GET-IT Documentation

Release 1.3

Anna Basoni, Stefano Menegon, Andrea Vianello, Alessandro Ogg

Jun 01, 2017

Contents

Documentation of 1.3 GET-IT version. Link to the old documentation pages: http://sp7.irea.cnr.it/wiki/index.php/Category:Learning

CHAPTER 1

MapQuest Maps Issue

From July 2016 the MapQuest Open is shutting down and moving to a new model which requires signing up and getting a 'key' to get access to map tiles.

In order to remove the dependence from the MapQuest Maps please follow these instructions:

mapquest_maps_issue

CHAPTER 2

Bing Maps Issue

From 6th October 2015 the Bing Maps API that are used by default in every GET-IT/GeoNode map are not working anymore.

The effect is that every map cannot be zoomed properly.

In order to remove the dependence from the Bing Maps please follow these instructions:

bing_maps_issue

CHAPTER 3

Presentation

Short presentation to GET-IT features and its architecture

GET-IT is a software package developed by SP7 researchers - working in different CNR Institutes - expert in Data infrastructure and geospatial services. The software was developed for RITMARE SP7 project. This documentation wants to help researchers to provide web services to share spatial and observation data following national and international standards. In particular the documentation is usefull for researcher team and/or institutes of RITMARE project that don't have an adequate data infrastructure.

The apllication was developed starting from open source packages with the addition of packages developed for project needs. It offers an interface for **data management, sharing, visualisation and** (optionaly) **download** through OGC standard web services for the following data categories:

1. **maps or layers** (spatial data). It permits to public and visualize spatial data (vector or raster). It use the web application GeoNode with customization for project needs. It provides the principal operations: upload data, data storing, styling and sharing data through standard services (WMS, WFS, WCS).
2. **observations** coming from different sensor types (buoys, glider, mooring, meteorological sensors, etc.). It uses an open-source solution based on Sensor Web Enablement (SWE) specifications, like implemented by 52° North (52N) project. It provides the operations to upload observation data, data storing and data management on a Data Base Management System, sharing observation on the web (through standard SOS interface).
3. **documents** (text files, spreadsheet, images, etc.).

One of the main function of GET-IT is the **metadata editing tool** named EDI that permits to create, edit and read metadata about uploaded data (spatial data, observation data and documents) to describe data in a national and international standard way (Repetorio Nazionale dei Dati Territoriali -RNDT-, INSPIRE and SensorML). Metadata (compiled by this tool) are improved in semantic to increase discovery operations.

GET-IT also permits to manage (for every data type uploaded) permission for registered users to:

- visualize
- edit
- download data

GET-IT is distributed with an open-source license (GPL v.3.0) and distributed through a ready to use virtual machine. After installation it provides a user friendly interface, the tools and services that compose a local infrastructure with above-mentioned features.

Activities by SP7 team

Here is a list of publications and other activities of the SP7 project team.

Publications on Journals

- C.Fugazza, A. Oggioni, P. Carrara, "RITMARE: Linked Open Data for Italian Marine Research", ERCIM-News (Issn: 0926-4981), No. 96 (January 2014), pp. 17-18, [Online] http://ercim-news.ercim.eu/en96/special/ritmare-linked-open-data-for-italian-marine-research
- F.Pavesi, A. Basoni, C. Fugazza, S. Menegon, A. Oggioni, M. Pepe, P. Tagliolato, P. Carrara, "EDI - A template-driven metadata editor for research data", JORS, 2016, in press.

International Conferences

- P.Carrara, A. Sarretta, A. Giorgetti, M. Ribera D'Alcalà, A. Oggioni, E. Partescano, "An interoperable infrastructure for the Italian Marine Research", IMDIS 2013 (Int. Conf. on Marine Data and Information Systems), ISSN 0006-6729, Lucca (Italia), September 23-25, 2013, pp. 186-190
- C.Fugazza, A. Basoni, S. Menegon, A. Oggioni, F. Pavesi, M. Pepe, A. Sarretta, P. Carrara, "RITMARE: Semantics-aware Harmonisation of Data in Italian Marine Research, Current Research Information Systems 2014 (CRIS2014)", Rome 13-15 May 2014, Proceedings published in Procedia Computer Science 33 (2014) 261 – 265, doi: 10.1016/j.procs.2014.06.041, available online at www.sciencedirect.com
- C.Fugazza, S. Menegon, A., Oggioni, F. Pavesi, M. Pepe, P. Carrara, "The RITMARE Starter Kit: Bottom-up capacity building for geospatial data providers", ICSOFT 2014 (9th International Conference on Software Paradigm Trends), Vienna 29-31 August 2014
- A.Basoni, M. Bastianini, C. Fugazza, S. Menegon, T. Minuzzo, A. Oggioni, F. Pavesi,M. Pepe, A. Sarretta, P. Tagliolato and P. Carrara, "Fostering bottom-up capacity in managing and sharing marine observations: the RITMARE StarterKit", accepted at EuroGOOS 2014, Lisbon 28-30 October 2014 (poster session)
- C.Fugazza et al., "Sensor metadata blueprints and computer-aided editing for disciplined SensorML", accepted at Digital Earth 2015 / 9th ISDE Symposium in Halifax, NS, Canada
- M.Bastianini, F. Bernardi Aubry, F. Bianchi, A. Boldrin, E. Camatti, P. Carrara, A. Delazzari, S. Guerzoni, S. Menegon, A. Oggioni, A. Pugnetti, A. Sarretta, G. Socal, P. Tagliolato, A. Vianello, "The LTER site Gulf of Venice and the project RITMARE: a case study for the recovery, search, view and sharing of long term ecological marine research data", submitted at XXII AIOL Congress
- A.Oggioni, A. Basoni, M. Bastianini, C. Fugazza, S. Menegon, F. Pavesi, M. Pepe, A. Sarretta, P. Carrara, "Lo StarterKit RITMARE: uno strumento abilitante per la costruzione bottom-up di un'infrastruttura di dati marini". 18a Conf. Naz. ASITA, Firenze 14-16 ottobre 2014, pp. 921-922, ISBN 978-88-903132-9-5
- A.Oggioni, P. Tagliolato, C. Fugazza, I. Rosati, L. Criscuolo, P. Carrara, "On the opportunity of exploiting open Geospatial standard for biotic data management," in 13th European Ecological Federation (EEF), 2015.
- M.Pepe, A. Basoni, M. Bastianini, C. Fugazza, S. Menegon, A. Oggioni, F. Pavesi, A. Sarretta, P. Carrara, "Bottom-up capacity building for data providers in RITMARE", Geophysical Research Abstracts, Vol. 16, EGU2014-10536, 2014, EGU General Assembly 28-30 April 2014, ESSI1.2, 30 April 2014

- C.Fugazza, M. Pepe, A. Oggioni, P. Tagliolato, P. Carrara, 2016, "Streamlining geospatial metadata in the Semantic Web". IOP Conference Series: Earth and Environmental Science, 34(1), 12009. doi:10.1088/1755-1315/34/1/012009
- P.Tagliolato, A. Oggioni, C. Fugazza, M. Pepe, P. Carrara, (2016), "Sensor metadata blueprints and computer-aided editing for disciplined SensorML". IOP Conference Series: Earth and Environmental Science, 34(1), 12036. doi:10.1088/1755-1315/34/1/012036
- P.Tagliolato, A. Oggioni, C. Fugazza, M. Pepe and P. Cararra, "Supporting provision of sensor metadata through multi-tenanted management of SensorML documents", INSPIRE Conference 2016, Barcelona, 26-30 September 2016
- C.Fugazza, M. Pepe, A. Oggioni, P. Tagliolato and P. Carrara, "Streamlining INSPIRE metadata for the Semantic Web", INSPIRE Conference 2016, Barcelona, 26-30 September 2016.

Books

- A.Oggioni, P. Tagliolato, C. Fugazza, M. Pepe, S. Menegon, F. Pavesi, and P. Carrara, "Interoperability in marine sensor networks through SWE services," in Oceanographic and Marine Cross-Domain Data Management for Sustainable Development, P. Diviacco, A. Leadbetter, and H. Glaves, Eds. Hershey: IGI Global, 2017, p. in press.

IPR Protection of the suite software GET-IT

Action to protect IPR (Intellectual Property Rights) of the developers of the suite software GET-IT Starter Kit within the Italian flagship project RITMARE.

A trademark application request has been submitted on May 2015 to the Italian Patent and Trademark Office (UIB) to protect the Intellectual Property Right of the suite software developed during the RITMARE project, enabling researchers to create their services to disseminate observed and geographical data (and their metadata) in spatial data infrastructures compliant with OGC standards. The name registred is: GET-IT Geoinformation Enabling Toolkit Starter Kit.

Short presentation to GET-IT features and its architecture

Activities by SP7 team

CHAPTER 4

Tutorials

Tutorials

Users Tutorials

Welcome to the Get-It Users Tutorial documentation.

This tutorial will teach how to use the Get-it going in depth into what we can do with software application. At the end of this section you will you will know how to:

1. Manage users accounts and how to modify them.
2. Use and manage the different Get-It basic resouces.
3. Manage Sensors and Observations and pubblish them.
4. Manage Layers and Views (Maps) and publish them.
5. Manage Metadata and pubblish them.

Prerequisites

Before proceeding with the reading, it is strongly recommended to be sure having clear the following concepts:

1. GeoNode basic concepts.
2. What is a geospatial server and a basic knowledge of the geospatial web services.
3. What is Open Geospatial Consortium (OGC) and it's standards
4. What is a metadata and a catalog.
5. What is a map and a legend.
6. What is a sensor and a observation (Sensor Observation System, OGC standard).

Accounts and users

This section will guide you through account registration, updating your account information, and viewing other user accounts.

Learn how to register and manage an account.

Managing layers

Layers are a published resource representing a raster or vector spatial data source. Layers also can be associated with metadata, ratings, and comments.

In this section, you will learn how to create a new layer by uploading a local data set, add layer info, change the style of the layer, and share the results.

Managing layers

Learn how to manage layers.

Managing metadata

Learn how to manage metadata.

Managing observations

An Observation is an action whose result is an estimate of the value of some property of the feature-of-interest (i.e. station, animals or tissue), at a specific point in time, obtained using a specified procedure or sensor (After Cox 2008 cited by INSPIRE Cross Thematic Working Group on Observations & Measurements, 2011). An observation can be associated with a specific sensor that have collected the observation, it may be associated with metadata.

In this section, you will learn how to add new sensor info (metadata of the sensors), to upload new observations, and share these through specific web services: Sensor Observation Services (SOS).

In the GET-IT home page Sensor section is dedicated to managing observation and sensors, the figure below show the different interaction parts for managing observations and sensors.

Two tabs, **Explore SOS** and **Upload Observations**, are dedicated respectively to:

- show the list of the sensors within local repository and a link with sensor details in order to improve knowledge, provided in XML and in HTML (clicking on Sensor Details green button). More information in *How create Sensors Metadata* or *How check Sensors Metadata*;
- upload observations by friendly user interface. Mere information in *How upload Observations*

Two buttons, in the header, provides to register new sensor and get capabilities of the local Sensors Observations Services (SOS).

All the functionality making SWE pure requests.

How create Sensors Metadata

In order to discover, integrate, and exploit sensor data (and also to presserve them for future needs), we need to accurately record sensor information.

The Sensor Model Language (SensorML) has been adopted by OGC for developing the SWE framework.

SensorML provides a definition language that supports all details of sensors and sensor-to-platform constellations.

Sensor meta-data creation has been performed by GET-IT metadata editor, called EDI, which allows ease and friendly instrument registration (SensorML editing version v1.0 and v2.0) through graphical user interfaces (GUI) and auto completion facilities linked to vocabularies. Some sections of SensorML have been borrowed from the terms present in RDF, in order to harmonize and semantically enrich the metadata. In particular for the SensorML templates implemented in the EDI:

- Parameters: P01 and P02 NERC vocabularies or EnvThes
- Units of measure: P06 NERC vocabulary or EnvThes
- Sensor types: P07 NERC vocabulary
- Manufacturers: FOAF (Friends Of A Friend) graph version of Esonet Yellow Pages.
- Operators, and owners: FOAF (Friends Of A Friend) graph. In this case the graph changed independently from the project to which it refers.

Within EDI template, implemented in GET_IT, we adopted the sensor profile suggested in the OGC document titled "OGC® Best Practice for Sensor Web Enablement Lightweight SOS Profile for Stationary In-Situ Sensors. To be compliant to this profile every sensors will be modeled as a PhysicalSystem and the following metadata items are mandatory:

- gml:description - short textual description of the sensor or sensor system;
- gml:identifier - unique identifier of the sensor system.
- sml:keywords - terms which help to describe the sensor system and serve for discovery purposes;
- sml:identification - this element contains identifiers of the sensor system;
- sml:classification - this element contains classifiers for the sensor system;
- sml:contacts - this element contains contact information about the operator of the sensor;
- sml:featuresOfInterest - this element contains the real world entity, the feature of interest, which is observed by the sensor system. In case of this profile, the feature of interest is a station and modelled as a SamplingPoint;
- sml:outputs - the outputs of the sensors attached to the sensor system.

Within EDI user interface items in bold are the mandatory ones.

Before insert any observations is necessary to create SensorML by EDI template.

The figure below show the EDI user interface for OGC SensorML v2.0 Lightweight SOS Profile for Stationary In-Situ Sensors.

The video allows to have a idea about the interaction between user and EDI user interface.

How check Sensors Metadata

How upload Observations

For other options to insert real time (RT) or near real time (NRT) observations, within GET-IT, should be read the part of documentation about that *Examples of NRT or RT insert observations*.

Managing views

Accounts and users This section will guide you through account registration, updating your account information, and viewing other user accounts.

Managing layers Create, delete, manage and share Layers and Metadata.

Managing observations Manage observations, create and insert sensors.

Managing views Create, delete, manage and share Maps. Add, view and download observation.

compilazionemetadati you will learn how to compile metadata.

editing_metadati you will learn how to compile xxxxxx.

conversione_geotiff how to create a correct geotiff format file.

file_prj description of what is a prj file.

servizi_jrc JRC web services.

external_services external web services usefull for RITMARE project.

sistemi_riferimento principal reference system used in Italy.

utilizzo_wms you will learn how to use wms services.

vincoli_dati you will learn what "data restriction" are.

plugin_qgis a usefull plugin of QGIS for mangare GET-IT layers.

Administrators Tutorials

Welcome to the GET-IT Administrators Workshop!

GET-IT quick installation Learn how to install GET-IT by virtual machine.

Installing GET-IT from scratch Learn how to install GET-IT from scratch step by step.

Server requirements Minimum server requirements.

Advanced administration Advanced administration

Security issues Customize important setting for security.

Choosing the "right" Domain Name Learn how to choose and change the domain name of your server.

bing_maps_issue Bing Maps Issue

GET-IT quick installation

Before starting installation read the *Server requirements* page.

GET-IT is distributed as a virtual appliance in "vmdk"(virtual machine disk) format.

A video demo is available HERE (you can change the settings in YouTube to improve the resolution of the video).

Download and installation of the Virtual Machine

Download the file ''starterkit.vmdk.gz" using the following LINK , we suggest to download the file in the machine where the virtualizer is present.

Normally you can use the "wget" command followed by the url:

```
wget http://geosk.ve.ismar.cnr.it/static/vm-sk-1.2-10-5345344/starterkit-disk1.vmdk
```

Extract the file with "gunzip" command:

```
gunzip starterkit.vmdk.gz
```

Domain Name

To let GET-IT work properly you need to use a right domain name, See the page *Choosing the "right" Domain Name* to learn how to change it.

Security

It's important for your server security follow instruction on the page: *Security issues*

Update GET-IT

At the end of the installation it's important to update your GET-IT to the last version available following the steps described in *Keep GET-IT up-to-date*.

Installing GET-IT from scratch

Before starting installation read the *Server requirements* page.

If you just want to install GET-IT, it is recommended to use Ubuntu 12.04.

Prerequisites:

1. GeoNode (version 2.0.x)
2. 52°North SOS (version 4.0.0 or 4.x)

GeoNode Installation

Install geonode from PPA in Ubuntu 12.04:

```
$ sudo apt-get install python-software-properties

$ sudo add-apt-repository ppa:geonode/release

$ sudo apt-get update

$ sudo apt-get install geonode
```

Setup the IP address and create a superuser:

$ sudo geonode-updateip 127.0.0.1

$ geonode createsuperuser

52°North SOS Installation

Follow the instructions (for SOS version 4.x) at https://wiki.52north.org/bin/view/SensorWeb/SensorObservationServiceIVDocumentation#Installation

PostgreSQL and PostGIS (currently at version 9.1 and 2.0) are already installed by default in GeoNode. For this reason it is not necessary to make a new installation. An update of the current version of Java it is necessary, 52°North requires Java runtime environment (JRE) 7.0 or higher:

$ java -version

$ sudo add-apt-repository ppa:webupd8team/java

$ sudo apt-get update

$ sudo apt-get install oracle-java7-installer

$ java -version

Just a note: GET-IT expects to find a webapp named "observations". So, before "Install and configure Tomcat", rename the 52n-sos-webapp.war into observations.war.

After the deploy of war file, in the http://127.0.0.1:8080/observations you can check the installation of the SOS admin, create a new PostgreSQL database using the PostGIS template created during the PostGIS installation.

$ sudo su - postgres

$ psql -U <user>

$ CREATE DATABASE <db_name>;

$ connect <db_name>;

$ CREATE EXTENSION postgis;

If return error like *could not open extension control file "/usr/share/postgresql/9.1/extension/postgis.control": No such file or directory*, please exit to the PostgreSQL DB and to the user, execute:

$ sudo apt-get install postgis postgresql-9.X-postgis-scripts

after retry the commands above. Check extensions:

$ dx

Navigate to the http://127.0.0.1:8080/observations and complete the installation process. Follow the steps on the screen to configure your SOS instance (you don't have to do this if you've build the preconfigured webapp). More information about the settings can be found in the Configure/Administrate the SOS section of this page. Before to start copy user (DATABASE_USER) and password (DATABASE_PASSWORD) of the PostgreSQL:

$ nano /etc/geonode/local_settings.py

Change the 52 North SOS Configuration (http://127.0.0.1:8080/observations/admin/settings (Transactional Security) by setting the "Transactional Authorization Token"

Complete the installation follow the steps and finally set the user and password.

GET-IT Installation

Install SK from archive file

$ sudo pip install starterkit

Edit the starterkit local_settings.py by setting the SITEURL, SITENAME, TRANSACTIONAL_AUTHORIZATION_TOKEN (
$ nano /etc/starterkit/local_settings.py

Edit the file /etc/apache2/sites-available/geonode and change the following directive from:

WSGIScriptAlias / /var/www/geonode/wsgi/geonode.wsgi

to:

WSGIScriptAlias / /usr/local/lib/python2.7/dist-packages/geosk/wsgi.py # path to geosk/wsgi.py

Restart apache:

```
$ sudo service apache2 restart
```

Edit the templates in geosk/templates, the css and images to match your needs.

Syncdb and collectstatic:

```
$ sk syncdb

$ sk collectstatic
```

Domain Name

To let GET-IT work properly you need to use a right domain name, See the page *Choosing the "right" Domain Name* to learn how to change it.

Security

It's important for your server security follow instruction on the page: *Security issues*

Server requirements

GET-IT is distributed by VMware virtual machine with open license (GPL v3.0) and can be installed on any server (Linux, Window, MAC) with VMware client installed.

For specific necessary we can provide different standard virtual machine format.

GET-IT doesn't need a very efficient server, we council the server feature according our experience (not mandatory):

```
* RAM: 6 GB minimum, better 8 GB
* CPU number:  1CPU  4 core
* HDisk capacity: virtual machine occupy 30 GB
* hardware 64-bit is recommended
```

This features may be upgraded to improve performance or for big data ammount in the portal.

Advanced administration

Keep your system up-to-date

It's very important to maintain the system updated for security and stability.

To update your system in command line, run:

```
$ sudo aptitude update
$ sudo aptitude full-upgrade
```

Alternatively, you may use an interactive text interface:

```
$ sudo aptitude
```

Occasionally, you should remove the old kernel versions. Here an automatic script to cleanup (use with caution):

```
echo $(dpkg --list | grep linux-image | awk '{ print $2 }' | sort -V | sed -n '/
↪'`uname -r`'/q;p') $(dpkg --list | grep linux-headers | awk '{ print $2 }' | sort -
↪V | sed -n '/'"$(uname -r | sed "s/\([0-9.-]*\)-\([^0-9]\+\)/\1/")"'/q;p') | xargs␣
↪sudo apt-get -y purge
```

Keep GET-IT up-to-date

Starting from **1.2** version, updating GET-IT it's possible in a easy way. The administrator need run a command to update the software to the last version available. The command to run with administrator permission is:

```
sudo pip install --upgrade --no-deps starterkit
sudo pip install django-analytical==1.0.0 owslib==0.10.3
sudo sk  collectstatic --noinput -i externals -i node_modules -i SOSClient
sudo sk migrate mdtools
sudo /etc/init.d/apache2 reload
```

Unfortunately **this update command is not possible if you have a GET-IT version precedent the 1.2a4** (only first two release), in this case you will need to contact your tutor who will provide to contact the GET-IT developement team to update your system. Later you will be able to run the update command by yourself.

Oracle JDK (v. 6) Installation

GeoServer's speed depends a lot on the chosen Java Runtime Environment (JRE). For best performance, use Oracle JRE 6 (also known as JRE 1.6) or newer (http://docs.geoserver.org/2.4.x/en/user/production/java.html).

Installation steps on Ubuntu 12.04 LTS:

```
sudo apt-get install python-software-properties
sudo add-apt-repository ppa:webupd8team/java
sudo apt-get update
sudo apt-get install oracle-java6-installer
```

Install native JAI and JAI Image I/O extensions

In order to improve the performance, install the native JAI version in your JDK/JDE (see http://docs.geoserver.org/2.4.x/en/user/production/java.html).

Unfortunately, at time of writing the Oracle package is not available so here an alternative installation procedure Ubuntu 12.04 LTS (with Oracle JRE 6):

```
sudo apt-get install libjai-core-java libjai-imageio-core-java
sudo cp /usr/lib/jni/{libclib_jiio.so,libmlib_jai.so} /usr/lib/jvm/java-6-oracle/jre/
↪lib/amd64/
sudo cp /usr/share/java/{jai_core-1.1.4.jar,jai_codec-1.1.4.jar,jai_imageio-1.2.jar,
↪mlibwrapper_jai-1.1.4.jar,clibwrapper_jiio-1.2.jar} /usr/share/geoserver/WEB-INF/
↪lib/
sudo /etc/init.d/tomcat7 restart
```

Security issues

Please customize these settings:

Changing DB password of “geonode” user

:: $ sudo -u postgres psql –command ‘password geonode’

Edit the configuration file to reflect the changes /etc/starterkit/local_settings.py * DATABASE_PASSWORD

Changing 52 North SOS Transactional Security Token

http://demo1.get-it.it/observations/admin/datasource/settings Edit the configuration file to reflect the changes /etc/starterkit/local_settings.py * TRANSACTIONAL_AUTHORIZATION_TOKEN

Keeping cryptography secure

/etc/starterkit/local_settings.py * SECRET_KEY

Choosing the “right” Domain Name

To let GET-IT work, you cannot use the IP address in configuration files, you need to set a domain name.

- Keep “OGC Endpoints” persistent and alive (for remote users and cascading services)
- Stable URIs/URLs as semantic web “best practice”
- GET-IT uses Domain Name as default prefix for different resources URIs (i.e. Sensors, Offerings)

Updating Domain Name

To update the domain name of your server use the following command:

```
$ sudo sk-updateip [domain name] # utility for assign a domain name to the GET-IT
```

Example: $ sudo sk-updateip demo1.get-it.it

Data Flow Engine Tutorials

Welcome to the GET-IT Administrators Workshop!

Examples of NRT or RT insert observations ...

Examples of NRT or RT insert observations

The fastest way for inserting new observations within GET-IT is to use the graphical user interface (GUI) (*How upload Observations*) implemented ad-hoc for this purpose. This GUI is specifically developed for users who want to insert data that were prevously collected within spreadsheets or similar files. A different scenario is the automatic upload from data logger, remote station, etc.: In this case the correct way to insert observation data in GET-IT is directly through the exposed SOS interface, exploiting standard transactional requests of Sensor Web (SWE) and SOS. In particular the SOS accept the insertObservation request to insert standard encoded observations. The use of XML language ensures a complete OGC-SWE compliance. Below an example of insertObservation request:

```
<sos:InsertObservation service="SOS" version="2.0.0"
    xmlns:sos="http://www.opengis.net/sos/2.0"
    xmlns:swes="http://www.opengis.net/swes/2.0"
    xmlns:swe="http://www.opengis.net/swe/2.0"
    xmlns:sml="http://www.opengis.net/sensorML/1.0.1"
    xmlns:gml="http://www.opengis.net/gml/3.2"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:om="http://www.opengis.net/om/2.0"
    xmlns:sams="http://www.opengis.net/samplingSpatial/2.0"
    xmlns:sf="http://www.opengis.net/sampling/2.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.opengis.net/sos/2.0 http://schemas.opengis.net/sos/
↪2.0/sos.xsd http://www.opengis.net/samplingSpatial/2.0 http://schemas.opengis.net/
↪samplingSpatial/2.0/spatialSamplingFeature.xsd">

    <sos:offering>offering:https://www.website.org/procedure/procedureType_B/
↪observations</sos:offering>

    <!-- multiple offerings are possible -->
    <!-- Start of OM_Observation o1 -->
    <sos:observation>
        <om:OM_Observation gml:id="o1">
            <om:type xlink:href="http://www.opengis.net/def/observationType/OGC-OM/2.
↪0/OM_Measurement"/>

            <om:phenomenonTime>
                <gml:TimeInstant gml:id="phenomenonTime_o1">
                    <gml:timePosition>2015-08-27T12:32:16.000+02:00</gml:timePosition>
                </gml:TimeInstant>
            </om:phenomenonTime>
```

```
            <om:resultTime xlink:href="#phenomenonTime_o1"/>

            <om:procedure xlink:href="https://www.website.org/procedure/procedureType_
↪B" xlink:arcrole="http://www.website.org/2.0/sensors"/>

            <om:observedProperty xlink:href="http://vocab.nerc.ac.uk/collection/P02/
↪current/ASLV/"/>

            <om:featureOfInterest>
                <sams:SF_SpatialSamplingFeature gml:id="sample_o1">
                    <gml:identifier codeSpace="">http://www.website.org/sensors/SSF/
↪SSF_b</gml:identifier>
                    <gml:name>Lago Paione inferiore</gml:name>
                    <sf:type xlink:href="http://www.opengis.net/def/
↪samplingFeatureType/OGC-OM/2.0/SF_SamplingPoint"/>
                    <sf:sampledFeature
                        xlink:title="Lago Paione inferiore"/>
                    <sams:shape>
                        <gml:Point gml:id="p1">
                            <gml:pos srsName="http://www.opengis.net/def/crs/EPSG/0/
↪4326">46.168979 8.18923</gml:pos>
                        </gml:Point>
                    </sams:shape>
                </sams:SF_SpatialSamplingFeature>
            </om:featureOfInterest>

            <om:result xsi:type="gml:MeasureType" uom="m">0.28</om:result>

        </om:OM_Observation>
    </sos:observation>
    <!-- End of OM_Observation o1 -->

    <!-- Start of OM_Observation o2 -->
    <sos:observation>
        <om:OM_Observation gml:id="o2">
            <om:type xlink:href="http://www.opengis.net/def/observationType/OGC-OM/2.
↪0/OM_Measurement"/>
            <om:phenomenonTime xlink:href="#phenomenonTime_o1"/>
            <om:resultTime xlink:href="#phenomenonTime_o1"/>
            <om:procedure xlink:href="https://www.website.org/procedure/procedureType_
↪B" xlink:arcrole="http://www.website.org/2.0/sensors"/>
            <om:observedProperty xlink:href="http://vocab.nerc.ac.uk/collection/P02/
↪current/ASLV/"/>
            <om:featureOfInterest xlink:href="#sample_o1"/>
            <om:result xsi:type="xs:integer">0.12</om:result>
        </om:OM_Observation>
    </sos:observation>
    <!-- End of OM_Observation o2 -->

    <!-- Start of OM_Observation o3 -->
    <sos:observation>
        <om:OM_Observation gml:id="o3">
            <om:type xlink:href="http://www.opengis.net/def/observationType/OGC-OM/2.
↪0/OM_Measurement"/>
            <om:phenomenonTime xlink:href="#phenomenonTime_o1"/>
            <om:resultTime xlink:href="#phenomenonTime_o1"/>
            <om:procedure xlink:href="https://www.website.org/procedure/procedureType_
↪B" xlink:arcrole="http://www.website.org/2.0/sensors"/>
```

```
            <om:observedProperty xlink:href="http://vocab.nerc.ac.uk/collection/P02/
↪current/ASLV/"/>
            <om:featureOfInterest xlink:href="#sample_o1"/>

            <!-- example of data array -->
            <om:result xsi:type="swe:DataArrayPropertyType">
            <swe:DataArray>
                <swe:elementCount>
                    <swe:Count>
                        <swe:value>15</swe:value>
                    </swe:Count>
                </swe:elementCount>
                <swe:elementType name="defs">
                    <swe:DataRecord>
                        <swe:field name="phenomenonTime">
                            <swe:Time definition="http://www.opengis.net/def/property/
↪OGC/0/PhenomenonTime">
                                <swe:uom xlink:href="http://www.opengis.net/def/uom/
↪ISO-8601/0/Gregorian"/>
                            </swe:Time>
                        </swe:field>
                        <swe:field name="ASLV">
                            <swe:Quantity definition="http://vocab.nerc.ac.uk/
↪collection/P02/current/ASLV/">
                                <swe:uom code="m"/>
                            </swe:Quantity>
                        </swe:field>
                    </swe:DataRecord>
                </swe:elementType>
                <swe:encoding>
                    <swe:TextEncoding tokenSeparator="#" blockSeparator="@"/>
                </swe:encoding>
                <swe:values>2012-11-19T13:30:00+02:00#0.15@2012-11-19T13:31:00+02:00
↪#0.15@2012-11-19T13:32:00+02:00#0.85@2012-11-19T13:33:00+02:00#0.5@2012-11-
↪19T13:34:00+02:00#0.9@2012-11-19T13:35:00+02:00#0.7@2012-11-19T13:36:00+02:00#0.
↪5@2012-11-19T13:37:00+02:00#0.6@2012-11-19T13:38:00+02:00#0.5@2012-11-
↪19T13:39:00+02:00#0.4@2012-11-19T13:40:00+02:00#0.34@2012-11-19T13:41:00+02:00#0.
↪25@2012-11-19T13:42:00+02:00#0.79@2012-11-19T13:43:00+02:00#0.56@2012-11-
↪19T13:44:00+02:00#0.25</swe:values>
            </swe:DataArray>
        </om:result>
        </om:OM_Observation>
    </sos:observation>
    <!-- End of OM_Observation o3 -->
</sos:InsertObservation>
```

It is possible to further reduce the XML relying insertResultTemplate and insertResult, below the example of both requests.

insertResultTemplate

```
<?xml version="1.0" encoding="UTF-8"?>
<sos:InsertResultTemplate service="SOS" version="2.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:swes="http://www.opengis.net/swes/2.0"
    xmlns:sos="http://www.opengis.net/sos/2.0"
    xmlns:swe="http://www.opengis.net/swe/2.0"
    xmlns:sml="http://www.opengis.net/sensorML/1.0.1"
```

```
    xmlns:gml="http://www.opengis.net/gml/3.2"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:om="http://www.opengis.net/om/2.0"
    xmlns:sams="http://www.opengis.net/samplingSpatial/2.0"
    xmlns:sf="http://www.opengis.net/sampling/2.0"
    xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:schemaLocation="http://www.
↪opengis.net/sos/2.0 http://schemas.opengis.net/sos/2.0/sosInsertResultTemplate.xsd␣
↪http://www.opengis.net/om/2.0 http://schemas.opengis.net/om/2.0/observation.xsd ␣
↪http://www.opengis.net/samplingSpatial/2.0 http://schemas.opengis.net/
↪samplingSpatial/2.0/spatialSamplingFeature.xsd">
    <sos:proposedTemplate>
        <sos:ResultTemplate>
            <swes:identifier>http://www.website.org/test/procedure/procedureType_B/
↪template/1</swes:identifier>

            <sos:offering>offering:https://www.website.org/procedure/procedureType_B/
↪observations</sos:offering>

            <sos:observationTemplate>
                <om:OM_Observation gml:id="sensor2obsTemplate">
                    <om:type xlink:href="http://www.opengis.net/def/observationType/
↪OGC-OM/2.0/OM_Measurement"/>
                    <om:phenomenonTime nilReason="template"/>
                    <om:resultTime nilReason="template"/>
                    <om:procedure xlink:href="https://www.website.org/procedure/
↪procedureType_B"/>
                    <om:observedProperty xlink:href="http://vocab.nerc.ac.uk/
↪collection/P02/current/ASLV/"/>
                    <om:featureOfInterest>
                        <sams:SF_SpatialSamplingFeature gml:id="sample_o1">
                            <gml:identifier codeSpace="">http://www.website.org/
↪sensors/SSF/SSF_b</gml:identifier>
                            <gml:name>Lago Paione inferiore</gml:name>
                            <sf:type xlink:href="http://www.opengis.net/def/
↪samplingFeatureType/OGC-OM/2.0/SF_SamplingPoint"/>
                            <sf:sampledFeature
                                xlink:title="Lago Paione inferiore"/>
                            <sams:shape>
                                <gml:Point gml:id="p1">
                                    <gml:pos srsName="http://www.opengis.net/def/crs/
↪EPSG/0/4326">46.168979 8.18923</gml:pos>
                                </gml:Point>
                            </sams:shape>
                        </sams:SF_SpatialSamplingFeature>
                    </om:featureOfInterest>
                    <om:result/>
                </om:OM_Observation>
            </sos:observationTemplate>

            <sos:resultStructure>
                <swe:DataRecord>
                    <swe:field name="phenomenonTime">
                        <swe:Time definition="http://www.opengis.net/def/property/OGC/
↪0/PhenomenonTime">
                            <swe:uom xlink:href="http://www.opengis.net/def/uom/ISO-
↪8601/0/Gregorian"/>
                        </swe:Time>
                    </swe:field>
```

```
                    <swe:field name="ASLV">
                        <swe:Quantity definition="http://vocab.nerc.ac.uk/collection/
↪P02/current/ASLV/">
                            <swe:uom code="m"/>
                        </swe:Quantity>
                    </swe:field>
                </swe:DataRecord>
            </sos:resultStructure>

            <sos:resultEncoding>
                <swe:TextEncoding tokenSeparator="#" blockSeparator="@"/>
            </sos:resultEncoding>

        </sos:ResultTemplate>
    </sos:proposedTemplate>
</sos:InsertResultTemplate>
```

and insertResult

```
<?xml version="1.0" encoding="UTF-8"?>
<sos:InsertResult service="SOS" version="2.0.0"
    xmlns:sos="http://www.opengis.net/sos/2.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
↪www.opengis.net/sos/2.0 http://schemas.opengis.net/sos/2.0/sos.xsd">

    <sos:template>http://www.website.org/test/procedure/procedureType_B/template/1</
↪sos:template>

    <swe:values>2012-11-19T13:30:00+02:00#0.15@2012-11-19T13:31:00+02:00#0.15@2012-11-
↪19T13:32:00+02:00#0.85@2012-11-19T13:33:00+02:00#0.5@2012-11-19T13:34:00+02:00#0.
↪9@2012-11-19T13:35:00+02:00#0.7@2012-11-19T13:36:00+02:00#0.5@2012-11-
↪19T13:37:00+02:00#0.6@2012-11-19T13:38:00+02:00#0.5@2012-11-19T13:39:00+02:00#0.
↪4@2012-11-19T13:40:00+02:00#0.34@2012-11-19T13:41:00+02:00#0.25@2012-11-
↪19T13:42:00+02:00#0.79@2012-11-19T13:43:00+02:00#0.56@2012-11-19T13:44:00+02:00#0.25
↪</swe:values>
</sos:InsertResult>
```

Otherwise the SOS server integrated within GET-IT allows to use JSON for sending requests. In OGC SWE, the JSON "binding" is not yet standardized, but this feature can ease the work of programmers familiar with JSON. Below an example of insertObservations JSON:

```
{
  'observation': {
      'featureOfInterest': {
          'geometry': {
              'coordinates': [45.43,
                              12.33],
              'crs': {
                  'properties': {
                      'name': 'EPSG:4326'
                  },
                  'type': 'name'
              },
              'type': 'Point'
          },
          'identifier': {
              'codespace': 'http://www.opengis.net/def/nil/OGC/0/unknown',
              'value': u'http://sp7.irea.cnr.it/featureOfInterest/
↪PuntaSaluteCanaleGiudecca'
```

```
          },
          'name': [{
              'codespace': 'http://www.opengis.net/def/nil/OGC/0/unknown',
              'value': u'Laguna di Venezia - Punta Salute Canale della Giudecca'}],
          'sampledFeature': [u'lagunaVenezia002']
      },
      'identifier': {
          'codespace': 'http://www.opengis.net/def/nil/OGC/0/unknown',
          'value': 'record410549'
      },
      'observedProperty': u'http://vocab.nerc.ac.uk/collection/P02/current/ASLV/',
      'phenomenonTime': '2015-01-12T11:40:00+00:00',
      'procedure': u'http://sp7.irea.cnr.it/sensors/mareesk.irea.cnr.it/procedure/
↪SIAPMICROS/DA9000/noSerialNumberDeclared/20140723044959616_PuntaSaluteCanaleGiudecca
↪',
      'result': {
          'uom': u'm', 'value': 0.03
      },
      'resultTime': '2015-01-12T11:40:00+00:00',
      'type': 'http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement'
  },
  'offering': u'offering:http://sp7.irea.cnr.it/sensors/mareesk.irea.cnr.it/procedure/
↪SIAPMICROS/DA9000/noSerialNumberDeclared/20140723044959616_
↪PuntaSaluteCanaleGiudecca/observations',
  'request': 'InsertObservation',
  'service': 'SOS',
  'version': '2.0.0'
 }
```

The requests should be sent to SOS endpoint that could be found in the GET-IT interface (Services -> SOS e.g. http://demo2.get-it.it/about_services/#sos), by client-side URL transfers (e.g. cURL).

The tutorials are based around performing tasks, like adding data or publishing maps. The tutorials are written in a workshop like format and are broken into two groups *Users* and *Administrators*.

Users Tutorials In the user workshop you will learn how to create an account on GET-IT, add layers, maps, sensors and observations to your account as well as publishing those.

Administrators Tutorials The administrator workshop will guide you through the installation and configuration of GET-IT, as well as explore further possibilities with GET-IT.

Data Flow Engine Tutorials TODO

```
This section is on development
```

The Tutorials section contains step-by-step workshops that are oriented around performing particular sets of tasks, like adding data or publishing maps, setting up and maintaining a server, or setting up a project to extend from GET-IT. These tutorials are written in a workshop-like format and are broken into three groups: Users, Administrators and Developers.

Tutorials

CHAPTER 5

Need Help?

Having trouble? Can't find what you are looking for? We'd like to help!!

- Write us an email to: help.skritmare@irea.cnr.it
- Report bugs in GET-IT in our ticket tracker.