



Documentation
of
RESTful SOS

Document Change Control

<i>Revision</i>	<i>Date Of Issue</i>	<i>Author(s)</i>	<i>Brief Description Of Changes</i>
0.1	16.10.12	Eike Hinderk Jürrens	Initialization
0.2	17.10.12	Eike Hinderk Jürrens	Revision of background
0.3	24.10.12	Eike Hinderk Jürrens	Addition of example requests / responses
0.4	25.10.12	Eike Hinderk Jürrens	Addition of example requests / responses
0.5	26.10.12	Arne Bröring	Polishing
0.6	30.10.12	Eike Hinderk Jürrens	Updated Sections 5 - 7
0.7	05.11.12	Eike Hinderk Jürrens	Fix Image Bug
0.8	07.11.12	Eike Hinderk Jürrens	Added 5.1Service Endpoint
0.9	09.11.12	Eike Hinderk Jürrens	Updated 5.3.2
0.10	12.11.12	Eike Hinderk Jürrens	Updated 5.6.1.2, 5.5.2.2
0.11	11.12.12	Eike Hinderk Jürrens	Added 5.2Resource Relations, updated 7.3.2
0.12	12.12.12	Eike Hinderk Jürrens	Updated 6, 4.3
0.13	02.04.13	Eike Hinderk Jürrens	Updated 4.2, 4.3, 6

Licence

This document is part of 52° North

Copyright (C) 2012 by 52 North Initiative for Geospatial Open Source Software GmbH

Contact: Andreas Wytzisk,
52 North Initiative for Geospatial Open Source Software GmbH,
Martin-Luther-King-Weg 24,
48155 Muenster, Germany, info@52north.org

This program is free software; you can redistribute and/or modify it under the terms of the GNU General Public License version 2 as published by the Free Software Foundation.

This program is distributed WITHOUT ANY WARRANTY; even without the implied WARRANTY OF MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program (see gnu-gpl v2.txt). If not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA or visit the Free Software Foundation web page, <http://www.fsf.org>.

For more information, contact:

52° North Initiative for Geospatial Open Source Software GmbH
Martin-Luther-King-Weg 24
48155 Münster, Germany

52north.org

Table of contents

1 Introduction.....	5
1.1 Scope.....	5
1.2 SOS.....	5
1.3 Relation Between SOS and RESTful SOS.....	5
2 Architecture.....	6
2.1 Workflow.....	9
3 System Requirements.....	10
4 Installation Procedure.....	10
4.1 Get the Programmes.....	10
4.2 Install and Create Database.....	11
4.3 Install RESTful SOS.....	11
5 Functionality.....	12
5.1 Service Endpoint.....	12
5.2 Resource Relations.....	12
5.3 Capabilities.....	12
5.4 Features.....	14
5.5 Observations.....	17
5.6 Offerings.....	22
5.7 Sensors.....	24
6 Developer Guide.....	30
7 Appendix.....	31
7.1 HTTP Response Codes.....	31
7.2 Troubleshooting.....	31
7.3 Known Issues.....	31

1 Introduction

1.1 Scope

This document describes

- the architecture,
- the installation of the RESTful SOS, and
- the set-up for developers.

The document is structured as follows. This section gives a general introduction to the SOS. Right after this, the relationship between the SOS and the RESTful SOS is explained in detail.

Section 2 contains the big picture by giving an overview of the architecture of the RESTful SOS. In addition, the work flow within the RESTful SOS and the communication with the 52° North SOS framework is outlined.

Section 3 lists the requirements of the RESTful SOS in terms of software components and versions. The installation procedure is described in Section 4 including the set-up of the required database and configuration adjustments when using the release archive.

The functionality of the RESTful SOS in terms of available resources and allowed operations are explained in detail in Section 5. This includes example requests and responses.

Instructions on how to set-up a development environment are given in Section 6.

Any additional information is contained in the last Section 7.

1.2 SOS

The SOS provides a standardized interface for managing and retrieving metadata and observations from heterogeneous sensor systems. Sensor systems contribute the largest part of geospatial data used in geospatial systems today. Sensor systems include for example in situ sensors (e.g. river gauges), moving sensor platforms (e.g. satellites or unmanned aerial vehicles) or networks of static sensors (e.g. seismic arrays). Used in conjunction with other OGC specifications the SOS provides a broad range of interoperable capability for discovering, binding to and interrogating individual sensors, sensor platforms, or networked constellations of sensors in real-time, archived or simulated environments (see [OGC 12-006]).

1.3 Relation Between SOS and RESTful SOS

The RESTful SOS provides means for accessing observations in a RESTful way. Therefore, several resources (see Section 5 for more details on requesting and receiving resources) are defined:

- Capabilities
- Offerings
- Sensors
- Features
- Observations

The implementation is based on the 52° North SOS by providing a RESTful binding. A binding (see Section 2 for more details) is a plug-in to the 52° North SOS that enables the understanding of another protocol. Further, the dynamic en- and decoder functionality of the 52° North SOS is extended for supporting the sosREST XML-schema ([link](#) to schema repository) as defined for this project.

In relation to the SOS 2.0 standard [OGC 12-006], the information received is extended. For details,

take a look at Section 5.

2 Architecture

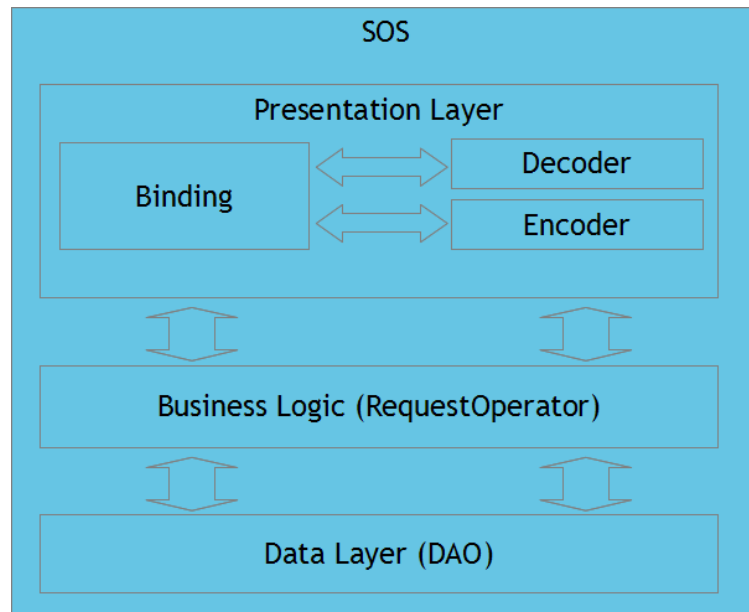


Figure 1: 52°North SOS core architecture

The 52°North SOS implementation is separated into three tiers (Figure 1):

- presentation layer,
- business logic, and
- data layer.

The presentation layer contains the implementation of the protocol bindings. They provide means for accessing the SOS via their implemented protocol. This includes the de- and encoder for the protocol specific encoding.

The business logic tier provides the means for calling the operations of the SOS, e.g., GetCapabilities, GetObservations, and DescribeSensor.

The lowest tier, the data layer, provides the means for accessing the data storage, most commonly a database. This layer does not need to be touched for the implementation of the RESTful SOS.

To develop the RESTful SOS the most implementation takes place in the presentation layer with calls to the business logic layer. As being RESTful requires an additional protocol, for which the binding loading functionality is used. Hence, de- and encoders are implemented for the RESTful requests and responses. The work flow is outlined in Section 2.1.

Next, the package structure of the RESTful SOS binding is explained in more detail.

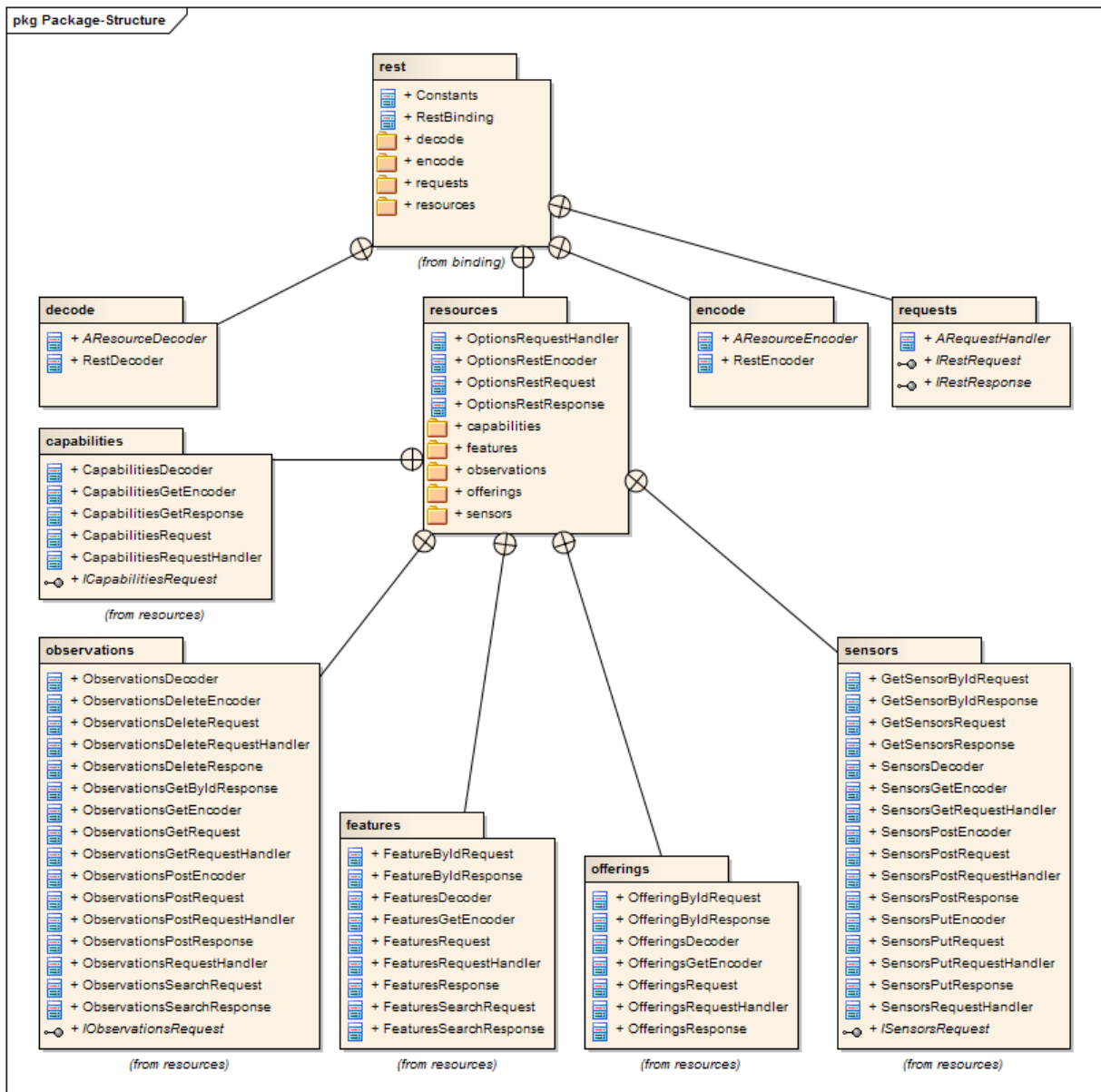


Figure 2: RESTful binding: package overview

The packages outlined in Figure 2 contain the following elements (for simplification, the package names are reduced in the following by removing the preceding “org.n52.sos.binding.”):

- rest
 - Core package
 - Binding implementation (reference to SOS core Binding class)
 - Constants class with configured properties
 - Controller for the whole binding
 - decode
 - Decoder controller
 - Decoder implementation with abstract super class offering common decoder functionality
 - Implementation of `IDecoder<T,S>` from SOS core
 - encode
 - Encoder controller
 - Encoder implementation with abstract super class offering common encoder

- functionality
 - Implementation of Encoder<T,S> from SOS core
- requests
 - Interfaces for the internal message objects
 - Abstract super class for the request handling classes (implemented for each resource type)
- resources
 - Containing the packages for each resource
 - HTTP OPTIONS handling classes
 - capabilities
 - Everything regarding the capabilities resource
 - features
 - Everything regarding the features resource
 - observations
 - Everything regarding the observations resource
 - observations
 - Everything regarding the observations resource
 - offerings
 - Everything regarding the offerings resource
 - sensors
 - Everything regarding the sensors resource

2.1 Workflow

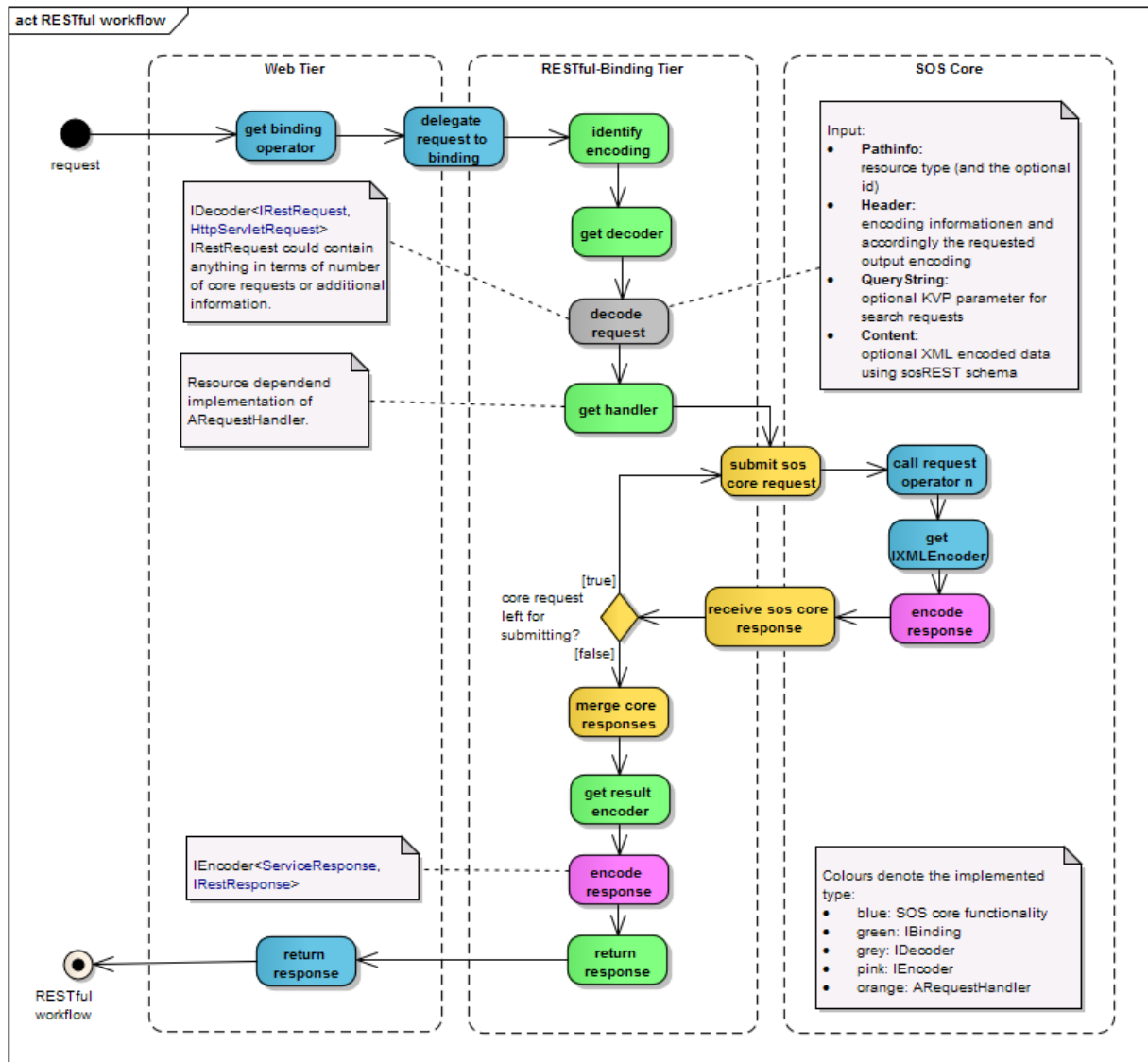


Figure 3: RESTful work flow: integration of RESTful binding in SOS architecture tiers

The work flow of the RESTful SOS is outlined in Figure 3. The following steps are performed:

- Each request is delegated from SOS web tier to corresponding binding implementation using the URL pattern.
- The binding implementation is the class RestBinding and the controller of the whole process.
- For each request, the following steps are executed:
 - Decode request
 - Handle request
 - Encode response
 - [Optional: error handling]
- The decoding results in an IRestRequest object which is delegated to the right ARequestHandler implementation depending on the type of the request.

- The handling of the request includes:
 - Send requests to SOS core
 - Handle response from core and encode into internal response types.
- The encoding of the internal results is done by the resource specific RestEncoder implementations and result in an SOS core ServiceResponse
- The error handling is done in the RestBinding class which catches error specific Exceptions and encodes these into a ServiceResponse having error codes as specified in Appendix 7.1.
- The HTTP OPTIONS method is handled using classes from the package resources.

3 System Requirements

- The following software needs to be pre-installed (in brackets the versions that the RESTful SOS has been tested for).
 - Windows 2000 or higher, Linux/Unix [tested with Windows 7 SP1]
 - JRE/JDK 6 [1.6.x]
 - Apache Tomcat [7.0.x, 6.0.24]
 - PostgreSQL Version [9.0.x]
 - PostGIS Version [2.0.x]
 - SVN client
 - Apache Maven [3.0.3]

4 Installation Procedure

4.1 Get the Programmes

- Tomcat
 - Download Apache Jakarta Tomcat from:
<http://tomcat.apache.org/>
Follow the installation instructions given on the Apache website to install the Apache Jakarta Tomcat.
- Maven
 - Install Apache Maven from:
<http://maven.apache.org/>
Follow the installation instructions given on the website to install the Apache Maven.
- PostgreSQL
 - Download PostgreSQL from:
<http://www.postgresql.org/download/>
 - Install PostgreSQL as described in the PostgreSQL manual available at
<http://www.postgresql.org/docs/manuals/>
including the installation of JDBC's
- PostGIS

- Download PostGIS from:
<http://www.postgis.org/download/>
or use the PostgreSQL → Application Stack Builder
Install PostGIS as described in the PostGIS documentation available at
<http://www.postgis.org/documentation/>
- JDBC driver
 - Install the JDBC driver for PostGIS and PostgreSQL in the shared libs folder of the servlet container (e.g. Tomcat: \$CATALINA_HOME/lib):
 - PostGIS: postgis-jdbc-2.x.y.jar
 - Extract prior download PostGIS distribution and go to folder java/jdbc
 - Run “mvn package” and use artefact from folder java/jdbc/target
 - For troubleshooting refer to the file README in that folder
 - PostgreSQL: postgresql-9.x-xyz.jdbc4.jar (from: e.g. Windows: C:\Program Files (x86)\PostgreSQL\pgJDBC)

4.2 Install and Create Database

1. Install PostgreSQL
2. Install PostGIS
3. Create one PostGIS enabled database (e.g. using template template_postgis)

4.3 Install RESTful SOS

1. Deploy the war file in the servlet container.
2. Go to WEB_APP/install/index and follow the steps to set-up the service
3. Go to WEB_APP/admin/settings#restful_binding and adjust the settings to the needs of your environment.
4. Go to WEB_APP/admin/settings#service_settings and adjust the SOS URL.
5. OPTIONAL: To disable other bindings than the RESTful binding, do the following:
 1. Stop the WEB_APP.
 2. Go to WEB_APP/WEB_INF/lib.
 3. Remove all 52n-sos-binding-*.jar files where * NOT contains rest.
 4. Restart the WEB_APP.

5 Functionality

This section describes the functionality of the SDC in terms of request and response per resource and request type.

5.1 Service Endpoint

The service is redirecting all requests to `/SDC/sos/rest/` to a configured default resource, e.g. `/SDC/sos/rest/capabilities` using the HTTP response 303 “see other” and the Location header field. For changing this behaviour use the configuration entry `default.landingpoint.resource`.

5.2 Resource Relations

Relation Identifier	Description ¹
features-get	Link between this resource and the according feature resource collection.
feature-get	Link between this resource and ONE according feature resource.
sensor-create	Link to create new sensor resources.
sensor-update	Link to create an existing sensor resource.
sensor-get	Link between this resource and ONE according sensor resource.
sensors-get	Link between this resource and the according sensor resource collection.
sensor-delete	Link to delete an existing sensor resource. ²
offerings-get	Link between this resource and the according offering resource collection.
offering-get	Link between this resource and ONE according offering resource
self	Link to this resource.
property-get	Link to ONE according property resource (hosted externally).
observation-create	Link to create observation resources.
observation-delete	Link to delete an existing observation resource.
observation-get	Link to get ONE observation resource.
observations-get	Link between this resource and the according observation resource collection.

Table 1: Resource Relation Identifier

5.3 Capabilities

5.3.1 Request

```
GET /SDC/sos/rest/capabilities HTTP/1.1
Accept: application/gml+xml
```

Listing 1: HTTP GET for capabilities resource

- 1 The contained examples require URL encoding before being submitted.
- 2 Not yet implemented.

5.3.2 Response

Header:

```
Allow: GET, OPTIONS
Content-Type: application/gml+xml
Status: 200 OK
[...]
```

Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<sosREST:Capabilities xmlns:sosREST="http://www.opengis.net/sosREST/1.0">
  <ows:ServiceIdentification />
  <ows:ServiceProvider />
  <extension />
  <filterCapabilities />
</sosREST:Capabilities>
<link rel="http://www.opengis.net/sosREST/1.0/self"
href="http://vpn.52north.org:80/SDC/sos/rest/capabilities" type="application/gml+xml"/>
<link rel="http://www.opengis.net/sosREST/1.0/offering-get"
href="http://vpn.52north.org:80/SDC/sos/rest/offering-get" type="application/gml+xml"/>
<link rel="http://www.opengis.net/sosREST/1.0/offering-get"
href="http://vpn.52north.org:80/SDC/sos/rest/offering-get" type="application/gml+xml"/>
<link rel="http://www.opengis.net/sosREST/1.0/offering-get"
href="http://vpn.52north.org:80/SDC/sos/rest/offering-get" type="application/gml+xml"/>
<link rel="http://www.opengis.net/sosREST/1.0/offering-get"
href="http://vpn.52north.org:80/SDC/sos/rest/offering-get" type="application/gml+xml"/>
[...]
```

Listing 2: HTTP GET response for capabilities resource

5.4 Features

It is only possible to request feature resources hosted by the RESTful SOS. Features referenced from other services within sensor descriptions or observations are not considered.

5.4.1 Resource Collection

5.4.1.1 Request

```
GET /SDC/sos/rest/features HTTP/1.1
Accept: application/gml+xml
```

Listing 3: HTTP GET for features resource collection

5.4.1.2 Response

Header:

```
Allow: GET, OPTIONS
Content-Type: application/gml+xml
Status: 200 OK
[...]
```

Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<sosREST:FeatureCollection
  xmlns:sosREST="http://www.opengis.net/sosREST/1.0">
  <sosREST:link rel="http://www.opengis.net/sosREST/1.0/foi-get"
    href="http://sdc.host.example.com:80/SDC/sos/rest/features/foi_1"
    type="application/gml+xml"/>
  <sosREST:link rel="http://www.opengis.net/sosREST/1.0/foi-get"
    href="http://sdc.host.example.com:80/SDC/sos/rest/features/foi_2"
    type="application/gml+xml"/>
  <sosREST:link rel="http://www.opengis.net/sosREST/1.0/self"
    href="http://sdc.host.example.com:80/SDC/sos/rest/features"
    type="application/gml+xml"/>
</sosREST:FeatureCollection>
```

Listing 4: HTTP GET response for features resource collection

5.4.2 Search

5.4.2.1 Request

```
GET /SDC/sos/rest/features?procedures=52 HTTP/1.1
Accept: application/gml+xml
```

Listing 5: HTTP GET for features resource search

The KVP encoded parameters in Table 1 are allowed for this HTTP method and resource combination.

Filter Property	Description ³
feature	List of FOI resources to use as a spatial filter. Example: ?feature=foi_1,foi2
observedproperties	List of observed properties to use as a filter. Example: ?observedproperties=http://sweet.jpl.nasa.gov/2.0/hydroSurface.owl#Discharge
procedures	List of procedure IDs to use as a filter. Example: ?procedures=52,54
spatialfilter	Kvp Encoded spatial filter as defined in OGC#12-006. Supported types: BBOX. Example: ?namespaces=xmlns(sams,http://www.opengis.net/samplingSpatial/2.0), xmlns(om,http://www.opengis.net/om/2.0)& spatialFilter=om:featureOfInterest/*sams:shape,1.0,1.0,60.0,60.0,http:// www.opengis.net/def/crs/EPSSG/0/4326
namespaces ⁴	Namespace definition used by spatial- and temporalfilter. Example: ?namespaces=xmlns(sams,http://www.opengis.net/samplingSpatial/2.0), xmlns(om,http://www.opengis.net/om/2.0)

Table 2: Allowed KVP parameters for the feature resource

5.4.2.2 Response

Header:

Allow: GET, OPTIONS
Content-Type: application/gml+xml
Status: 200 OK
[...]

Body:

```
<?xml version="1.0" encoding="UTF-8"?>
< sosREST:FeatureCollection
  xmlns:sosREST="http://www.opengis.net/sosREST/1.0">
  < sosREST:link rel="http://www.opengis.net/sosREST/1.0/foi-get"
    href="http://sdc.host.example.com:80/SDC/sos/rest/features/foi_1"
  type="application/gml+xml"/>
  < sosREST:link rel="http://www.opengis.net/sosREST/1.0/foi-get"
    href="http://sdc.host.example.com:80/SDC/sos/rest/features/foi_2"
  type="application/gml+xml"/>
  < sosREST:link rel="http://www.opengis.net/sosREST/1.0/self"
    href="http://sdc.host.example.com:80/SDC/sos/rest/features"
  type="application/gml+xml"/>
</ sosREST:FeatureCollection>
```

Listing 6: HTTP GET response for features resource search

- 3 The contained examples require URL encoding before being submitted.
- 4 Parameter namespaces is required for spatialfilter and temporalfilter. See OGC SOS 2.0.0 specification for more details ([OGC#12-006](#)).

5.4.3 Single Resource

5.4.3.1 Request

```
GET /SDC/sos/rest/features/foi_1 HTTP/1.1
Accept: application/gml+xml
```

Listing 7: HTTP GET for a feature resource

5.4.3.2 Response

Header:

```
Allow: GET, OPTIONS
Content-Type: application/gml+xml
Status: 200 OK
[...]
```

Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<sosREST:Feature
  [...]>
  <sams:SF_SpatialSamplingFeature gml:id="sf_1">
    <gml:identifier codeSpace="">foi_1</gml:identifier>
    <sf:type xlink:href="http://www.opengis.net/def/samplingFeatureType/OGC-
      OM/2.0/SF_SamplingPoint"/>
    <sf:sampledFeature xlink:href="foi_20"/>
    <sams:shape>
      <gml:Point gml:id="point_sf_1">
        <gml:pos srsName="http://www.opengis.net/def/crs/EPSSG/0/4326">
          42.0 7.5
        </gml:pos>
      </gml:Point>
    </sams:shape>
  </sams:SF_SpatialSamplingFeature>
  <sosREST:link rel="http://www.opengis.net/sosREST/1.0/self"
    href="http://sdc.host.example.com:80/SDC/sos/rest/features/foi_1"
    type="application/gml+xml"/>
</sosREST:Feature>
```

Listing 8: HTTP GET response for a feature resource

5.5 Observations

5.5.1 Create

5.5.1.1 Request

```

POST /SDC/sos/rest/observations HTTP/1.1
Content-Type: application/gml+xml

<?xml version="1.0" encoding="UTF-8"?>
< sosREST:Observation
  [...] >
  < om:OM_Observation gml:id="o1" >
    < gml:identifier codespace="" >
      d4a0fd7c-8b74-45f0-bfae-325842f877c7
    < /gml:identifier >
    < om:phenomenonTime >
      < gml:TimeInstant gml:id="phenomenonTime" >
        < gml:timePosition >
          2012-10-12T11:15:00.000+02:00
        < /gml:timePosition >
      < /gml:TimeInstant >
    < /om:phenomenonTime >
    < om:resultTime xlink:href="#phenomenonTime" />
    < om:procedure xlink:href="http://vpn.52north.org:80/SDC/sos/rest/sensors/52" />
    < om:observedProperty
      xlink:href="http://sweet.jpl.nasa.gov/2.0/hydroSurface.owl#Discharge" />
    < om:featureOfInterest
      xlink:href="http://vpn.52north.org:80/SDC/sos/rest/features/foi_1" />
    < om:result xsi:type="gml:MeasureType" uom="urn:ogc:def:uom:OGC:m" >
      5.28
    < /om:result >
  < /om:OM_Observation >
  < sosREST:link
    rel="http://www.opengis.net/sosREST/1.0/offering-get"
    href="http://sdc.host.example.com:80/SDC/sos/rest/offerings/52"
    type="application/gml+xml" />
< /sosREST:Observation >

```

Listing 9: HTTP POST for a new observation resource

5.5.1.2 Response

```

Header:
Allow: GET, OPTIONS, POST, DELETE
Content-Type: application/gml+xml
Status: 201 Created
Location: http://sdc.host.example.com:80/SDC/sos/rest/observations/52
[...]

Body:

```

Listing 10: HTTP POST response for a new observation resource

5.5.2 Single Resource

5.5.2.1 Request

```
GET /SDC/sos/rest/observations/52 HTTP/1.1
Accept: application/gml+xml
```

Listing 11: HTTP GET for an observation resource

5.5.2.2 Response

Header:

```
Allow: GET, OPTIONS, POST, DELETE
Content-Type: application/gml+xml
Status: 200 OK
[...]
```

Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<sosREST:Observation
  [...]>
  <om:OM_Observation gml:id="o_1350398669527">
    <gml:identifier>52</gml:identifier>
    <om:type
      xlink:href="http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement"/>
    <om:phenomenonTime>
      <gml:TimeInstant gml:id="phenomenonTime_9">
        <gml:timePosition>1212-10-12T10:08:28.000+00:53:28</gml:timePosition>
      </gml:TimeInstant>
    </om:phenomenonTime>
    <om:resultTime xlink:href="#phenomenonTime_9"/>
    <om:procedure xlink:href="52"/>
    <om:observedProperty
      xlink:href="http://sweet.jpl.nasa.gov/2.0/hydroSurface.owl#Discharge"/>
    <om:featureOfInterest>
      <sams:SF_SpatialSamplingFeature gml:id="sf_2">
        <gml:identifier codeSpace="">foi_2</gml:identifier>
        <sf:type
          xlink:href="http://www.opengis.net/def/samplingFeatureType/OGC-OM/2.0/SF_SamplingPoint"/>
        <sf:sampledFeature xlink:href="foi_2"/>
        <sams:shape>
          <gml:Point gml:id="point_sf_2">
            <gml:pos
              srsName="http://www.opengis.net/def/crs/EPSSG/0/4326">
                52.0 7.5
              </gml:pos>
            </gml:Point>
          </sams:shape>
        </sams:SF_SpatialSamplingFeature>
      </om:featureOfInterest>
    <om:result uom="urn:ogc:def:uom:OGC:m" xsi:type="gml:MeasureType">5.28</om:result>
  </om:OM_Observation>
  <sosREST:link rel="http://www.opengis.net/sosREST/1.0/self"
    href="http://vpn.52north.org:80/SDC/sos/rest/observations/52" type="application/gml+xml"/>
  <sosREST:link rel="http://www.opengis.net/sosREST/1.0/observation-delete"
    href="http://vpn.52north.org:80/SDC/sos/rest/observations/52" type="application/gml+xml"/>
  <sosREST:link rel="http://www.opengis.net/sosREST/1.0/feature-get"
    href="http://vpn.52north.org:80/SDC/sos/rest/features/foi_2" type="application/gml+xml"/>
  <sosREST:link rel="http://www.opengis.net/sosREST/1.0/sensor-get"
    href="http://vpn.52north.org:80/SDC/sos/rest/sensors/52" type="application/gml+xml"/>
</sosREST:Observation>
```

Listing 12: HTTP GET response for a observation resource

5.5.3 Search

5.5.3.1 Request

```
GET /SDC/sos/rest/observations?procedure=52 HTTP/1.1
Accept: application/gml+xml
```

Listing 13: HTTP GET for observation resource search

The KVP encoded parameters in Table 3 are allowed for this HTTP method and resource combination.

Filter Property	Description ⁵
features	List of FOI resources to use as a spatial filter. Example: ?features=foi_1,foi2
observedproperties	List of observed properties to use as a filter. Example: ?observedproperties=http://sweet.jpl.nasa.gov/2.0/hydroSurface.owl%23Discharge
procedures	List of procedure IDs to use as a filter. Example: ?procedures=52
offering	List of offering Ids to use as a filter. Example: ?offering=52,53
spatialfilter	Kvp Encoded spatial filter as defined in OGC#12-006. Supported types: BBOX. Example: ?namespaces=xmlns(sams,http://www.opengis.net/samplingSpatial/2.0), xmlns(om,http://www.opengis.net/om/2.0)& spatialFilter=om:featureOfInterest/*/sams:shape,1.0,1.0,60.0,60.0,http:// www.opengis.net/def/crs/EPSSG/0/4326
temporalfilter	Kvp Encoded temporal filter as defined in OGC#12-006. Supported types: TM_EQUALS, TM_DURING. Example: ?namespaces=xmlns(om,http://www.opengis.net/om/2.0)& temporalFilter=om:phenomenonTime,2012-07-16T12:41:01.962%2B02:00 temporalFilter=om:phenomenonTime,2012-07- 16T12:41:01.962%2B02:00%2F2012-07-17T12:41:01.962%2B02:00
namespaces ⁶	Namespace definition used by spatial- and temporalfilter. Example: ?namespaces=xmlns(sams,http://www.opengis.net/samplingSpatial/2.0), xmlns(om,http://www.opengis.net/om/2.0)

Table 3: Allowed KVP parameters for the observation resource

⁵ The contained examples require URL encoding before being submitted.

⁶ Parameter namespaces is required for spatialfilter and temporalfilter. See OGC SOS 2.0.0 specification for more details ([OGC#12-006](#)).

5.5.3.2 Response

Header:

```
Allow: GET, OPTIONS, POST, DELETE
Content-Type: application/gml+xml
Status: 200 OK
[...]
```

Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<sosREST:ObservationCollection
  xmlns:sosREST="http://www.opengis.net/sosREST/1.0">
  <sosREST:Observation>
    <om:OM_Observation>
      [...]
    <sosREST:link
      rel="http://www.opengis.net/sosREST/1.0/self"
      href="http://sdc.host.example.com:80/SDC/sos/rest/observations/1"
      type="application/gml+xml"/>
    </sosREST:Observation>
  <sosREST:Observation>
    <om:OM_Observation>
      [...]
    <sosREST:link
      rel="http://www.opengis.net/sosREST/1.0/self"
      href="http://sdc.host.example.com:80/SDC/sos/rest/observations/2"
      type="application/gml+xml"/>
    </sosREST:Observation>
  [...]
  <sosREST:link
    rel="http://www.opengis.net/sosREST/1.0/self"
    href="http://sdc.host.example.com:80/SDC/sos/rest/observations?procedure=52"
    type="application/gml+xml"/>
</sosREST:ObservationCollection>
```

Listing 14: HTTP GET response for observation resource search

5.5.4 Delete

5.5.4.1 Request

```
DELETE /SDC/sos/rest/observations/2 HTTP/1.1  
Accept: application/gml+xml
```

Listing 15: HTTP DELETE for an observation resource

5.5.4.2 Response

```
Header:  
Allow: GET, OPTIONS, POST, DELETE  
Content-Type: application/gml+xml  
Status: 204 OK - No content  
X-Deleted-Resource-Id: 2  
[...]
```

Listing 16: HTTP DELETE response for an observation resource

5.6 Offerings

5.6.1 Resource Collection

5.6.1.1 Request

```
GET /SDC/sos/rest/offerings HTTP/1.1
Accept: application/gml+xml
```

Listing 17: HTTP GET request for all offerings

5.6.1.2 Response

Header:

```
Allow: GET, OPTIONS
Content-Type: application/gml+xml
Status: 200 OK
```

Body:

```
<?xml version="1.0" encoding="UTF-8"?>
< sosREST:OfferingCollection xmlns:sosREST="http://www.opengis.net/sosREST/1.0">
  < sosREST:link rel="http://www.opengis.net/sosREST/1.0/self"
href="http://sdc.host.example.com:80//SDC/sos/rest/offerings" type="application/gml+xml"/>
  < sosREST:link rel="http://www.opengis.net/sosREST/1.0/offering-get"
href="http://sdc.host.example.com:80//SDC/sos/rest/offerings/1003"
type="application/gml+xml"/>
  < sosREST:link rel="http://www.opengis.net/sosREST/1.0/offering-get"
href="http://sdc.host.example.com:80//SDC/sos/rest/offerings/1003"
type="application/gml+xml"/>
  [...]
</ sosREST:OfferingCollection >
```

Listing 18: HTTP GET response for all offerings in an OfferingCollection

5.6.2 Single Offering

5.6.2.1 Request

```
GET /SDC/sos/rest/offerings/1003 HTTP/1.1
Accept: application/gml+xml
```

Listing 19: HTTP GET request for one offering resource

5.6.2.2 Response

Header:

```
Allow: GET, OPTIONS
Content-Type: application/gml+xml
Status: 200 OK
```

Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<sosREST:ObservationOffering xmlns:sosREST="http://www.opengis.net/sosREST/1.0">
  <sos:ObservationOffering [...]>
    <swes:identifier>1000</swes:identifier>
    <swes:name>Offering for Sensor 1000</swes:name>
    <swes:procedure>1000</swes:procedure>

    <swes:procedureDescriptionFormat>http://www.opengis.net/sensorML/1.0.1</swes:procedureDescriptionFormat>

    <swes:observableProperty>http://sweet.jpl.nasa.gov/2.0/hydroSurface.owl#Discharge</swes:observableProperty>
    <sos:observedArea>
      <gml:Envelope srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
        <gml:lowerCorner>-90.1655 -178.3457</gml:lowerCorner>
        <gml:upperCorner>-8.5197 -22.7481</gml:upperCorner>
      </gml:Envelope>
    </sos:observedArea>
    <sos:phenomenonTime>
      <gml:TimePeriod gml:id="phenomenonTime_10">
        <gml:beginPosition>1271-01-01T12:53:28.000+00:53:28</gml:beginPosition>
        <gml:endPosition>1970-01-01T01:00:00.001+01:00</gml:endPosition>
      </gml:TimePeriod>
    </sos:phenomenonTime>
    <sos:responseFormat>application/zip</sos:responseFormat>
    <sos:responseFormat>http://www.opengis.net/om/2.0</sos:responseFormat>
    <sos:observationType>http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement</sos:observationType>
    <sos:featureOfInterestType>http://www.opengis.net/def/samplingFeatureType/OGC-OM/2.0/SF_SamplingSurface</sos:featureOfInterestType>
    <sos:featureOfInterestType>http://www.opengis.net/def/samplingFeatureType/OGC-OM/2.0/SF_SamplingCurve</sos:featureOfInterestType>
    <sos:featureOfInterestType>http://www.opengis.net/def/samplingFeatureType/OGC-OM/2.0/SF_SamplingPoint</sos:featureOfInterestType>
  </sos:ObservationOffering>
  <sosREST:link rel="http://www.opengis.net/sosREST/1.0/self"
href="http://sdc.host.example.com:80/SDC/sos/rest/offerings/1000"
type="application/gml+xml"/>
  <sosREST:link rel="http://www.opengis.net/sosREST/1.0/sensor-get" href=
"http://vpn.52north.org:80/SDC/sos/rest/sensors/1003" type="application/gmx+xml"/>
</sosREST:ObservationOffering>
```

Listing 20: HTTP GET response for one offering resource

5.7 Sensors

5.7.1 Resource Collection

5.7.1.1 Request

```
GET /SDC/sos/rest/sensors HTTP/1.1
Accept: application/gml+xml
```

Listing 21: HTTP GET request for all sensors

5.7.1.2 Response

Header:

```
Allow: GET, OPTIONS, POST, PUT, DELETE
Content-Type: application/gml+xml
Status: 200 OK
```

Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<sosREST:SensorCollection xmlns:sosREST="http://www.opengis.net/sosREST/1.0">
  <sosREST:link rel="http://www.opengis.net/sosREST/1.0/self"
href="http://sdc.host.example.com:80/SDC/sos/rest/sensors" type="application/gml+xml"/>
  <sosREST:link rel="http://www.opengis.net/sosREST/1.0/sensor-get"
href="http://sdc.host.example.com:80/SDC/sos/rest/sensors/1000"
type="application/gml+xml"/>
  <sosREST:link rel="http://www.opengis.net/sosREST/1.0/sensor-get"
href="http://sdc.host.example.com:80/SDC/sos/rest/sensors/1001"
type="application/gml+xml"/>
  [...]
</sosREST:SensorCollection>
```

Listing 22: HTTP GET response for all sensors in an SensorCollection

5.7.2 Single Resource

5.7.2.1 Request

```
GET /SDC/sos/rest/sensors/1000 HTTP/1.1
Accept: application/gml+xml
```

Listing 23: HTTP GET request for all sensors

5.7.2.2 Response

Header:

```
Allow: GET, OPTIONS, POST, PUT, DELETE
Content-Type: application/gml+xml
Status: 200 OK
```

Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<sosREST:Sensor xmlns:sosREST="http://www.opengis.net/sosREST/1.0">
  <sml:System >
    [...]
  </sml:System>
  <sosREST:link rel="http://www.opengis.net/sosREST/1.0/self"
href="http://vpn.52north.org:80/SDC/sos/rest/sensors/1000" type="application/gml+xml"/>
<sosREST:link rel="http://www.opengis.net/sosREST/1.0/sensor-delete"
href="http://vpn.52north.org:80/SDC/sos/rest/sensors/1000" type="application/gml+xml"/>
<sosREST:link rel="http://www.opengis.net/sosREST/1.0/sensor-update"
href="http://vpn.52north.org:80/SDC/sos/rest/sensors/1000" type="application/gml+xml"/>
<sosREST:link rel="http://www.opengis.net/sosREST/1.0/feature-get"
href="http://vpn.52north.org:80/SDC/sos/rest/features?procedures=1000"
type="application/gml+xml"/>
<sosREST:link rel="http://www.opengis.net/sosREST/1.0/observation-get"
href="http://vpn.52north.org:80/SDC/sos/rest/observations?procedures=1000"
type="application/gml+xml"/>
<sosREST:link rel="http://www.opengis.net/sosREST/1.0/property-get"
href="http://sweet.jpl.nasa.gov/2.0/hydroSurface.owl#Discharge"
type="application/rdf+xml"/>
</sosREST:Sensor>
```

Listing 24: HTTP GET response for all sensors in an SensorCollection

5.7.3 Create

5.7.3.1 Request

```

POST /SDC/sos/rest/sensors HTTP/1.1
Accept: application/gml+xml

<?xml version="1.0" encoding="UTF-8"?>
<sosREST:Sensor[...]>
  <sml:System>
    <sml:identification>
      <sml:IdentifierList>
        <sml:identifier>
          <sml:Term definition="urn:ogc:def:identifier:OGC:uniqueID">
            <sml:value>1000</sml:value>
          </sml:Term>
        </sml:identifier>
      </sml:IdentifierList>
    </sml:identification>
    <sml:identification xlink:title="offerings">
      <sml:IdentifierList>
        <sml:identifier name="Offering for Sensor 1000">
          <sml:Term definition="urn:ogc:def:identifier:OGC:offeringID">
            <sml:value>1000</sml:value>
          </sml:Term>
        </sml:identifier>
      </sml:IdentifierList>
    </sml:identification>
    <sml:capabilities name="InsertionMetadata">
      <swe:SimpleDataRecord>
        <swe:field name="sos:ObservationType">
          <swe:Text>
            <swe:value>http://www.opengis.net/def/observationType/OGC-
OM/2.0/OM_Measurement</swe:value>
          </swe:Text>
        </swe:field>
        <swe:field name="sos:FeatureOfInterestType">
          <swe:Text>
            <swe:value>http://www.opengis.net/def/samplingFeatureType/OGC-
OM/2.0/SF_SamplingPoint</swe:value>
          </swe:Text>
        </swe:field>
      </swe:SimpleDataRecord>
    </sml:capabilities>
    [...]
    <sml:inputs>
      <sml:InputList>
        <sml:input name="discharge" xlink:title="application/rdf+xml">
          <swe:ObservableProperty
definition="http://sweet.jpl.nasa.gov/2.0/hydroSurface.owl#Discharge"/>
        </sml:input>
      </sml:InputList>
    </sml:inputs>
    [...]
  </sml:System>
</sosREST:Sensor>

```

Listing 25: HTTP POST request for one sensor resource

5.7.3.2 Response

Header:

Allow: GET, OPTIONS, POST, PUT, DELETE
Content-Type: application/gml+xml
Status: 200 OK

Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<sosREST:Sensor xmlns:sosREST="http://www.opengis.net/sosREST/1.0">
  <sm1:System >
    [...]
    <sosREST:link rel="http://www.opengis.net/sosREST/1.0/self"
href="http://vpn.52north.org:80/SDC/sos/rest/sensors/1000" type="application/gml+xml"/>
    <sosREST:link rel="http://www.opengis.net/sosREST/1.0/sensor-delete"
href="http://vpn.52north.org:80/SDC/sos/rest/sensors/1000" type="application/gml+xml"/>
    <sosREST:link rel="http://www.opengis.net/sosREST/1.0/sensor-update"
href="http://vpn.52north.org:80/SDC/sos/rest/sensors/1000" type="application/gml+xml"/>
    <sosREST:link rel="http://www.opengis.net/sosREST/1.0/feature-get"
href="http://vpn.52north.org:80/SDC/sos/rest/features?procedures=1000"
type="application/gml+xml"/>
    <sosREST:link rel="http://www.opengis.net/sosREST/1.0/observation-get"
href="http://vpn.52north.org:80/SDC/sos/rest/observations?procedures=1000"
type="application/gml+xml"/>
    <sosREST:link rel="http://www.opengis.net/sosREST/1.0/property-get"
href="http://sweet.jpl.nasa.gov/2.0/hydroSurface.owl#Discharge"
type="application/rdf+xml"/>
  </sosREST:Sensor>
```

Listing 26: HTTP POST response for one sensor resource

5.7.4 Update

5.7.4.1 Request

```
PUT /SDC/sos/rest/sensors/1000 HTTP/1.1
Accept: application/gml+xml

<?xml version="1.0" encoding="UTF-8"?>
<sosREST:Sensor[...]>
  <sm1:System>
    [...]
  </sm1:System>
</sosREST:Sensor>
```

Listing 27: HTTP PUT request for one sensor resource

5.7.4.2 Response

Header:

```
Allow: GET, OPTIONS, POST, PUT, DELETE
Content-Type: application/gml+xml
Status: 200 OK
Location: http://sdc.host.example.com:80/SDC/sos/rest/sensors/1052
```

Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<sosREST:Sensor xmlns:sosREST="http://www.opengis.net/sosREST/1.0">
  <sm1:System >
    [...]
    <sosREST:link rel="http://www.opengis.net/sosREST/1.0/self"
href="http://vpn.52north.org:80/SDC/sos/rest/sensors/1000" type="application/gml+xml"/>
    <sosREST:link rel="http://www.opengis.net/sosREST/1.0/sensor-delete"
href="http://vpn.52north.org:80/SDC/sos/rest/sensors/1000" type="application/gml+xml"/>
    <sosREST:link rel="http://www.opengis.net/sosREST/1.0/sensor-update"
href="http://vpn.52north.org:80/SDC/sos/rest/sensors/1000" type="application/gml+xml"/>
    <sosREST:link rel="http://www.opengis.net/sosREST/1.0/feature-get"
href="http://vpn.52north.org:80/SDC/sos/rest/features?procedures=1000"
type="application/gml+xml"/>
    <sosREST:link rel="http://www.opengis.net/sosREST/1.0/observation-get"
href="http://vpn.52north.org:80/SDC/sos/rest/observations?procedures=1000"
type="application/gml+xml"/>
    <sosREST:link rel="http://www.opengis.net/sosREST/1.0/property-get"
href="http://sweet.jpl.nasa.gov/2.0/hydroSurface.owl#Discharge"
type="application/rdf+xml"/>
  </sosREST:Sensor>
```

Listing 28: HTTP PUT response for one sensor resource

6 Developer Guide

Setting up Development Environment

- 1 Check out branch 52n-sos-400-refactored from <https://svn.52north.org/svn/swe/main/SOS/Service/branches/>
- 2 Adjust the settings in the `conf/datasource.properties` file to match the set-up in the development environment
- 3 Adjust the root pom property `$conf.sos.name` if required.
- 4 Use the following profiles for the minimum set-up (incl. other bindings like KVP and SOAP): `develop`, `configure-datasource`, `use-default-settings`, `do`, `restful` when building the application.

Figure 4 shows the package explorer of Eclipse IDE with all three maven projects and imported maven sub modules.

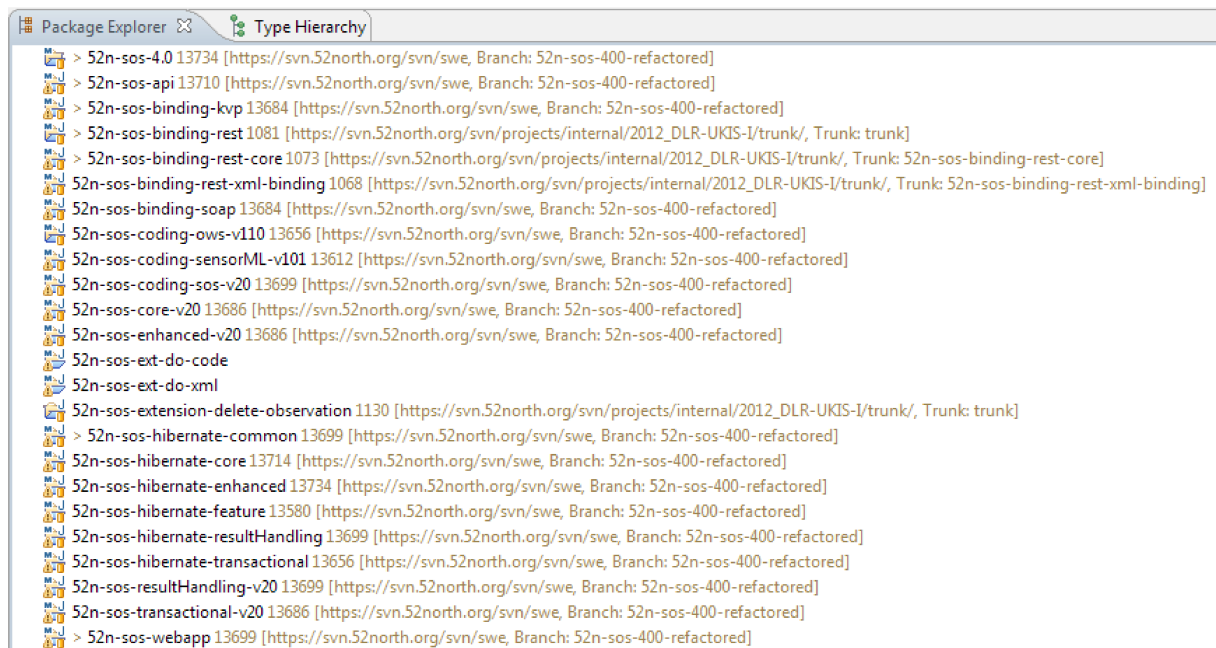


Figure 4: Example workspace layout with all maven modules imported in Eclipse IDE

7 Appendix

7.1 HTTP Response Codes

See google doc:

<https://docs.google.com/spreadsheet/ccc?key=0AgYaiXbFLY79dEE2VWRBblNoNUM2dFZaULJwaWitWnc>

Still, not all exceptions are caught and encoded with HTTP 500 (e.g. NullPointerExceptions).

7.2 Troubleshooting

In the case of having any problems, please take a look at the regarding bugzilla section:

https://bugzilla.52north.org/buglist.cgi?product=sensorweb_DLR-UKIS-!&component=Sensor%20Data%20Center

7.3 Known Issues

7.3.1 Sensor Description

Accepted SensorML has to comply to a profile:

- Version: 1.0.1
- Types: System
- Required Elements:
 - `<sml:identification><sml:IdentifierList><sml:identifier>
<sml:Term definition="urn:ogc:def:identifier:OGC:uniqueID">
<sml:value> containing the sensor id.`
 - `<sml:identification
xlink:title="offerings"><sml:IdentifierList>
<sml:identifier name="$OFFERING_DESCRIPTION"><sml:Term
definition="urn:ogc:def:identifier:OGC:offeringID"><sml:value>
containing the sensor id.`
 - `<sml:capabilities name="InsertionMetadata">
 <swe:SimpleDataRecord>
 <swe:field name="sos:ObservationType">7
 <swe:Text>
 <swe:value>$OBSERVATION_TYPE</swe:value>
 </swe:Text>
 </swe:field>
 <swe:field name="sos:FeatureOfInterestType">8
 <swe:Text>
 <swe:value>$FEATURE_TYPE</swe:value>
 [...]`
 - The observed property of the observations MUST be contained as

⁷ As often as required. See Section 7.3.2 for currently supported types.

⁸ As often as required. Supported types are:

- http://www.opengis.net/def/samplingFeatureType/OGC-OM/2.0/SF_SamplingPoint
- http://www.opengis.net/def/samplingFeatureType/OGC-OM/2.0/SF_SamplingCurve
- http://www.opengis.net/def/samplingFeatureType/OGC-OM/2.0/SF_SamplingSurface

```
<sml:input name="$UNUSED_NAME" xlink:title="$CONTENT_TYPE"9>  
  <swe:ObservableProperty definition="$OBSERVABLE_PROPERTY"/>  
</sml:input> of the sensor description.
```

7.3.2 Observation Encoding

- Version: 2.0
- Number of Observations for each HTTP POST request: 1
- Supported Observation types:
 - http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement
 - http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_CategoryObservation
 - http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_SWEArrayObservation¹⁰
 - http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_CountObservation
 - http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_TruthObservation
 - http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_TextObservation

9 If this element is not present, the implementation will use a default value defined by `default.content.type.undefined`.

10 The version supports currently one observed property per `SWEArrayObservation` and 1 dimensional `swe:DataRecords` as result structure.