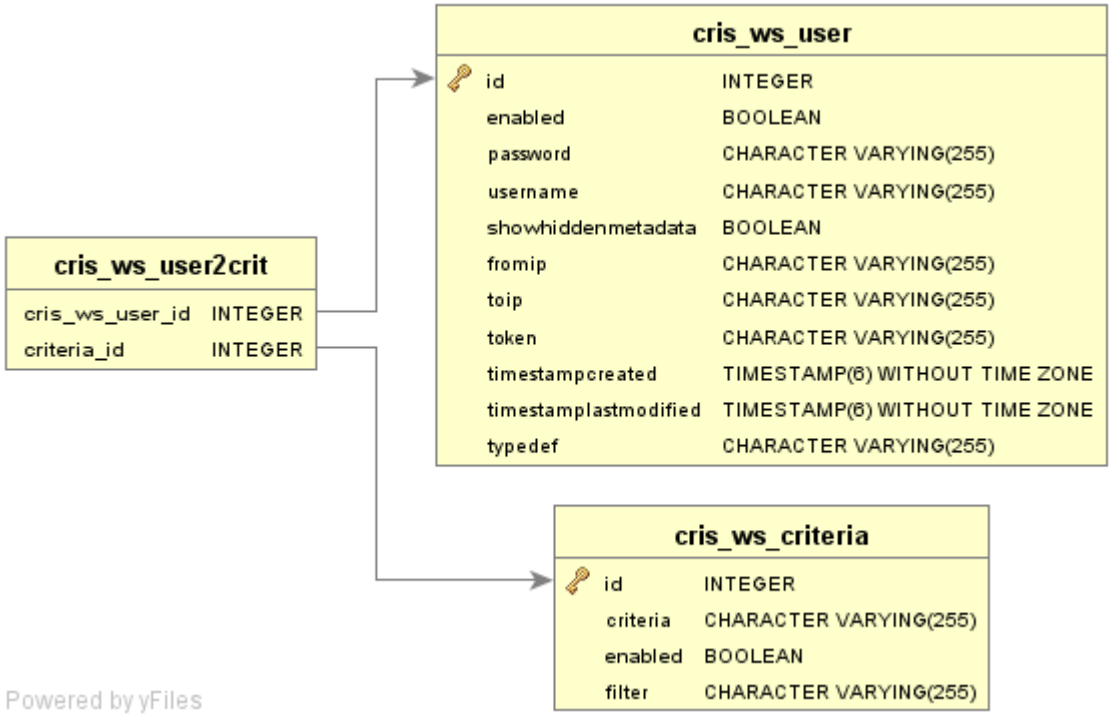
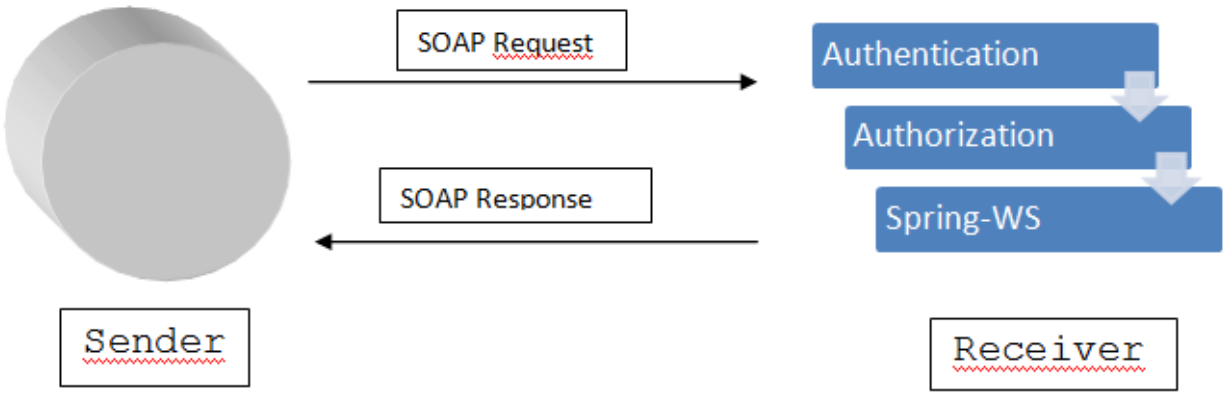
SOAP WebServices

Database reference

The following E-R diagram shows the SOAP webservice configuration tables that allow to access to DSpace-CRIS data: accounts, access methods, filters, etc.



Architecture



The “Authentication” and “Authorization” customized modules allow the administrator to manage access information.

Spring Web Services (Spring-WS) is provided by Spring community that is focused on creating document-driven Web services[1].

WSDL

A WSDL file is an XML document that describes a Web service. It specifies the location of the service and the operations (or methods) the service exposes. When creating Web services, there are two development styles: *Contract Last* and *Contract First*.

It was used Contract First because in Spring-WS it allows to start with WSDL definitions followed by Java code. Note that in Spring-WS, *writing the WSDL by hand is not required*. Based on the XSD and some conventions, Spring-WS may create the WSDL by itself.

Fragment of **spring-ws-servlet.xml:**

```
<bean id="wsdlforwsservices" class="…DynamicWsdl11Definition">
    <property name="builder">
            <bean class="…XsdBasedSoap11Wsdl4jDefinitionBuilder">
                            <property name="schema" value="<WEBINF>/crisrequestforwsdl.xsd" />
                <property name="portTypeName" value="cris" />
                <property name="locationUri" value="${dspace.url}/webservices/" />
                <property name="targetNamespace"
                          value="http://www.cilea.it/cris/definitions" />
                   </bean>
    </property>
</bean>
```

Point browser or a SOAP client like SoapUI (download from here) to http://<dspace-webapp>/webservices/wsdlforwsservices.wsdl

Spring-WS can generate WSDL for us based on XML Schema called **crisrequestforwsdl.xsd**. [m1]

Others XML Schema imported on XML Schema for WSDL called *requestresearcherpage.xsd* and *responseresearcherpage.xsd*, *requestresearchergrants.xsd* and *responseresearchergrant.xsd*, *staticrequestpublications.xsd* and *staticresponsepublicatins.xsd* are not only your service contract but also are your data contract.[m2]

Data contract (with **request** prefix on xsd file name) defines the message format to accept and the objects type to return (with **response** prefix).

In fact, at begin of WSDL, you can find import of those xsd, this is the contract for request and response. [m3] Each *simpleType* of request contract defines object and field types to respond.

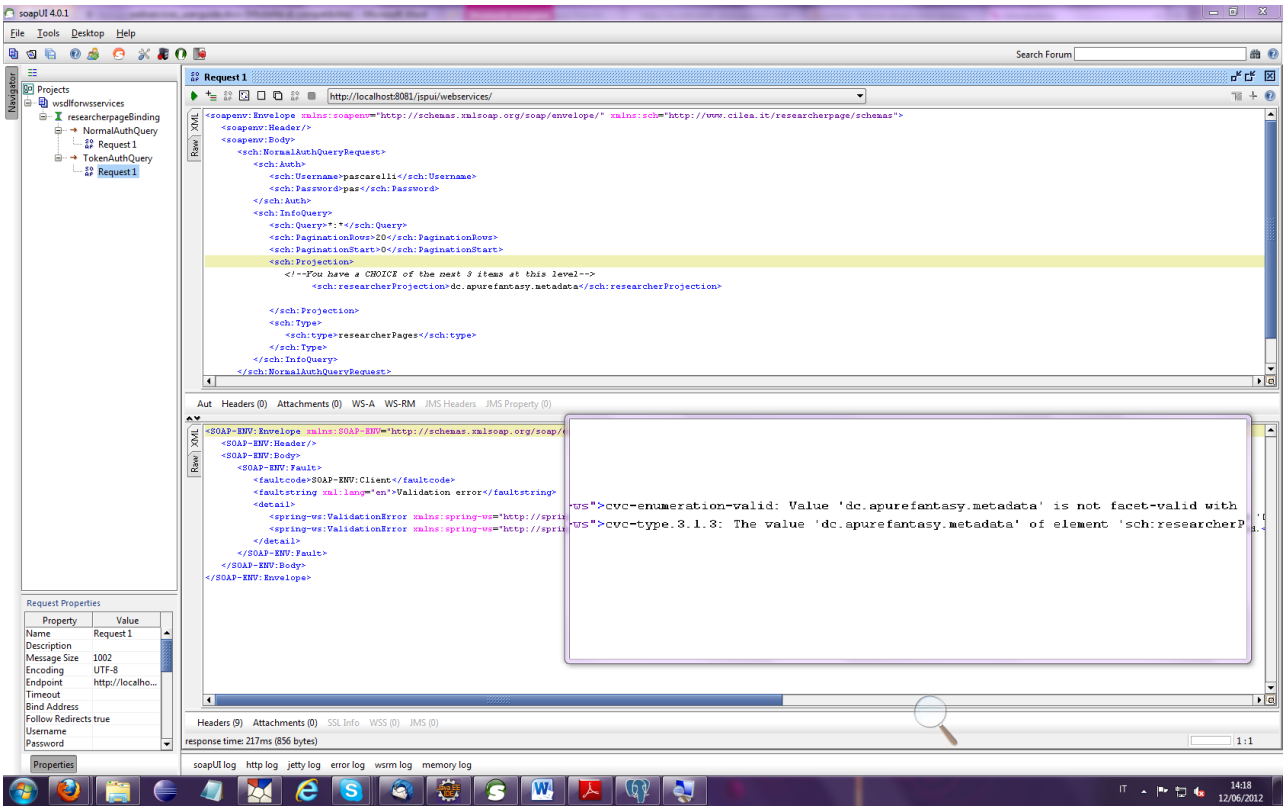
There are two schema related to WSDL:

- **crisrequestforwsdl.xsd** in the classpath is used to generate public WSDL;
- **crisrequest.xsd** in the external folder is used to validate data request.

Note that **crisrequestforwsdl.xsd** and **crisrequest.xsd** have the same contents, they differ only by the initial imports.

```
<xs:simpleType name="ValueListPublications">
        <xs:restriction base="xs:string">
              <xs:enumeration value="search.resourceid" />
              <xs:enumeration value="dc.type" />
              <xs:enumeration value="dc.title" />
              ...
              <xs:enumeration value="dc.contributor.author" />
        </xs:restriction>
    </xs:simpleType>
```

Validation is done on request call based on xsd. Note that for entities in the system the xsd like for RP entity or GRANT entity will generate at system startup and put them in **ws-xsd** (configurated on cris.cfg with key *webservices.xsd.path*) directory on dspace local directory. The only one to copy onto directory by hand is request contract and response contract for publications (you can find these in **WEB-INF\classes** directory onto webapp).



The service endpoints are managed by **PayloadRootQNameEndpointMapping** class provided by Spring Framework.

This class maps SOAP request

"{http://www.cilea.it/cris/schemas}NormalAuthQueryRequest"

to

rpNormalEndpoint

bean service and

"{http://www.cilea.it/cris/schemas}TokenAuthQueryRequest"

to **rpTokenEndpoint** bean service.

rpNormalEndpoint and rpTokenEndpoint first authenticate and authorize user, then manage request and build response to fulfill data contract.

Web Service User Management

The administration menu shows the “Web services” item through which the web service users may be managed.

It allows to create two kind of user authentication:

- username/password
- with a token and a range or single IP

It also allows to:

- enable or disable users
- enable or disable permissions
- filter criteria (SOLR-like)
- show values

Objects managed appear in dspace.cfg:

```
##Web Services Configuration
webservices.criteria.type = publications, researcherPages, grants
webservices.criteria.type.publications = 2
webservices.criteria.type.researcherPages  = 9
webservices.criteria.type.grants  = 10
```

Note: to add new entity it is mandatory add new type in dspace.cfg and also manage XML Schema definition.

Web Service Test

SOAP UI allow to contact web services. Followed it is the SOAP XML document aka SOAP envelope[m6] . For further information see the code fragment comments.

Follows a token authentication request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://www.cilea.it
/researcherpage/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:TokenAuthQueryRequest>
         <sch:Auth>
              <!-- the token -->
            <sch:Token>xsbcsdsdfsf-awe12-asdad</sch:Token>
         </sch:Auth>
         <sch:InfoQuery>
              <!-- the query -->
            <sch:Query>*:*</sch:Query>
              <!-- pagination -->
            <sch:PaginationRows>20</sch:PaginationRows>
            <sch:PaginationStart>0</sch:PaginationStart>
            <sch:Projection>
               <!--You have a CHOICE of the next 3 items at this level-->
               <!-- CHOOSE only one and take the consistent type -->
               <!—if leave empty then the result depends on user profile->
               <sch:grantProjection>search.resourceid grant_keywords</sch:grantProjection>
            </sch:Projection>
            <sch:Type>
                 <!-- publication,researcherPages,grants are allowed -->
               <sch:type>grants</sch:type>
            </sch:Type>
         </sch:InfoQuery>
      </sch:TokenAuthQueryRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Server side, the token, get by SOAP messages, and the caller address IP, get by transport service aka HTTP, are validated. If validation is true then a query SOLR is builded in the follow manner:[m7]

- Getting query from SOAP message
- Getting type and building a query (es. fq=search.resourcetype:2 where 2 stands for Item, 9 for RP and 10 for Grants)
- Getting in the from User profile other queries

- Shoot query on SOLR server (reminder: pagination is active)[m8]

Follows a sample response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <cris:TokenAuthQueryResponse hit="15206" rows="1" start="0" type="grants" xmlns:cris="http://www.cilea.it/cris/schemas">
      <grant:crisobjects xmlns:grant="http://www.cilea.it/grant/schemas">
        <grant:crisobject businessID="4" publicID="1" type="10" uuid="f5c01ae7-0921-4b9b-be84-
c66bacaa6011                                                                                    ">
          <grant:fundingyear visibility="1">1995/1996</grant:fundingyear>
          <grant:amount visibility="1">12000</grant:amount>
          <grant:startdate visibility="1">01-08-1995</grant:startdate>
          <grant:extdate visibility="1">31-07-1996</grant:extdate>
          <grant:projecttitle visibility="1">Biomechanical evaluation of the human sacrum for lumbosacral fixation</grant:projecttitle>
          <grant:keywords visibility="1">orthopaedic</grant:keywords>
          <grant:discipline visibility="1">Orthopaedics/Traumatology</grant:discipline>
          <grant:granttype visibility="1">Departmental Budget - General Award</grant:granttype>
          <grant:status visibility="1">Completed</grant:status>
          <grant:investigator visibility="1">Professor Leong John Chi Yan</grant:investigator>
          <grant:coInvestigators>
            <grant:coInvestigator rpkey="411">Lu, William Weijia</grant:coInvestigator>
            <grant:coInvestigator>Professor Zhong</grant:coInvestigator>
            <grant:coInvestigator>Dr Zhu</grant:coInvestigator>
          </grant:coInvestigators>
        </grant:crisobject>
      </grant:crisobjects>
    </cris:TokenAuthQueryResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

---

[1] Spring Web Services. http://static.springsource.org/spring-ws/sites/1.0/reference/html/index.html