

How to use the Oceanontron's OGC/SOS frontdesk

Table of Contents

1 Purpose.....	4
2 External references.....	4
3 Which datasets are disseminated with OGC/SOS.....	4
4 Which software should I use to request Oceanotron/OGC/SOS frontend?.....	5
5 How oceanotron's OGC/SOS interface basically works ?.....	5
6 How to discover what is provided by the Oceanotron/SOS service ?.....	5
7 Get Observations.....	9
7.1 General information.....	9
7.2 Subset a dataset.....	10
7.3 Get the result.....	12

1 Purpose

This document describes how to use the OGC/SOS interface provided by oceanotron server.

SOS is the web service request standard for SWE (Sensor Web Enablement) framework.

2 External references

The detailed OGC/SOS protocol standard is given at https://portal.opengeospatial.org/files/?artifact_id=47599

Sensor Web Enablement : <http://www.opengeospatial.org/ogc/markets-technologies/swe>

3 Which datasets are disseminated with OGC/SOS

The oceanotron application provides OGC/SOS access to every dataset having as feature type:

- profiles
- pointSeries
- trajectories

The OGC/SOS interface gives full access on oceanotron datasets and observation data.

It enables dataset and observation subsetting with the following criteria:

- coordinates (x,y,z,t)
- phenomenon or observed property (e.g. temperature)
- platform code

The results are in the following format:

- XML (O&M model, v2.0¹)
- JSON (O&M model, v2.0)
- netCDF4 (US-NODC templates²)

Below a configurable volume of data, the result can be downloaded synchronously in a delay compliant with the web request time scales (a few seconds).

The data can also be downloaded asynchronously, especially for bigger volume, when the result processing is longer.

1 http://portal.opengeospatial.org/files/?artifact_id=41510

2 <http://www.nodc.noaa.gov/data/formats/netcdf/>

4 Which software should I use to request Oceanotron/OGC/SOS frontend?

As the OGC/SOS standard version implemented is 2.0, there are no known GIS application which can request the OGC/SOS oceanotron's frontend.

However, it is possible to request it in command line using **curl** linux command line (as shown in the examples given in the tutorial).

It is also possible to develop web application front-end using **javascript** to query the SOS service. An implementation in OpenLayer3 is expected in the future.

5 How oceanotron's OGC/SOS interface basically works ?

The service can be requested as follow:

`http://<oceanotron server url>/SOS/default?<sos request>`

For example:

<http://www.ifremer.fr/oceanotron/SOS/default?service=SOS&request=getCapabilities>

The SOS request can be send with GET HTTP request with Key-Value pairs. It can also be send as an XML request with POST HTTP.

The request supported by Oceanotron/SOS are:

GetCapabilities	Discover the capabilities (offering, observed properties, procedure) provided by the service
GetObservation	Subset and access observation data

Vocabularies: the OGC/SWE standards uses specific terms which have their 'translation' in oceanotron or oceanography community:

Offering	Dataset or collection of observations
Procedure	Observation platform, instrument or sensor
Observed properties	Ocean variables (temperature, salinity, ...)

6 How to discover what is provided by the Oceanotron/SOS service ?

Do a **getCapabilities** request, for example:

<http://www.ifremer.fr/oceanotron/SOS/default?service=SOS&request=getCapabilities>

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns24:Capabilities xmlns:fra="http://www.cnig.gouv.fr/2005/fra">
  (...)
  <ows:OperationsMetadata>
    <!-- #####
    one section for each allowed operation:
    - GetCapabilities,
    - GetObservation,
    - GetObservationById
    #####-->
    <!-- ##### getCapabilities #####-->
    <ows:Operation name="GetCapabilities">
      (...)
    </ows:Operation>
    <!-- ##### getObservation #####-->
    <ows:Operation name="GetObservation">
      <ows:DCP>
        <ows:HTTP>
          <ows:Post xlink:href="http://visi-oceanotron-
tomcat.ifremer.fr/oceanotron/SOS/sos/default?"/>
        </ows:HTTP>
      </ows:DCP>
      (...)
      <ows:Parameter name="offering">
        <ows:AllowedValues>
          <ows:Value>INS-CORIOLIS-GLO-NRT-
OBS_TRAJECTORIES_LATEST</ows:Value>
          <ows:Value>Oceansites-TENATSO</ows:Value>
          (...)
        </ows:AllowedValues>
      </ows:Parameter>
      (...)
      <ows:Parameter name="procedure">
        <ows:AllowedValues>
          <ows:Value>INS-CORIOLIS-GLO-NRT-
OBS_TRAJECTORIES_ARCHIVE#25545</ows:Value>
          <ows:Value>INS-CORIOLIS-GLO-NRT-
OBS_TRAJECTORIES_ARCHIVE#25546</ows:Value>
          (...)
        </ows:AllowedValues>
      </ows:Parameter>
      <ows:Parameter name="observedProperty">
        <ows:AllowedValues>
          <ows:Value>sea_water_pressure</ows:Value>
          <ows:Value>sea_water_temperature</ows:Value>
          <ows:Value>sea_water_electrical_conductivity</ows:Value>
          (...)
        </ows:AllowedValues>
      </ows:Parameter>
      <ows:Parameter name="featureOfInterest">
        <ows:AllowedValues>
          <ows:Value>http://www.seadatanet.org/collection/sampling-
features/current/trajectory</ows:Value>
          <ows:Value>http://www.seadatanet.org/collection/sampling-
features/current/profile</ows:Value>
          <ows:Value>http://www.seadatanet.org/collection/sampling-
features/current/pointSeries</ows:Value>
        </ows:AllowedValues>
      </ows:Parameter>

```

```
(...)
<ows:Parameter name="responseFormat">
  <ows:AllowedValues>
    <ows:Value>text/xml;subtype="http://www.opengis.net/om/2.0"</ows:Value>
    <ows:Value>application/json;subtype="http://www.opengis.net/om/2.0"</ows:Value>
    <ows:Value>application/netcdf</ows:Value>
  </ows:AllowedValues>
</ows:Parameter>
(...)
<ows:Parameter name="responseMode">
  <ows:AllowedValues>
    <ows:Value>inline</ows:Value>
    <ows:Value>out-of-band</ows:Value>
  </ows:AllowedValues>
</ows:Parameter>
</ows:Operation>
<!-- ##### getObservationById ##### -->
<ows:Operation name="GetObservationById">
  <ows:DCP>
    <ows:HTTP>
      <ows:Post xlink:href="http://visi-oceanotron-
tomcat.ifremer.fr/oceanotron/SOS/sos/default?"/>
    </ows:HTTP>
  </ows:DCP>
</ows:Operation>
(...)
</ows:OperationsMetadata>
<ns24:filterCapabilities>
  <fes:Filter_Capabilities>
    (...)
  </fes:Filter_Capabilities>
</ns24:filterCapabilities>
<ns24:contents>
  <ns24:Contents>
    <swes:offering>
      <ns24:ObservationOffering swes:id="INS-CORIOLIS-GLO-NRT-OBS_TRAJECTORIES_LATEST">
        <swes:description/>
        <swes:name>INS-CORIOLIS-GLO-NRT-OBS_TRAJECTORIES_LATEST</swes:name>
        <swes:procedure>INS-CORIOLIS-GLO-NRT-OBS_TRAJECTORIES_LATEST</swes:procedure>
      </ns24:ObservationOffering>
    </swes:offering>
    <swes:observableProperty>sea_water_salinity</swes:observableProperty>
    <swes:observableProperty>sea_water_temperature</swes:observableProperty>
    <swes:observableProperty>mole_concentration_of_dissolved_molecular_oxygen_in_sea_w
ater</swes:observableProperty>
    <ns24:observedArea>
      <ns19:Envelope>
        <ns19:lowerCorner>-179.998 -75.927</ns19:lowerCorner>
        <ns19:upperCorner>180.0 89.919</ns19:upperCorner>
      </ns19:Envelope>
    </ns24:observedArea>
    <ns24:phenomenonTime/>
  </ns24:Contents>
</ns24:contents>
</ns24:filterCapabilities>
</ows:OperationsMetadata>
</ows:ServiceMetadataBase>
</wfs:GetCapabilitiesResponse>
```



```

<ns24:responseFormat>text/xml;subtype="http://www.opengis.net/om/2.0"</ns24:responseFormat>

<ns24:responseFormat>application/json;subtype="http://www.opengis.net/om/2.0"</ns24:responseFormat>
    <ns24:responseFormat>application/netcdf</ns24:responseFormat>

<ns24:observationType>http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Observation</ns24:observationType>
    </ns24:ObservationOffering>
</swes:offering>
    (...)
</ns24:Contents>
</ns24:contents>
</ns24:Capabilities>

```

Table 1: The result is an XML stream as follow (commented output)

The result gives details about each supported operation (getCapabilities, getObservation). For each, allowed parameters and parameter's values are detailed.

Note version 1.3.2: getObservationById is not currently supported although it is described in the getCapabilities.

7 Get Observations

7.1 General information

The getObservation operation is used to subset and access a dataset.

On oceanotron is has been tested with POST request with XML encoded query. For example:

URL	http://oceanotrndemo1.ifremer.fr/oceanotron/SOS/default? service=SOS&request=getObservation
POST query	<pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <sos:GetObservation service="SOS" version="2.0.0" xmlns:sos="http://www.opengis.net/sos/2.0" xmlns:fes="http://www.opengis.net/fes/2.0" xmlns:gml="http://www.opengis.net/gml/3.2" > <sos:offering>ARGO-FILE-PROFILE</sos:offering> <sos:procedure>ARGO-FILE-PROFILE#1900063</sos:procedure> <sos:observedProperty>sea_water_temperature</sos:observedProperty> <sos:temporalFilter> <fes:After> </pre>

```

<fes:ValueReference></fes:ValueReference>
<gml:TimeInstant>
  <gml:timePosition>2002-03-26T05:36:00.000+01:00</gml:timePosition>
</gml:TimeInstant>
</fes:After>
</sos:temporalFilter>

<sos:responseFormat>text/xml;subtype="http://www.opengis.net/om/2.0"</sos:responseFormat>
</sos:GetObservation>

```

With CURL command line, the request can be launched as follow:

```

curl -X POST -d @getObservationARGO-xml.xml
'http://oceanotrndemo1.ifremer.fr/oceanotron/SOS/default?
service=SOS&request=getObservation' --header "Content-Type:text/xml" >
results/getObservationARGO-xml.xml

```

Where:

getObservationARGO-xml.xml contains the XML POST query
and results/getObservationARGO-xml.xml is the file where the result is downloaded.

In the post criteria the **offering** must be given. The offering criteria gives the dataset from which where observation are requested. The available offerings are listed in the getCapabilities (see 6).

7.2 Subset a dataset

The criteria on which subsetting applies are:

- observed properties (mandatory in version 1.3.2, *to be done* optional in version 1.3.3)
- procedure
- geographical coordinates
- vertical level coordinates (not available in version 1.3.2, *to be done* in version 1.3.3)
- temporal coordinates

Each criteria is optional.

Criteria are combined with AND logical operators (e.g. procedure AND geographical coordinates).

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sos:GetObservation
  service="SOS" version="2.0.0"
  xmlns:sos="http://www.opengis.net/sos/2.0" >
  <sos:offering>ArgoNetCDFToProfile</sos:offering>
  <sos:observedProperty>TEMP</sos:observedProperty>

  <sos:responseFormat>text/xml;subtype="http://www.opengis.net/om/2.0"</sos:responseFormat>
</sos:GetObservation>

```

Table 2: Subset with observed property criteria

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns20:GetObservation version="2.0.0" service="SOS"
xmlns:ns20="http://www.opengis.net/sos/2.0">
  <ns20:procedure>Oceansites-EUROSITES#CIS-2</ns20:procedure>
  <ns20:offering>Oceansites-EUROSITES</ns20:offering>
  <ns20:observedProperty>sea_water_temperature</ns20:observedProperty>
  <ns20:observedProperty>sea_water_salinity</ns20:observedProperty>

  <ns20:responseFormat>text/xml;subtype="http://www.opengis.net/om/2.0"</ns20:responseFormat>
</ns20:GetObservation>

```

Table 3: Subset with procedure criteria

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sos:GetObservation service="SOS" version="2.0.0"
  xmlns:sos="http://www.opengis.net/sos/2.0"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  >
  <sos:offering>ArgoNetCDFToProfile</sos:offering>
  <sos:observedProperty>sea_water_temperature</sos:observedProperty>
  <sos:spatialFilter>
    <fes:BBOX>
      <fes:ValueReference>whatever</fes:ValueReference>
      <gml:Envelope srsName="urn:ogc:crs:espg:4326">
        <gml:lowerCorner>-21.0 25.0</gml:lowerCorner>
        <gml:upperCorner>0.0 30.0</gml:upperCorner>
      </gml:Envelope>
    </fes:BBOX>
  </sos:spatialFilter>

  <sos:responseFormat>text/xml;subtype="http://www.opengis.net/om/2.0"</sos:responseFormat>
</sos:GetObservation>

```

Table 4: Subset with geographical criteria

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sos:GetObservation service="SOS" version="2.0.0"
  xmlns:sos="http://www.opengis.net/sos/2.0"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  >
  <sos:offering>ArgoNetCDFToProfile</sos:offering>
  <sos:observedProperty>sea_water_temperature</sos:observedProperty>
  <sos:temporalFilter>
    <fes:After>
      <fes:ValueReference></fes:ValueReference>
      <gml:TimeInstant>
        <gml:timePosition>2002-03-26T05:36:00.000+01:00</gml:timePosition>
      </gml:TimeInstant>
    </fes:After>
  </sos:temporalFilter>

  <sos:responseFormat>text/xml;subtype="http://www.opengis.net/om/2.0"</sos:response
Format>
</sos:GetObservation>

```

Table 5: Subset with temporal criteria

7.3 Get the result

You can choose the format of your result:

- XML (O&M model, v2.0)
- JSON (O&M model, v2.0)
- netCDF4 (US-NODC templates)

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns20:GetObservation version="2.0.0" service="SOS"
  xmlns:ns20="http://www.opengis.net/sos/2.0">
  <ns20:procedure>Oceansites-EUROSITES#CIS-2</ns20:procedure>
  <ns20:offering>Oceansites-EUROSITES</ns20:offering>
  <ns20:observedProperty>sea_water_temperature</ns20:observedProperty>
  <ns20:observedProperty>sea_water_salinity</ns20:observedProperty>

  <ns20:responseFormat>text/xml;subtype="http://www.opengis.net/om/2.0"</ns20:respon
seFormat>
</ns20:GetObservation>

```

Table 6: request XML (O&M model, v2.0) result

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns20:GetObservation version="2.0.0" service="SOS"
xmlns:ns20="http://www.opengis.net/sos/2.0">
  <ns20:procedure>Oceansites-EUROSITES#CIS-2</ns20:procedure>
  <ns20:offering>Oceansites-EUROSITES</ns20:offering>
  <ns20:observedProperty>sea_water_temperature</ns20:observedProperty>
  <ns20:observedProperty>sea_water_salinity</ns20:observedProperty>

  <ns20:responseFormat>application/json;subtype="http://www.opengis.net/om/2.0"</ns20:responseFormat>
</ns20:GetObservation>
```

Table 7: request JSON (O&M model, v2) result

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sos:GetObservation service="SOS" version="1.0.0"
xmlns:sos="http://www.opengis.net/sos/1.0">
  <sos:offering>ArgoNetCDFToProfile</sos:offering>
  <sos:procedure>ArgoNetCDFToProfile#1900063</sos:procedure>
  <sos:observedProperty>sea_water_temperature</sos:observedProperty>
  <sos:responseFormat>application/netcdf</sos:responseFormat>
</sos:GetObservation>
```

Table 8: request netCDF file result

When the volume is expected to be big, you can request the data out-of-band.

When result is requested out-of-band, the request immediately sends back to the user the URL where the file will be prepared.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sos:GetObservation service="SOS" version="1.0.0"
xmlns:sos="http://www.opengis.net/sos/1.0">
  <sos:offering>ArgoNetCDFToProfile</sos:offering>
  <sos:procedure>ArgoNetCDFToProfile#1900063</sos:procedure>
  <sos:observedProperty>sea_water_temperature</sos:observedProperty>
  <sos:responseFormat>application/netcdf</sos:responseFormat>
  <sos:resultModel>om:Measurement</sos:resultModel>
  <sos:responseMode>out-of-band</sos:responseMode>
</sos:GetObservation>
```

Table 9: Request with out-of-band result (asynchronous for big volumes)

```
#curl -X POST -d @getObservationARGOoutofbandintegration.xml
'http://www.ifremer.fr/oceanotron/SOS/default?service=SOS&request=getObservation'
--header "Content-Type:text/xml"
```

The request immediately send back the URL where the file is processed, for example:

```
http://www.ifremer.fr/oceanotron/SOS/download/ArgoNetCDFToProfile-f4a05844-aba7-4f17-b2e5-69e77a9d2aef.nc
```

Note 1.3.2: the URL sent by the server is not the correct one. This will be corrected soon

Until the file is ready, the requested file URL send back a 204 status (service temporally unavailable):

When the result is ready, the file can be donwloaded:

```
wget http://www.ifremer.fr/oceanotron/SOS/download/ArgoNetCDFToProfile-f4a05844-aba7-4f17-b2e5-69e77a9d2aef.nc
```