EDI Client

Fabio Pavesi

June 21, 2023

Contents

1 Introduction

EDI is a template-based metadata editor.

2 Templates

Templates define the rules for the standards the metadatum it represents must comply to.

From a standards perspective it specifies all destination XPaths for the elements and items it contains and whether said destinations can be repeated or not.

Templates also specify how the metadata editor will behave, both in terms of validation and of manner elements and items are displayed.

Every template must contain x sections:

- settings
- endpointTypes
- datasources
- group

Templates are written in XML.

An XSD schema is available for validating at this URL A collection of sample templates is available here.

2.1 Settings

2.1.1 userInterfaceLanguage

Labels can be defined in as many languages as required, by using the *xml:lang* attribute.

The userInterfaceLanguage tag defines which *xml:lang* value should be selected for labels and help tooltips.

2.1.2 metadataLanguage

Defines the language to be used when retrieving datasets from datasources.

2.1.3 metadataEndpoint

Defines the endpoint of the EDI Server instance that should be used to convert the metadata into its XML format.

2.1.4 sparqlEndpoint

Defines the default SparQL endpoint.

2.1.5 requiresValidation

Can be set to false (default is true), if you want the metadata to be sent even if they have some errors.

2.1.6 baseDocument

This is, as the name suggests, the base of the XML document to be generated: it is a CDATA and it must include the root element, along with any namespaces that need to be defined.

2.1.7 xsltChain

A chain of XSL Transformations can be specified to further process the XML generated.

> **Warning**
>
> It is the template author's responsibility to make sure they are correct.

2.2 endpointTypes

Contains one or more endpointType tags. Each tag defines the interface to communicate with a SparQL endpoint.

It must have these attributes:

Attribute	Description
xml:id	virtuoso or fuseki
method	GET or POST
query	name of the parameter holding the query

And the child tag *parameters*, whose children are *parameter* tags. Each parameter defines a query-string parameter with name and value to be sent to the endpoint.

Name and value are specified as attributes of the *parameter* tags.

2.3 datasources

Datasources provide valid values for specific *items*.

A collection of datasources: each datasource can be one of *codelist*, *sparql* or *singleton*.

2.3.1 sparql

The most general type of datasource is a SparQL query.

It has two attributes:

Attribute	Description
xml:id	unique id
endpointType	reference to an existing (declared) endpointType

It requires one child tag named *query*, specifying the SparQL query.

Optional child tag *url* allows to override the *sparqlEndpoint* (see 2.1.4) specified in the *settings* tag.
Query can include a *$search_param* token, which, if found, will be given a value based, for example, on user text.

2.3.2 codelist

A codelist is a simplified version of a *sparql* datasource, based on a pre-defined query, accessed via its URI, specified by the child tag *uri*.

Optional child tag *url* allows to override the *sparqlEndpoint* (see 2.1.4) specified in the *settings* tag.

2.3.3 singleton

A singleton is a special stateful *sparql* datasource guaranteed to have only a single instance, so that it can be used to keep some items aligned to some other item whenever the latter changes.

The item triggering said alignment is specified by the attribute *triggerItem*.

Another datasource is always needed, for the singleton to work: the trigger item refers to a sparql or codelist datasource, whereas the dependent items are connected to it via the singleton, which will refresh and select a single row of the singleton dataset, which is linked, in turn, to the uri of the row selected by the trigger item.

Optional child tag *url* allows to override the *sparqlEndpoint* (see 2.1.4) specified in the *settings* tag.

2.4 datasources

Datasources provide valid values for specific *items*.

A collection of datasources: each datasource can be one of *codelist*, *sparql* or *singleton*.

2.4.1 sparql

The most general type of datasource is a SparQL query.

It has two attributes:

Attribute	Description
xml:id	unique id
endpointType	reference to an existing (declared) endpointType

It requires one child tag named *query*, specifying the SparQL query.

Optional child tag *url* allows to override the *sparqlEndpoint* (see 2.1.4) specified in the *settings* tag.
Query can include a *$search_param* token, which, if found, will be given a value based, for example, on user text.

2.4.2 codelist

A codelist is a simplified version of a *sparql* datasource, based on a pre-defined query, accessed via its URI, specified by the child tag *uri*.

Optional child tag *url* allows to override the *sparqlEndpoint* (see 2.1.4) specified in the *settings* tag.

2.4.3 singleton

A singleton is a special stateful *sparql* datasource guaranteed to have only a single instance, so that it can be used to keep some items aligned to some other item whenever the latter changes.

The item triggering said alignment is specified by the attribute *triggerItem*.

Another datasource is always needed, for the singleton to work: the trigger item refers to a sparql or codelist datasource, whereas the dependent items are connected to it via the singleton, which will refresh and select a single row of the singleton dataset, which is linked, in turn, to the uri of the row selected by the trigger item.

Optional child tag *url* allows to override the *sparqlEndpoint* (see 2.1.4) specified in the *settings* tag.

3 Data Types

3.1 Base data types

3.1.1 text

3.1.2 string

Simplest control type: a small rectangle accepting generic text.

Figure 1: Text example

3.1.3 URN

Calculated by the server if **isFixed="true"**. Server will generate a valid and unique URN for you.

3.1.4 URI

Accepts a string and verifies it's an URI.

3.1.5 URL

Accepts a string and verifies it's an URL.

3.1.6 int

Accepts a string and verifies it's an only contains numeric digits.

3.1.7 float

Accepts a string and verifies it's an only contains numeric digits or the decimal separator (i.e. a dot).

With attribute **show="sliderfloat"**
Shows a slider with a minimum and a maximum value and the position generates a value for this control.

```
<item hasDatatype="float" show="sliderfloat" hasIndex="4"
xml:id="slider2" isFixed="false" min="0.0" max="100.00"
step="0.5">
<label xml:lang="en">Slider Float 2</label>
<label xml:lang="it">Slider Float 2</label>
<defaultValue>49</defaultValue>
<hasValue>70</hasValue>
<hasPath>slider</hasPath>
</item>

```

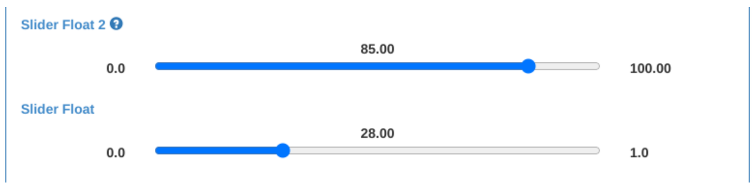


Figure 2: Example of floating-point control with sliderfloat

Shows a slider with a minimum and a maximum value and the position generates a value for this control.

```
<item hasDatatype="float" show="sliderfloat" hasIndex="4" xml:id=
  "slider2" isFixed="false" min="0.0" max="100.00" step="0.5">
<label xml:lang="en">Slider Float 2</label>
<label xml:lang="it">Slider Float 2</label>
<defaultValue>49</defaultValue>
<hasValue>70</hasValue>
<hasPath>slider</hasPath>
</item>
```

3.1.8 real

Same as *float*.

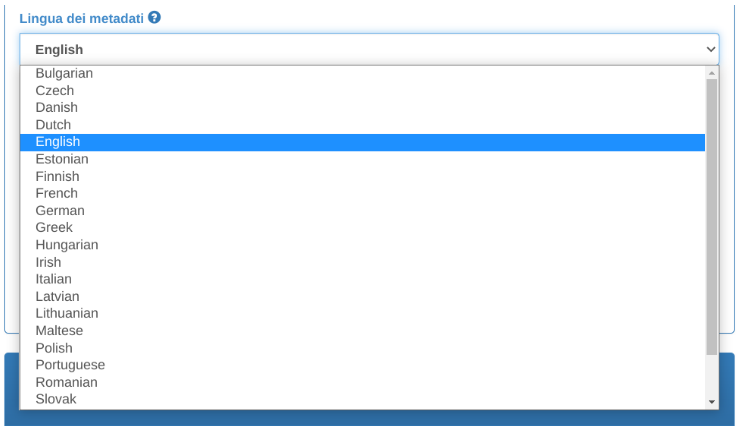
3.1.9 double

Same as *float*.

3.1.10 codelist

```
<item hasIndex="1" xml:id="ling_md_1" isLanguageNeutral="true"
  isFixed="false" hasDatatype="codelist" datasource="languages"
  show="combobox">
<hasPath>/gmd:MD_Metadata/gmd:language/gmd:LanguageCode</hasPath>
</item>
```

Figure 3: Codelist example with *show="combobox"*



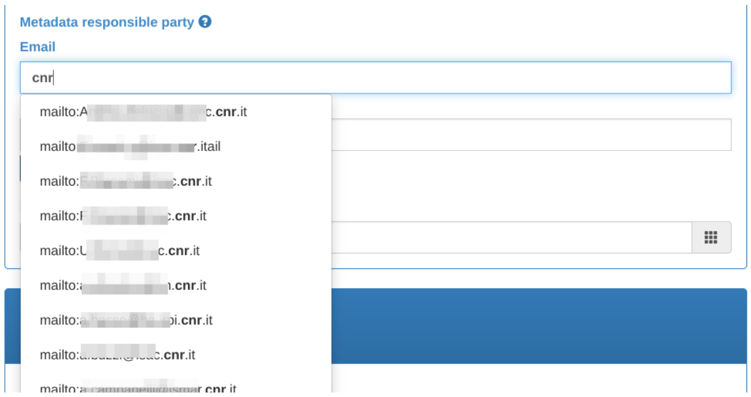
3.1.11 autoCompletion

Similar to a *codelist*, but preferrable with datasources containing many rows.

A textbox querying the datasource associated to the control for matching values. Starts querying when at least 3 characters are entered.

```xml
<item hasIndex="1" xml:id="md_resp_1" outIndex="2" isFixed="false" hasDatatype="autoCompletion" datasource="person">
<label xml:lang="en">Email</label>
<label xml:lang="it">Email</label>
<hasPath>/gmd:MD_Metadata/gmd:contact/gmd:CI_ResponsibleParty/gmd:contactInfo/gmd:CI_Contact/gmd:address/gmd:CI_Address/gmd:electronicMailAddress/gco:CharacterString</hasPath>
</item>
```

Figure 4: AutoCompletion example


Metadata responsible party

Email

cnr

mailto:A[redacted]c.cnr.it

mailto[redacted].itail

mailto:[redacted].cnr.it

mailto:I[redacted].cnr.it

mailto:U[redacted]c.cnr.it

mailto:[redacted].cnr.it

mailto:[redacted]i.cnr.it

mailto:[redacted]c.cnr.it

mailto:a[redacted]r.cnr.it

3.1.12 boolean

Shows a check-box, thus allowing only values **true** or **false**.

3.2 Special case data types

3.2.1 label

Shows read-only text.

3.2.2 image

Given an URL pointing to a valid image in the value, it shows the image.

3.2.3 qrcode

Shows whatever the value is as a QR code.

3.2.4 select

The select datatype signifies that the item's value is based on some selection occurring in another item called a *trigger item*.

It must be based on a data source of type **singleton** (see 2.4.3).

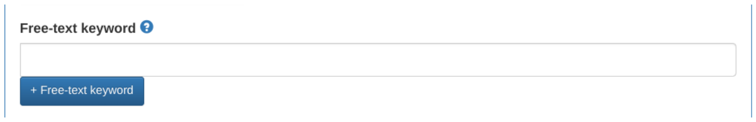
Sometimes the trigger item can be based on the same data source as its connected *select* items, but it can be based on its own data source.

Each *select* item must declare the field it represents in the datasouce.

```
<item hasIndex="2" xml:id="resp_2" outIndex="1" field="inst"
  isFixed="false" hasDatatype="select" datasource="personS_2">
<label xml:lang="en">Institute</label>
<label xml:lang="it">Ente</label>
<hasPath>/gmd:MD_Metadata/gmd:identificationInfo/
  gmd:MD_DataIdentification/gmd:citation/gmd:CI_Citation/
  gmd:citedResponsibleParty/gmd:CI_ResponsibleParty/
  gmd:organisationName/gco:CharacterString</hasPath>
</item>
```

3.2.5 copy

Figure 5: String example, in this case it is part of a *isMultiple="true"* element, as you can tell from the "+" button underneath it



3.2.6 function

Special data type. Its value is calculated by the server by using its template *hasValue* as an XPath run **against the parts of document that have already been generated**.

3.2.7 ref

Special data type. In the generated metadata document, it copies the Xpath specified in the *hasValue* attribute to the Xpath specified by the *hasPath* attribute.

```
<item hasDatatype="ref" hasIndex="7" xml:id="title_7" isFixed="
  true">
<hasPath>dct:description/@xml:lang</hasPath>
<hasValue>/rdf:RDF/dcatapit:Dataset/dct:title/@xml:lang</hasValue
  >
</item>
```

In the example above, once the XML document is fully written, the **xml:lang** attribute of **dct:description** is set to equal the same attribute in **/rdf:RDF/dcatapit:Dataset/dct:title**.

3.2.8 autonumber

Represents a value that's incremented every time it is encountered within the containing element.

Value is assigned by EDI Server.

3.2.9 hidden

Hidden item: it is *fixed* by default (i.e. read-only).

3.2.10 date

Requests a date from the user, via a small calendar.

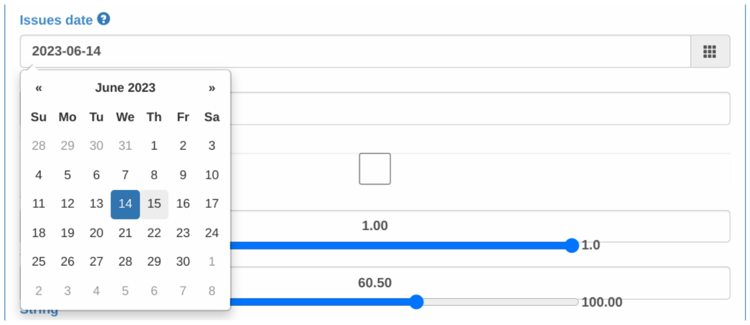
Requires a *defaultValue*, which can be the macro *$TODAY$*.

```

<item hasDatatype="date" hasIndex="7" xml:id="test_7" isFixed="false">
<label xml:lang="en">Issues date</label>
<label xml:lang="it">Data di rilascio</label>
<help xml:lang="en">Help</help>
<help xml:lang="it">Help</help>
<hasValue>dct:issued</hasValue>
<defaultValue>$TODAY$</defaultValue>
</item>

```

Figure 6: Date example



3.2.11 dateRange

Same as *date*, except that it requests a start and an end date.

```

<item hasIndex="8" xml:id="est_temp_8" isFixed="false" hasDatatype="dateRange">
<label xml:lang="en">Start date</label>
<label xml:lang="it">Data inizio</label>
<start>
<label xml:lang="en">Start date</label>
```

```
<label xml:lang="it">Data inizio</label>
<hasPath>/gmd:MD_Metadata/gmd:identificationInfo/
  gmd:MD_DataIdentification/gmd:extent/gmd:EX_Extent/
  gmd:temporalElement/gmd:EX_TemporalExtent/gmd:extent/
  gml:TimePeriod/gml:beginPosition</hasPath>
</start>
<end>
<label xml:lang="en">End date</label>
<label xml:lang="it">Data fine</label>
<hasPath>/gmd:MD_Metadata/gmd:identificationInfo/
  gmd:MD_DataIdentification/gmd:extent/gmd:EX_Extent/
  gmd:temporalElement/gmd:EX_TemporalExtent/gmd:extent/
  gml:TimePeriod/gml:endPosition</hasPath>
</end>
</item>

```

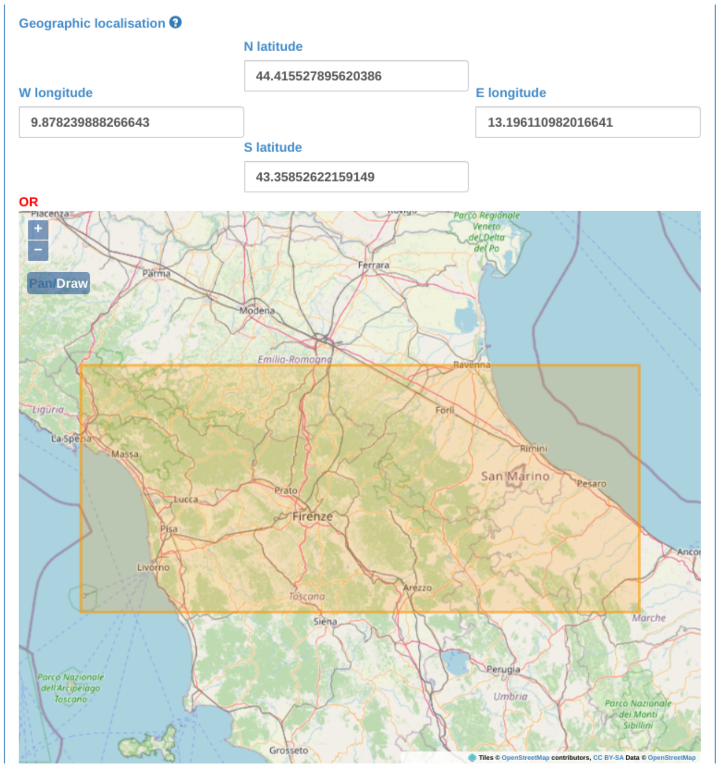
3.2.12 boundingBox

Requests a geographic bounding box from the user. It can be specified either by inputting the coordinates in 4 text boxes, or by drawing a rectangle on a map.

```

<item hasIndex="1" xml:id="loc_geo_1" isFixed="false" hasDatatype
  ="boundingBox">
<westLongitude outIndex="1" queryStringParameter="westlon">
<label xml:lang="en">W longitude</label>
<label xml:lang="it">Longitudine O</label>
<hasPath>/gmd:MD_Metadata/gmd:identificationInfo/
  gmd:MD_DataIdentification/gmd:extent/gmd:EX_Extent/
  gmd:geographicElement/gmd:EX_GeographicBoundingBox/
  gmd:westBoundLongitude/gco:Decimal</hasPath>
</westLongitude>
<eastLongitude outIndex="2" queryStringParameter="eastlon">
<label xml:lang="en">E longitude</label>
<label xml:lang="it">Longitudine E</label>
<hasPath>/gmd:MD_Metadata/gmd:identificationInfo/
  gmd:MD_DataIdentification/gmd:extent/gmd:EX_Extent/
  gmd:geographicElement/gmd:EX_GeographicBoundingBox/
  gmd:eastBoundLongitude/gco:Decimal</hasPath>
</eastLongitude>
<northLatitude outIndex="4" queryStringParameter="northlat">
<label xml:lang="en">N latitude</label>
<label xml:lang="it">Latitudine N</label>
<hasPath>/gmd:MD_Metadata/gmd:identificationInfo/
  gmd:MD_DataIdentification/gmd:extent/gmd:EX_Extent/
  gmd:geographicElement/gmd:EX_GeographicBoundingBox/
  gmd:northBoundLatitude/gco:Decimal</hasPath>
</northLatitude>
<southLatitude outIndex="3" queryStringParameter="southlat">
<label xml:lang="en">S latitude</label>
<label xml:lang="it">Latitudine S</label>
<hasPath>/gmd:MD_Metadata/gmd:identificationInfo/
  gmd:MD_DataIdentification/gmd:extent/gmd:EX_Extent/
  gmd:geographicElement/gmd:EX_GeographicBoundingBox/
  gmd:southBoundLatitude/gco:Decimal</hasPath>
</southLatitude>
</item>
```

Figure 7: Bounding box example



Geographic localisation

N latitude
44.415527895620386

W longitude
9.878239888266643

E longitude
13.196110982016641

S latitude
43.35852622159149

OR

Pan/Draw
Piacenza
Parma
Modena
Ferrara
Parco Regionale Veneto del Delta del Po
Emilia-Romagna
Ravenna
Liguria
La Spezia
Massa
Forlì
Rimini
San Marino
Pesaro
Prato
Lucca
Firenze
Pisa
Livorno
Arezzo
Toscana
Siena
Marche
Perugia
Umbria
Parco Nazionale dell'Arcipelago Toscano
Parco Nazionale dei Monti Sibillini
Grosseto
Tiles © OpenStreetMap contributors, CC BY-SA Data © OpenStreetMap