GUIDELINES FOR TCS Metadata Mapping

This guide has been created to support TCS Metadata Mapping in the following three steps:

1. Create the XML file
2. Validate the XML file
3. Upload the XML file

1. Create the XML file

In order to facilitate the process of mapping we provide on GitHub:

a) An example XML (compliant to EPOS-DCAT-AP.xsd) available on https://github.com/epos-eu/EPOS-DCAT-AP/blob/master/examples/EPOS-DCAT-AP_example.xml.

b) An Excel file https://github.com/epos-eu/EPOS-DCAT-AP/blob/master/docs/EPOS_DCAT-AP_Vocabulary_and_Specification.xlsx which contains the definition of the EPOS-DCAT-AP terms and their cardinalities.

In the example a) all EPOS entities are already filled. In this phase (before September) is necessary to fill only **WebService**, **Person** and **Organisation**. The others entities (e.g., Catalog, Publication, Project, etc.) can be removed.

For each entity you should:

- Use unique identifier (e.g., Person → ORCID, Organization → PIC, WebService → EndPoint)
- Link properly the entities between them, using their identifiers as shown in the figure 1
- Fill all mandatory tags (check the cardinalities using the excel file)
- Remove the optional tags if you don't have data to put in (don't leave them empty)

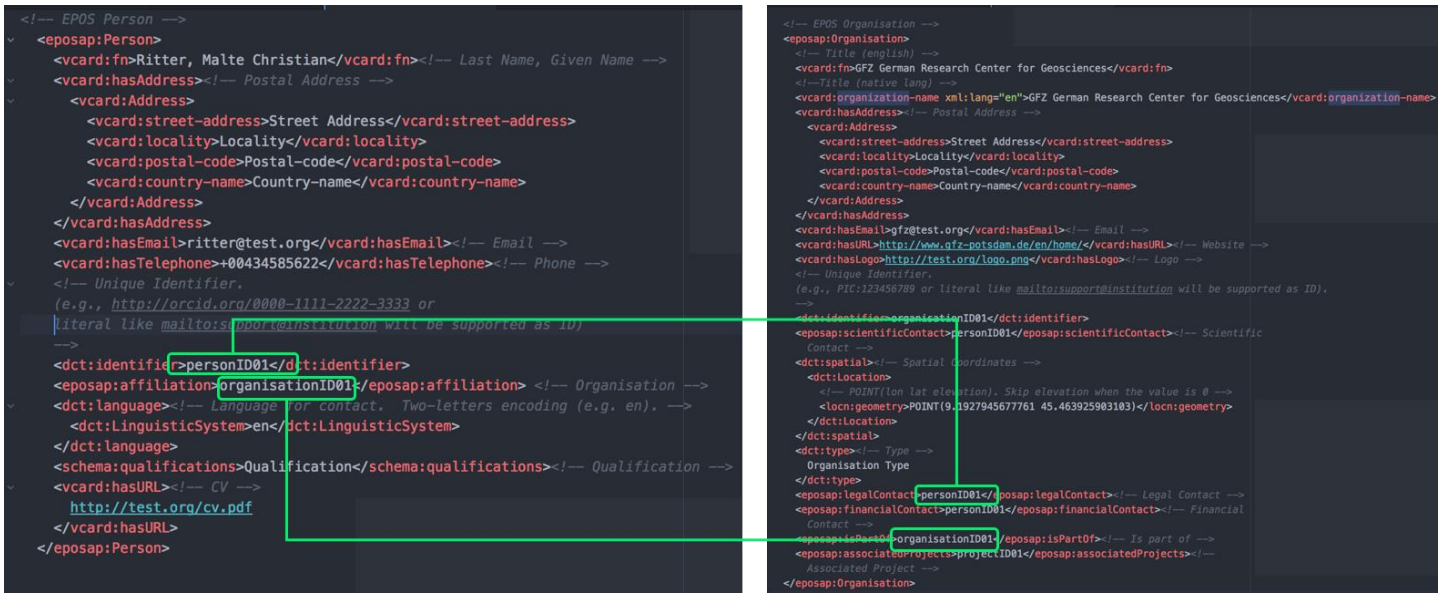


Figure 1 - How to link Person and Organisation entities

Please, note that the **WebService** is a very important entity because it will represent the main resource of the demonstrator. So you have to put attention to fill in the right way the green highlighted tags as in the figure below.

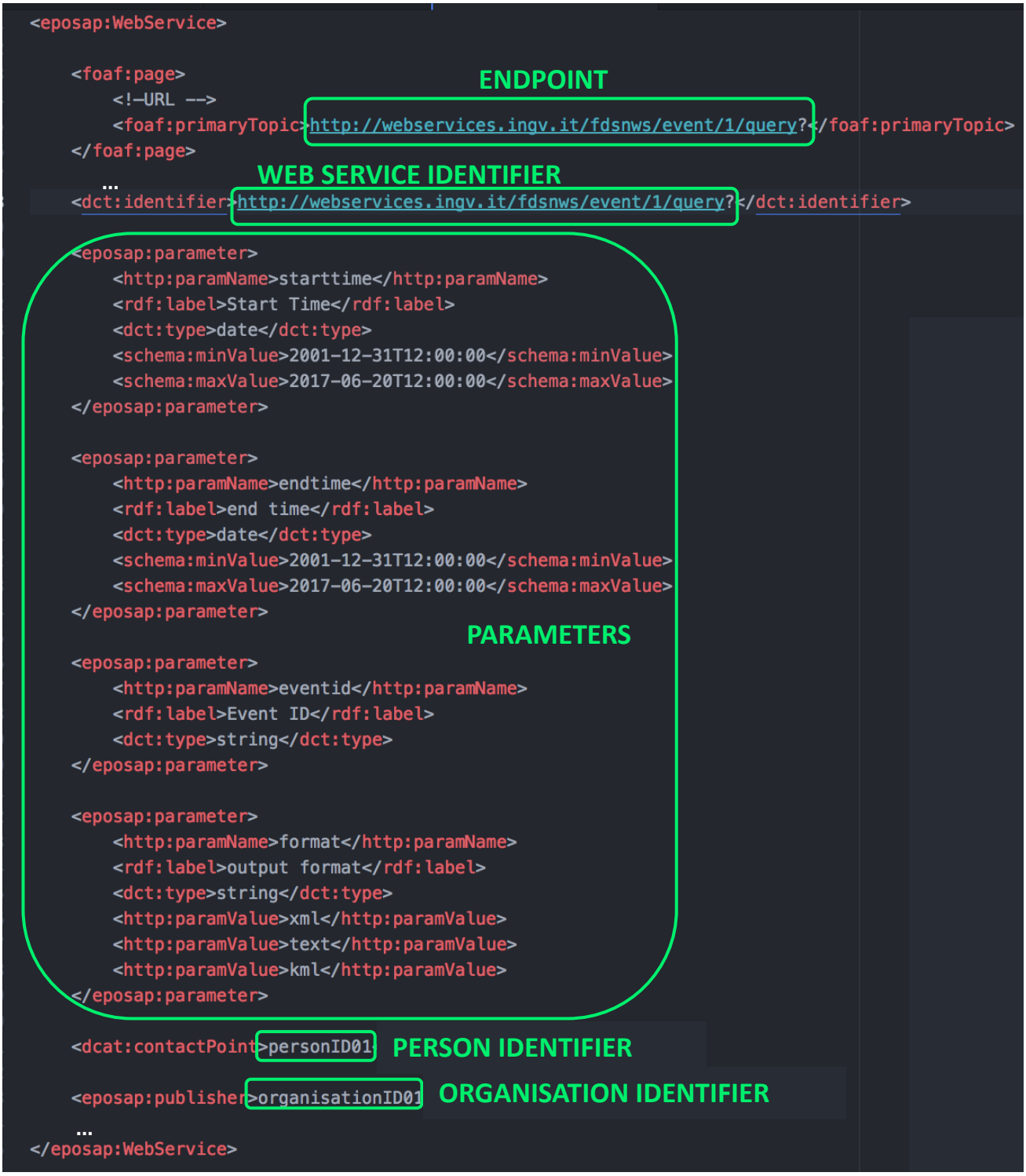


Figure 2 – Significant Web Service fields

- **ENDPOINT** (foaf:primaryTopic): this field contains the URL of the web service without parameters. This means that this URL coupled with the parameters, allows to query the web service. Note that the web service must have **no authentication or authorization (until validation phase)**.
- **WEB SERVICE IDENTIFIER** (dct:identifier): this field contains the unique identifier of the web service (e.g., base URL).
- **PERSON IDENTIFIER** (dct:contactPoint): this field contains the identifier used to link to the Person which represents the Contact Point
- **ORGANISATION IDENTIFIER** (eposap:publisher): this field contains the identifier used to link to the Organisation responsible for making the Web Service available.
- **PARAMETER** (eposap:parameter): this tag allows you to specify all the information needed to query the web service.
  It contains the following fields:

XML TAG	DESCRIPTION
http:paramName	The name of the parameter as required by web service specifications

rdf:label	The label is a short string used to describe the meaning of the parameter to the GUI user.
	(*e.g.*,
	`<http:paramName>starttime</http:paramName>`
	`<rdf:label>Start Time</rdf:label>`
	)

dct:type	The type of the parameter (*i.e.*, *string*, *integer*, *float*, *boolean*, *date*)

http:paramValue	This field represents one of the possible value which should be used in the web service query; it could be repeated as many times as needed.
	(e.g.,
	`<http:paramValue>xml</http:paramValue>`
	`<http:paramValue>text</http:paramValue>`
	`<http:paramValue>kml</http:paramValue>`
	This means that this parameter could assume only these values).

schema:minValue	The minimum value of the parameter
schema:maxValue	The maximum value of the parameter
owl:versionInfo	The version of the parameter

In order to check if your web service is mapped in right way we suggest to compose the entry Point URL with parameters written in your XML file and copy full URL in a browser to test the output expected (as shown in figure 3).
Below, some possible full URLs to check the right mapping of the example Web Service:

- http://webservices.ingv.it/fdsnws/event/1/query?starttime=2012-05-29T00:00:00&endtime=2012-05-29T23:59:59
- http://webservices.ingv.it/fdsnws/event/1/query?starttime=2012-05-29T00:00:00&endtime=2012-05-29T23:59:59&eventID=863301&format=text

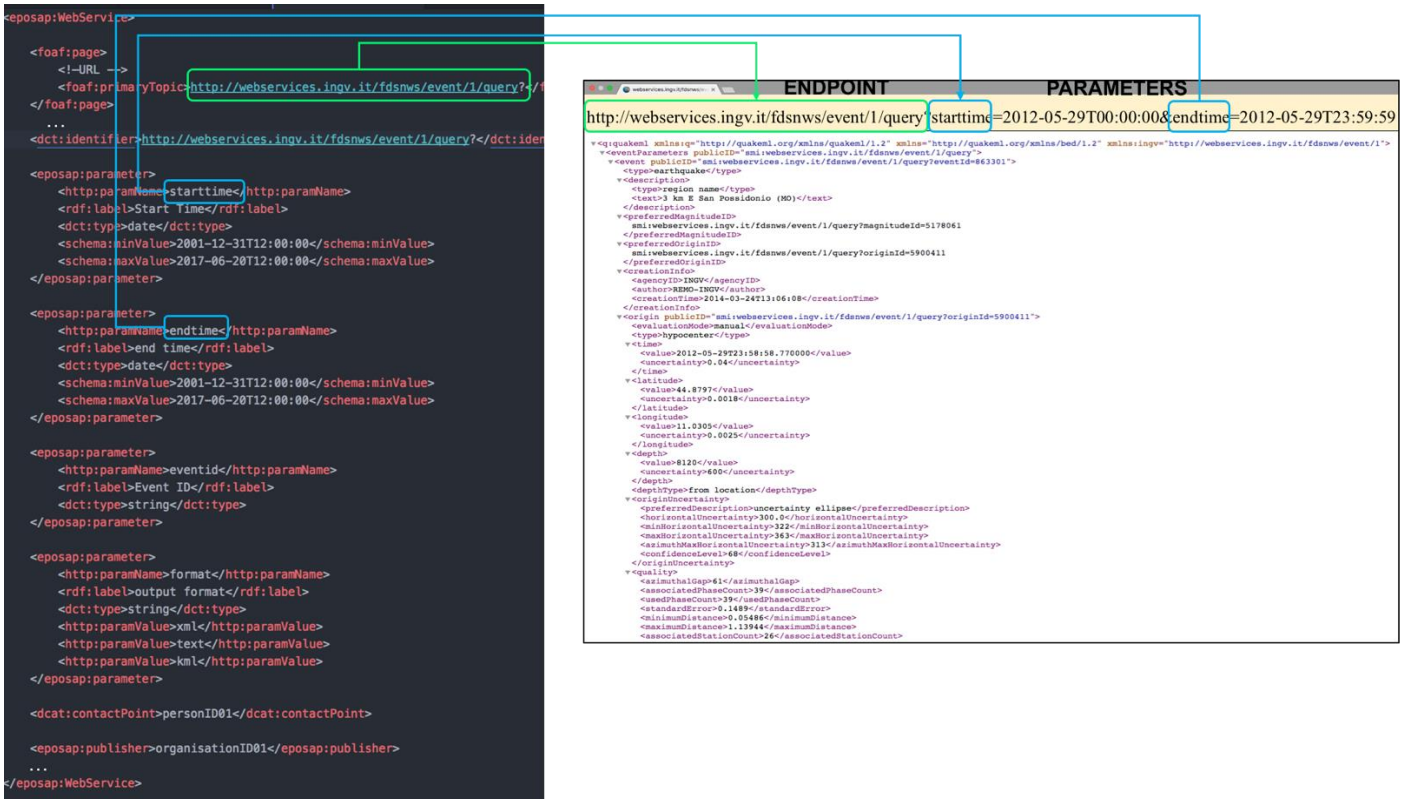


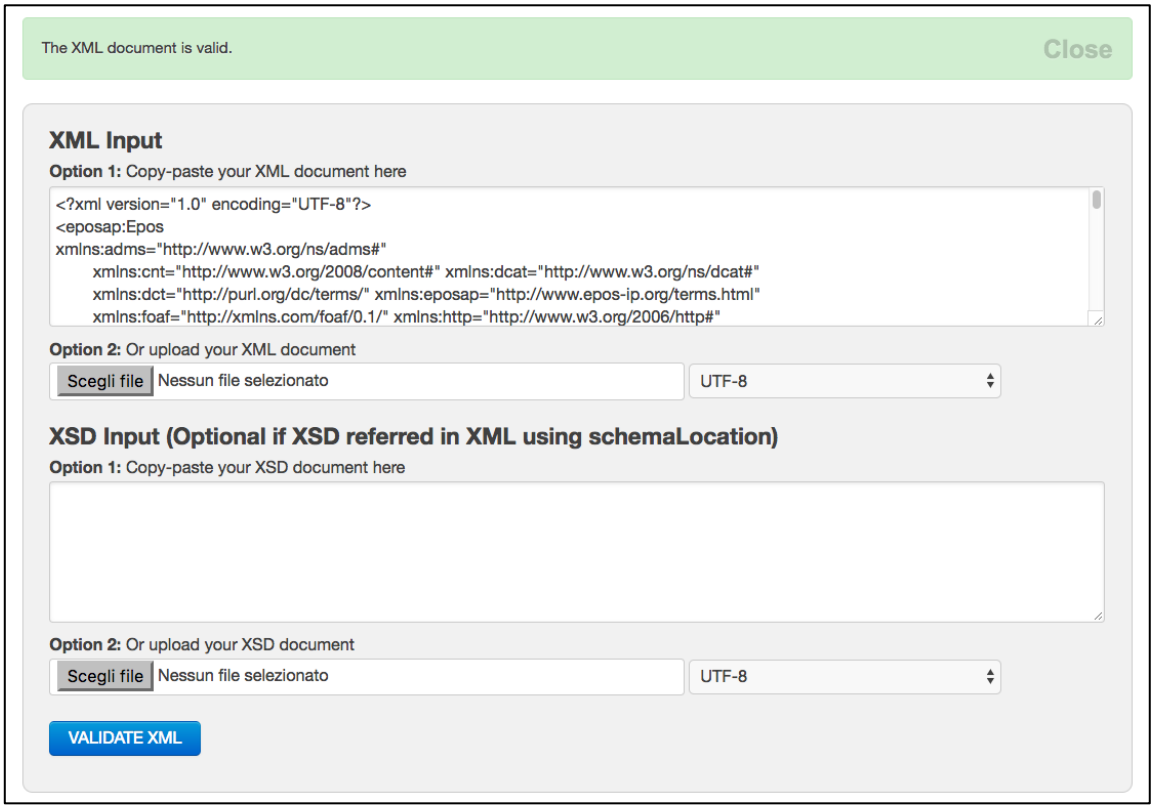
Figura 3 – Check the WebService mapping

2. Validate the XML file

You can use this tool https://www.freeformatter.com/xml-validator-xsd.html to validate your XML file.

1. Copy your XML file in “XML Input” box;
2. Click on "VALIDATE XML" button;
3. “The XML document is valid”, when the green box will appear. A red notification will advise you when some errors are reported;

Note that the XML file must be both **valid** (i.e., compliant to EPOS-DCAT-AP schema) and **consistent**. So, you have to put attention to fill the entities in the right way.



3. Upload the XML file

- Log in GitHub https://github.com/epos-eu/EPOS-DCAT-AP;
- Create a fork on your personal account by the "fork" button;
- Browse into your forked project (at the top of the page you should read youraccount/EPOS-DCAT-AP forked from epos-eu/EPOS-DCAT-AP) and go to the directory https://github.com/epos-eu/EPOS-DCAT-AP/tree/master/examples/WPXX;
- Click on "Upload files" button;
- Drag your file into the graphic dashed square;
- Do the commit, inserting a properly label (es. add xml dcat file);
- Move to the (second) tab ("pull requests") and click on "New pull request" button and then "Create Pull Request". Insert a title for the request and finally confirm by clicking on "Create Pull Request".
- The file will be available as soon as the Administrator will approve the request.