')", "sensing_date": "2024-02-02T21:24:16.705Z", "location": "POLYGON ((126.804024 7.050664, 127.139153 8.680891, 124.885643 9.133541, 124.559784 7.507817, 126.804024 7.050664))", "orbit_number": "163", "pass_direction": "DESCENDING"}, "image_after1": {"link": "https://colhub.copernicus.eu/dhus/odata/v1/Products('0e593b1a-b6be-4bb7-878a-f4d124cb88d4')", "sensing_date": "2024-02-14T21:24:16.443Z", "location": "POLYGON ((126.803497 7.050778, 127.138573 8.680758, 124.885063 9.133412, 124.55925 7.507935, 126.803497 7.050778))", "orbit_number": "163", "pass_direction": "DESCENDING"}, "magnitude": {"value": "5.7"}, "depth": {"value": "19.5"}, "epicentral_location": {"latitude": {"value": "8.65547"}, "longitude": {"value": "125.698"}}, "timestamp": "2024-02-10T03:22:07.16Z"}]
```

Alternatively, SF can be accessed using a cURL request, such as the following:

```
curl -X POST http://localhost:8087/SemanticFramework/api/retrieve -H "Content-type: application/json" -H "Authorization: Basic c2VhcmNodXNlcjoxMjM0NQ==" -d '{"page": "1", "nlp": {"event": "earthquake", "city": "Bayugan", "country": "Philippines", "year": 2024, "month": "null", "day": "null", "magnitude": 5.5, "comparative": ">", "point": false, "latitude": 0, "lognitude": 0}}'
```

containing a valid set of user credentials and a request body following the abovementioned structure.

3.5 Request for earthquake events having magnitude greater than specific value

In another example that the user requests for " I want an earthquake located in Athens, Greece during July 2019 with a magnitude value that is greater than 5.2", give as input:

```
{
  "text": " I want an earthquake located in Athens, Greece during July 2019 with a magnitude value that is greater than 5.2",
  "page": "1",
  "nlp": {
    "event": "earthquake",
    "city": " Athens ",
    "country": " Greece ",
    "year": "2019",
    "month": "07",
    "day": "19",
    "magnitude": 5.2,
    "comparative": ">",
    "point": false,
    "latitude": "null",
    "lognitude": "null"
  }
}
```

the expected output is:

```
[{"year":"2019","month":"07","day":"19","city":"Athens","country":"Greece","location_coordinates":"POINT (37.9755648 23.7348324)","images_before1":{"link":"https://catalogue.onda-dias.eu/dias-catalogue/Products(04c87b0b-9b06-43ea-a375-dd36fa83f6fe)","sensing_date":"2019-07-12T04:31:28.174Z","location":"MULTIPOLYGON (((24.452286 37.946476, 24.857328 39.573891, 21.898474 39.973515, 21.563047 38.347275, 24.452286 37.946476)))"},"image_after1":{"link":"https://catalogue.onda-dias.eu/dias-catalogue/Products(fb997d3c-0a3d-4d10-b9f2-46fb16cc982a)","sensing_date":"2019-07-24T04:31:29.01Z","location":"MULTIPOLYGON (((24.452011 37.945801, 24.857298 39.574074, 21.898468 39.973724, 21.562834 38.346619, 24.452011 37.945801)))"},"images_before2":{"link":"https://catalogue.onda-dias.eu/dias-catalogue/Products(50c4f69d-cc5a-4b13-ad78-440fd267231a)","sensing_date":"2019-07-12T04:31:52.995Z","location":"MULTIPOLYGON (((24.091356 36.453266, 24.487803 38.081844,
```



```

21.592066      38.4827,      21.26038      36.855572,      24.091356
36.453266))))}, "image_after2": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(ff23131f-b523-405e-bf74-cf36c255976a)", "sensing_date": "2019-07-
24T04:31:53.845Z", "location": "MULTIPOLYGON (((24.091064 36.452332, 24.487598 38.081154,
21.591896      38.48204,      21.260132      36.854664,      24.091064
36.452332))))}, "images_before3": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(b40091ac-2e0a-4329-afd0-39bd82677d39)", "sensing_date": "2019-07-
12T16:22:38.578Z", "location": "MULTIPOLYGON (((22.49559 36.403755, 25.358112
36.81102, 25.028145      38.4314, 22.100489      38.025593, 22.49559
36.403755))))}, "image_after3": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(97d02924-a820-4ce9-935a-bb4fb9f45c7d)", "sensing_date": "2019-07-
24T16:22:39.323Z", "location": "POLYGON ((22.09973 38.024979, 25.027424 38.430843, 25.357224
36.811459,      22.494625      36.404137,      22.09973
38.024979))"}, "images_before4": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(90803615-417c-4e0a-b830-f50dc235a806)", "sensing_date": "2019-07-
12T16:23:03.41Z", "location": "MULTIPOLYGON (((22.131927 37.897034, 25.054323
38.302967, 24.720682      39.922222, 21.728252      39.51746, 22.131927
37.897034))))}, "image_after4": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(951fd7f2-1527-4488-8f7a-9fc21fb853f6)", "sensing_date": "2019-07-
24T16:23:04.134Z", "location": "MULTIPOLYGON (((22.131233 37.896175, 25.053726
38.302174, 24.719849      39.922405, 21.727278      39.517582, 22.131233
37.896175))))}, "images_before5": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(810f9352-7dc3-40c4-b8d6-ef9b94b37804)", "sensing_date": "2019-07-
18T04:30:46.871Z", "location": "MULTIPOLYGON (((24.448282 37.946869, 24.853338
39.57478, 21.89777      39.973873, 21.562277      38.347134, 24.448282
37.946869))))}, "image_after5": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(846cf7b2-f5e0-4031-b800-7fa5922cfb38)", "sensing_date": "2019-07-
30T04:30:47.575Z", "location": "MULTIPOLYGON (((24.448351 37.946922, 24.853334
39.574467, 21.897919      39.973557, 21.562479      38.347183, 24.448351
37.946922))))}, "images_before6": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(49cca110-5d7c-43a7-9f73-f887cf60cce3)", "sensing_date": "2019-07-
18T04:31:11.694Z", "location": "MULTIPOLYGON (((24.087444 36.453655, 24.483875
38.082607, 21.59144      38.482922, 21.25971      36.855423, 24.087444
36.453655))))}, "image_after6": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(e91c32c2-8203-45dc-bfcf-bf68e447d9ec)", "sensing_date": "2019-07-
30T04:31:12.392Z", "location": "MULTIPOLYGON (((24.087404 36.453331, 24.483946 38.082661,
21.591644      38.482971,      21.259819      36.855095,      24.087404
36.453331))))}, "images_before7": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(6680a026-c05a-4c9f-a6a8-85b6de21497a)", "sensing_date": "2019-07-
18T16:23:19.737Z", "location": "MULTIPOLYGON (((22.525024 36.348648, 25.371651
36.753792, 25.04195      38.373825, 22.130701      37.970161, 22.525024
36.348648))))}, "image_after7": {"link": "https://catalogue.onda-dias.eu/dias-
    
```

```
catalogue/Products(07a72faf-8bd9-4274-8a24-68cc0377a750)","sensing_date":"2019-07-30T16:23:20.378Z","location":"MULTIPOLYGON (((22.525043 36.348248, 25.371748 36.753422, 25.04196 38.373821, 22.130619 37.970123, 22.525043 36.348248)))"},"images_before8":{"link":"https://catalogue.onda-dias.eu/dias-catalogue/Products(62de7444-cf07-4785-9273-4a4b3cb4bba5)","sensing_date":"2019-07-18T16:23:44.554Z","location":"MULTIPOLYGON ((22.162409 37.841137,25.068329 38.244911,24.734728 39.865055,21.759275 39.462471,22.162409 37.841137)))"},"image_after8":{"link":"https://catalogue.onda-dias.eu/dias-catalogue/Products(ea117c4d-b7ec-43f8-8160-965e7af9ac1e)","sensing_date":"2019-07-30T16:23:45.203Z","location":"MULTIPOLYGON (((22.162298 37.841225, 25.06835 38.245033, 24.734789 39.864925, 21.759212 39.462307, 22.162298 37.841225)))"},"magnitude":{"value":"5.2"},"depth":{"value":"19.7"},"epicentral_location":{"latitude":{"value":"38.1076"},"longitude":{"value":"23.5393"}}, "timestamp":"2019-07-19T11:13:17.66Z"]}
```

In some cases, there might not be earthquake events present on a 50 kilometer range of the given city, such as using "earthquake located in Rome, Italy in 2021", as an input text. The response in this case is an empty array `[]`.

3.6 Logs and Outputs

The logs of the Semantic Framework, named "logfile.log", can be found on the same path as "docker-compose.yml". The file contains information related to the input of each request, parsing according to the expected keywords and the response. Response code and timestamp of the request are also included.

```
May 24, 2024 10:11:00 AM com.application.SemanticFramework.MyResource retrieval
INFO: [latitude]:8.75611 [longitude]125.7675 [year]:2024 [month]:null [day]:null [magnitude]:5.5 [city]:Bayugan [country]:Philippines
May 24, 2024 10:11:02 AM com.application.SemanticFramework.MyResource retrieval
INFO: [Response code]: 200, [Response]:
[{"year":"2024","city":"Bayugan","country":"Philippines","location_coordinates":"POINT (8.75611 125.7675)","images_before1":{"link":"https://colhub.copernicus.eu/dhus/odata/v1/Products('e92537a b-d2f5-42f7-b2bb-c7c483b13486')","sensing_date":"2024-02-02T21:24:16.705Z","location":"POLYGON ((126.804024 7.050664, 127.139153 8.680891, 124.885643 9.133541, 124.559784 7.507817, 126.804024 7.050664))","orbit_number":"163","pass_direction":"DESCENDING"},"image_after1":{"link":"https://colhub.copernicus.eu/dhus/odata/v1/Products('0e593b1a-b6be-4bb7-878a-f4d124cb88d4')","sensing_date":"2024-02-14T21:24:16.443Z","location":"POLYGON ((126.803497 7.050778, 127.138573 8.680758, 124.885063 9.133412, 124.55925 7.507935, 126.803497 7.050778))","orbit_number":"163","pass_direction":"DESCENDING"},"magnitude":{"value":"5.7"},"depth":{"value":"19.5"},"epicentral_location":{"latitude":{"value":"8.65547"},"longitude":{"value":"125.698"}}, "timestamp":"2024-02-10T03:22:07.16Z"]}
```

In cases where errors occur, the file contains information related to the error or exception that has happened.

```
Apr 04, 2022 7:25:47 AM com.application.SemanticFramework.MyResource retrieval
INFO: [Response code]: 200
[Response]: {
  "error": "Location not found. Please make sure to add a valid city and country name."
}

Apr 04, 2022 12:27:01 PM com.application.SemanticFramework.LocationDetection retrieveLongitude
SEVERE: Query error: HttpException: 500
```

3.7 Error messages

On this section, we focus on different error message scenarios supported by the SF.

In case that a user calls an invalid URL such as <http://<IP>:<PORT>/SemanticFramework/api/task> an error message is returned as a response in the form of:

```
{
  "error": "Unexpected parameter {task}. Was expecting one of 'population' or 'retrieve'."
}
```

In case the user tries to access the semantic framework directly through <http://<IP>:<PORT>/SemanticFramework/api/retrieve> and the given user credentials are not valid, the following error message is returned as a response:

```
{
  "error": "Invalid user credentials."
}
```

In the following cases, we suppose that the user uses the `/retrieve` endpoint, but the input text does not match the expected parameters (e.g. for the content of the `nlp` please see Section 5). In case that the user adds only `"I search for earthquakes"` on the input text, which in the target format is:

```
{
  'text': 'I search for earthquakes',
  'page': '1',
  'nlp':
    { 'event': 'earthquake', 'city': 9999941, 'country': 9999941, 'year': 9999932, 'month': 9999932,
      'day': 9999932, 'magnitude': 9999911, 'comparative': 9999952, 'point': False, 'latitude': 'null',
      'lognitude': 'null' }
}
```

the response is the following:

```
{
  "error": "No date reference in the query: year is mandatory (e.g. 2018)"
}
```

In case the user searches for "I search for earthquakes during 2018", which in the target format is:

```
{
  'text': 'I search for earthquakes during 2018',
  'page': '1',
  'nlp':
    {'event': 'earthquake', 'city': 9999941, 'country': 9999941, 'year': 2018, 'month': 'null', 'day':
    'null', 'magnitude': 9999911, 'comparative': 9999952, 'point': False, 'latitude': 'null', 'lognitude': 'null'}
}
```

the input text is again invalid for this reason:

```
{
  "error": "No event location reference in the query:country is mandatory (e.g. Italy)"
}
```

Similarly, in case the user adds also a city and inserts "I search for earthquakes during 2018 at Rome" as an input, which in the target format is:

```
{
  'text': 'I search for earthquakes during 2018 at Rome',
  'page': '1',
  'nlp':
    {'event': 'earthquake', 'city': 9999941, 'country': 9999941, 'year': 2018, 'month': 'null', 'day':
    'null', 'magnitude': 9999911, 'comparative': 9999952, 'point': False, 'latitude': 'null', 'lognitude': 'null'}
}
```

the output will be the same:

```
{
  "error": "No event location reference in the query:country is mandatory (e.g. Italy)"
}
```

In case that the user requests for "I search for earthquakes during 2018 at Rome, Italy", which in the target format is:

```
{
  'text': 'I search for earthquakes during 2018 at Rome, Italy',
  'page': '1',
  'nlp':
    {'event': 'earthquake', 'city': 'Rome', 'country': 'Italy', 'year': 2018, 'month': 'null', 'day': 'null',
    'magnitude': 9999911, 'comparative': 9999952, 'point': False, 'latitude': 'null', 'lognitude': 'null'}
}
```

Then, no error will be returned since all the mandatory keywords are present (however it will return an empty array as a response, since there is no information of that sort in the KB).

In addition, please notice some additional errors that have been added and correspond to error values that are returned by the NL mechanism (see also 5.2.1).

If the **magnitude** value in the parsed NL text is 9999912 then the response will be:

```
{
  "error": "The query contains more than one magnitude references"
}
```

Next, if the **magnitude** value in the parsed NL text is 9999913 then the response will be:

```
{
  "error": "Unexpected error related to magnitude input"
}
```

Lastly, if the **magnitude** value in the parsed NL text is 9999914 then the response will be:

```
{
  "error": "Magnitude format error: decimal separator should be a point (e.g. 6.2)"
}
```

For the **event** value in the parsed NL text, for the number 9999921 the error message will be:

```
{
  "error": "Invalid eventType. EventType should match the configuration file"
}
```

If the **event** value in the parsed NL text is 9999922 then the error message will be:

```
{  
  "error": "Invalid eventType. EventType should match the configuration file"  
}
```

For the **year** value in the parsed NL text, if that is 9999932 then the error message will be:

```
{  
  "error": "No date reference in the query: year is mandatory (e.g. 2018)"  
}
```

If the **year** value in the parsed NL text, if it is 9999933 then the error message will be:

```
{  
  "error": "The query contains more than one date references"  
}
```

Lastly, if the **year** value in the parsed NL text is 9999934 then the response will be:

```
{  
  "error": "Unexpected error related to date reference input"  
}
```

For the **city** value in the parsed NL text, if it is 9999941 then the error message will be:

```
{  
  "error": "Location not found. Please make sure to add a valid city and country name."  
}
```

If the **city** value in the parsed NL text is 9999942 then the error message will be:

```
{  
  "error": "The query contains more than one city or country"  
}
```

Also, if the **city** value in the parsed NL text is 9999943 then the response will be:

```
{  
  "error": "The query contains more than one country"  
}
```

```
}
```

Moreover, if the **city** value in the parsed NL text is 9999944 then the response will be:

```
{  
  "error": "Unexpected error related to country or city references"  
}
```

Moreover, if the city value in the parsed NL text is 9999945 then the response will be:

```
{  
  "error": "City could not be recognized"  
}
```

For the **comparative** value in the parsed NL text, if the number is 9999951 then the error message will be:

```
{  
  "error": "The query contains more than one comparative adjective (>, <, =, >=, <=)"  
}
```

If the **comparative** value in the parsed NL text is 9999953 then the error message will be:

```
{  
  "error": "The query contains more than symbol comparative adjective"  
}
```

Lastly, if the **comparative** value in the parsed NL text is 9999954 then the response will be:

```
{  
  "error": "Unexpected error related to the comparative adjective input"  
}
```

Due to the fact that Scihub and Apihub are not supported anymore as data source, we provided one new subcase in which, if the user gives in the configuration file the data sources Scihub or Apihub, the following error message will appear.

```
{  
  "error": "Scihub and Apihub are not supported as data sources anymore"  
}
```

4. Natural Language

On this section we focus on the features that are currently supported during the final release of the NL module.

4.1 Service Installation Requirements

The NL is a service that can be installed along with the SF, locally, by any authorized user using the installation instructions reported in section 2. Even though it consists of a separated service, it is considered as a whole with the rest of the SF. Thus, the separation between NL and SF is only for the reader to examine the two components, while also to be aware that those can be accessed separately if needed. However, the proper steps of this work, require from a user to access the NL service and through that acquire any relevant information from the SF. To interact with the service, at this release, a client should be generated (i.e. using POSTMAN) that executes "POST" requests over the specified URLs. The "Content-type" should be set to "application/json". The nl-api interacts with the sf-api, which interacts with the dr-api, which communicates with the data sources (i.e. INGV, Data Hub Service (DHuS), ONDA-DIAS), and with the KB, which is exposed using the docker-compose and returns the results that correspond to the criteria defined by the user.

4.2 Current Features

This version of the NL module contains a wide list of features that match the requirements reported in [AD-4]. Some of the most important are reported below:

- Identification of Country (e.g. Greece) and City (e.g. Athens) mentions.
- Identification of various Date formats (e.g. 8/6/2023).
- Identification of Magnitude numbers (e.g. 6.5).
- Identification of Comparative mentions (e.g. bigger than).
- Verification whether the input query refers to an earthquake event.
- Integration with the rest of the SF.
- Logs at the back-end to monitor the outputs of the module.
- Dockerization of the service and deployment using a docker-compose.
- Code publishing using GitHub and DockerHub.
- User access monitoring using Keycloak.

It is worth mentioning that all the libraries and tools used for the NL module, are open-source.

4.3 Natural Language Examples

In this section we provide a guideline with some examples for the users that want to access the Natural Language module.

4.4 Request for earthquake events using NL

After performing the initial step described in section 2.3, the users with "Search" user role are able to access the following URL in order to execute their queries:

http://<IP>:<PORT>/NaturalLanguage

using basic authentication on the authorization parameters and the credentials (username and password) that have been generated on section 2.2 - Step 3 for this user role.

As already mentioned, the request should be executed using POST type and Content-Type should be defined to "application/json". The request body in this case, should be a JSON having the following structure:

```
{
  "text": "I want an earthquake located in Athens, Greece during July 2019 with a magnitude value that is greater than 5.2"
}
```

In this case the parsed output (visible on the back-end if configured properly through the dedicated configuration file) of the NL module will be:

```
{
  "text": "I want an earthquake located in Athens, Greece during July 2019 with a magnitude value that is greater than 5.2",
  "page": "1",
  "nlp": {
    "event": "earthquake",
    "city": "Athens ",
    "country": "Greece ",
    "year": 2019,
    "month": "07",
    "day": "null",
    "magnitude": 5.2,
    "comparative": ">",
    "point": false,
    "latitude": "null",
    "longitude": "null"
  }
}
```

And the output demonstrated to the user is:

```
[{"year": "2019", "month": "07", "day": "19", "city": "Athens", "country": "Greece", "location_coordinates": "POINT (37.9755648 23.7348324)", "images_before1": {"link": "https://catalogue.onda-dias.eu/dias-catalogue/Products(04c87b0b-9b06-43ea-a375-dd36fa83f6fe)", "sensing_date": "2019-07-12T04:31:28.174Z", "location": "MULTIPOLYGON (((24.452286 37.946476, 24.857328 39.573891,
```

Collaborative Data Hub Software - Maintenance and Evolution Services - Ready for Digital Twin Earth

```

21.898474 39.973515, 21.563047 38.347275, 24.452286
37.946476)))"}, "image_after1": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(fb997d3c-0a3d-4d10-b9f2-46fb16cc982a)", "sensing_date": "2019-07-
24T04:31:29.01Z", "location": "MULTIPOLYGON (((24.452011 37.945801, 24.857298 39.574074,
21.898468 39.973724, 21.562834 38.346619, 24.452011
37.945801)))"}, "images_before2": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(50c4f69d-cc5a-4b13-ad78-440fd267231a)", "sensing_date": "2019-07-
12T04:31:52.995Z", "location": "MULTIPOLYGON (((24.091356 36.453266, 24.487803 38.081844,
21.592066 38.4827, 21.26038 36.855572, 24.091356
36.453266)))"}, "image_after2": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(ff23131f-b523-405e-bf74-cf36c255976a)", "sensing_date": "2019-07-
24T04:31:53.845Z", "location": "MULTIPOLYGON (((24.091064 36.452332, 24.487598 38.081154,
21.591896 38.48204, 21.260132 36.854664, 24.091064
36.452332)))"}, "images_before3": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(b40091ac-2e0a-4329-afd0-39bd82677d39)", "sensing_date": "2019-07-
12T16:22:38.578Z", "location": "MULTIPOLYGON (((22.49559 36.403755, 25.358112
36.81102, 25.028145 38.4314, 22.100489 38.025593, 22.49559
36.403755)))"}, "image_after3": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(97d02924-a820-4ce9-935a-bb4fb9f45c7d)", "sensing_date": "2019-07-
24T16:22:39.323Z", "location": "POLYGON ((22.09973 38.024979, 25.027424 38.430843, 25.357224
36.811459, 22.494625 36.404137, 22.09973
38.024979)))"}, "images_before4": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(90803615-417c-4e0a-b830-f50dc235a806)", "sensing_date": "2019-07-
12T16:23:03.41Z", "location": "MULTIPOLYGON (((22.131927 37.897034, 25.054323
38.302967, 24.720682 39.922222, 21.728252 39.51746, 22.131927
37.897034)))"}, "image_after4": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(951fd7f2-1527-4488-8f7a-9fc21fb853f6)", "sensing_date": "2019-07-
24T16:23:04.134Z", "location": "MULTIPOLYGON (((22.131233 37.896175, 25.053726
38.302174, 24.719849 39.922405, 21.727278 39.517582, 22.131233
37.896175)))"}, "images_before5": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(810f9352-7dc3-40c4-b8d6-ef9b94b37804)", "sensing_date": "2019-07-
18T04:30:46.871Z", "location": "MULTIPOLYGON (((24.448282 37.946869, 24.853338
39.57478, 21.89777 39.973873, 21.562277 38.347134, 24.448282
37.946869)))"}, "image_after5": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(846cf7b2-f5e0-4031-b800-7fa5922cfb38)", "sensing_date": "2019-07-
30T04:30:47.575Z", "location": "MULTIPOLYGON (((24.448351 37.946922, 24.853334
39.574467, 21.897919 39.973557, 21.562479 38.347183, 24.448351
37.946922)))"}, "images_before6": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(49cca110-5d7c-43a7-9f73-f887cf60cce3)", "sensing_date": "2019-07-
18T04:31:11.694Z", "location": "MULTIPOLYGON (((24.087444 36.453655, 24.483875
38.082607, 21.59144 38.482922, 21.25971 36.855423, 24.087444
36.453655)))"}, "image_after6": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(e91c32c2-8203-45dc-bfcf-bf68e447d9ec)", "sensing_date": "2019-07-
30T04:31:12.392Z", "location": "MULTIPOLYGON (((24.087404 36.453331, 24.483946 38.082661,
21.591644 38.482971, 21.259819 36.855095, 24.087404
36.453331)))"}, "images_before7": {"link": "https://catalogue.onda-dias.eu/dias-
catalogue/Products(6680a026-c05a-4c9f-a6a8-85b6de21497a)", "sensing_date": "2019-07-
18T16:23:19.737Z", "location": "MULTIPOLYGON (((22.525024 36.348648, 25.371651
36.753792, 25.04195 38.373825, 22.130701 37.970161, 22.525024
36.348648)))"}, "image_after7": {"link": "https://catalogue.onda-dias.eu/dias-

```

```
catalogue/Products(07a72faf-8bd9-4274-8a24-68cc0377a750)","sensing_date":"2019-07-30T16:23:20.378Z","location":"MULTIPOLYGON (((22.525043 36.348248, 25.371748 36.753422, 25.04196 38.373821, 22.130619 37.970123, 22.525043 36.348248))))","images_before8":{"link":"https://catalogue.onda-dias.eu/dias-catalogue/Products(62de7444-cf07-4785-9273-4a4b3cb4bba5)","sensing_date":"2019-07-18T16:23:44.554Z","location":"MULTIPOLYGON (((22.162409 37.841137,25.068329 38.244911,24.734728 39.865055,21.759275 39.462471,22.162409 37.841137))))","image_after8":{"link":"https://catalogue.onda-dias.eu/dias-catalogue/Products(ea117c4d-b7ec-43f8-8160-965e7af9ac1e)","sensing_date":"2019-07-30T16:23:45.203Z","location":"MULTIPOLYGON (((22.162298 37.841225, 25.06835 38.245033, 24.734789 39.864925, 21.759212 39.462307, 22.162298 37.841225))))"},"magnitude":{"value":"5.2"},"depth":{"value":"19.7"},"epicentral_location":{"latitude":{"value":"38.1076"},"longitude":{"value":"23.5393"},"timestamp":"2019-07-19T11:13:17.66Z"}}
```

Alternatively, the NL module can be accessed using a cURL request, such as the following:

```
curl --location 'http://localhost:8000/NaturalLanguage' \
  --header 'Content-Type: application/json' \
  --header 'Authorization: Basic c2VhcmNodXNlcjoxMjM0' \
  --data '{
    "text": " I want an earthquake located in Athens, Greece during July 2019 with a
    magnitude value that is greater than 5.2"
  }
,'
```

containing a valid set of user credentials and a request body following the aforementioned structure.

In addition, a similar query with missing values such as 'I want an earthquake located in Greece during 2019 with a certain magnitude', would result the parsed output (back-end):

```
{'text': "I search earthquake located in Greece in 2019", 'page': '1', 'nlp': {'event': 'earthquake', 'city': 9999945, 'country': 'Greece', 'year': 2019, 'month': 'null', 'day': 'null', 'magnitude': 9999911, 'comparative': 9999952, 'point': False, 'latitude': 'null', 'lognitude': 'null'}}
```

A relevant error message would be demonstrated to the user.

Note: To acquire a complete example, it is required to perform queries related information that already exist on the KB. In that sense, if there is no information or events for a target user query in the KB, the response to the user will be an empty array `[]`.

4.5 Logs and Outputs

The logs of the NL module are demonstrated directly on the back-end of the nl-api service. To activate or deactivate the NL module logs, a dedicated file exists in the directory "SF_service/config_nl/nl_logger_config.json". In that file, setting the value of 'nl_logger' to 'true' or

'false', will demonstrate or not demonstrate the NL module logs on the backend, accordingly. By default the logger is activated. In case it need to be deactivate, the content of the nl_logger_config.json file should be:

```
{
  "nl_logger": false
}
```

Having the log activated, that will demonstrate on the back-end the input query, parsed through the NL module, in a format that is being passed on the rest of the SF. An example of it may be:

```
{
  "text": "I want an earthquake located in Bayugan, Philippines during July 2024 with a magnitude value that is greater than 5.5",
  "page": "1",
  "nlp": {
    "event": "earthquake",
    "city": "Bayugan",
    "country": "Philippines ",
    "year": 2024,
    "month": 7,
    "day": "null",
    "magnitude": 5.5,
    "comparative": ">",
    "point": false,
    "latitude": "null",
    "lognitude": "null"
  }
}
```

In the above, the NL module has parsed the target 'text' and identified all the necessary parameters on the 'nlp' part, which are passed on the SF to perform a query based on them.

4.1 Error Messages

In cases where there is missing information on the input query, an error code will be given on the relevant field. Following are, all the supported error codes:

```
9999912 – The query contains more than one magnitude references
9999913 – Unexpected error related to magnitude input
9999914 – Magnitude format error: decimal separator should be a point (e.g. 6.2)
```

9999921 – The query does not refer to an earthquake event
9999922 – Unexpected error related to earthquake event relatedness
9999932 – No date reference in the query: year is mandatory (e.g. 2018)
9999933 – The query contains more than one date references
9999934 – Unexpected error related to date reference input
9999941 – No event location reference in the query: country is mandatory (e.g. Italy)
9999942 – The query contains more than one city or country
9999943 – The query contains more than one country
9999944 – Unexpected error related to country or city references
9999945 – No city reference
9999951 – The query contains more than one comparative adjective.
9999953 – The query contains more than one symbol comparative adjective
9999954 – Unexpected error related to the comparative adjective input

An example of missing values is:

```
{
  "text": "I search for earthquakes during 2024 at Bayugan",
  "page": "1",
  "nlp": {
    "event": "earthquake",
    "city": 9999941,
    "country": 9999941,
    "year": 2024,
    "month": "null",
    "day": "null",
    "magnitude": 9999911,
    "comparative": 9999952,
    "point": false,
    "latitude": "null",
    "lognitude": "null"
  }
}
```

The error codes can assist the system administrator, but mostly their functionality is to inform the semantic framework of what type of missing information/error exists on the input query.