Europeana Inside

Manual Mapping and Transformation Service – release 1

W. de Croylaan 54 bus 5592 ■ BE-3001 Heverlee ■ +32 16 32 22 66 ■ into-info@libis.be ■ www.libis.be

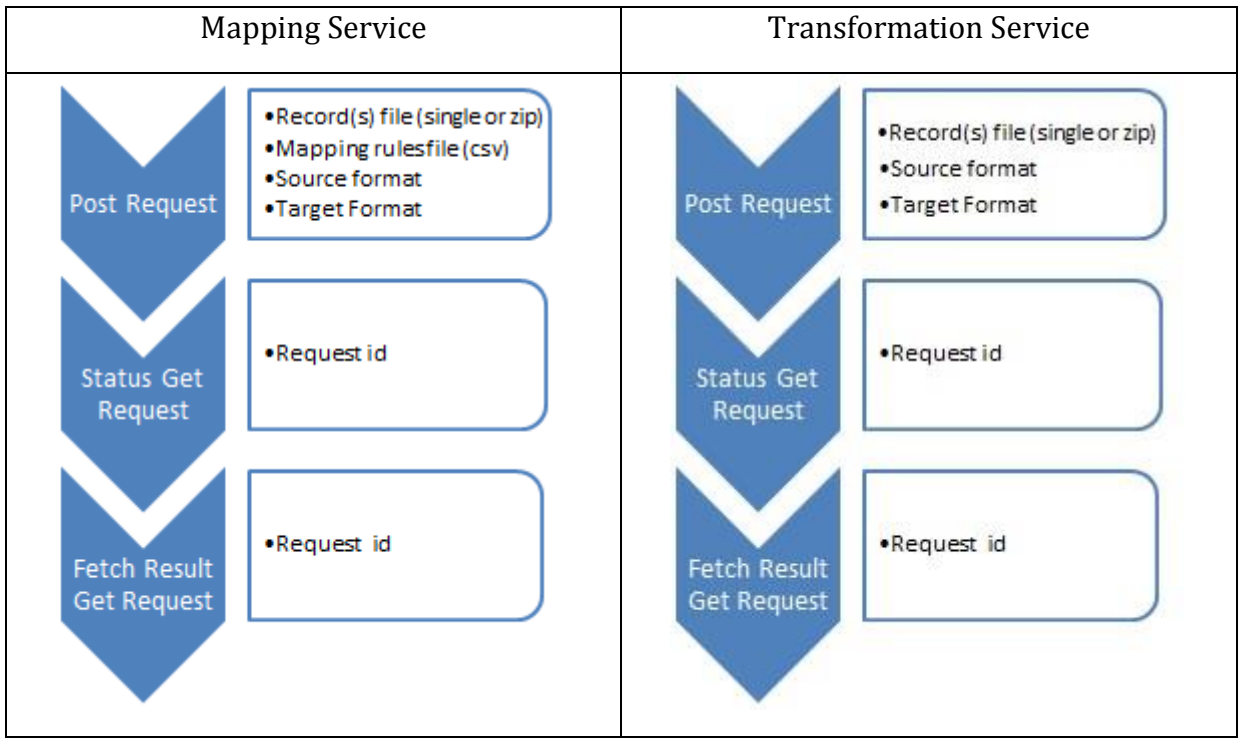
Introduction

This document describe the use of REST api developed to map and transform records from one format to another. The services of this api can be divide into two categories: Mapping Service and Transformation Service.

The mapping service can be used to convert LIDO to EDM and MARC to EDM. The mapping is defined in a text file with the specified mapping rules. The mapping rules consists of an action, input and output. Sometimes there is also a fixed value. The rules will be applied during the transformation.

The mapping service is only necessary if you want to make your own mapping. If you use the mapping rules specified by Linked Heritage project, then you don't need this mapping service. In this case you need to use the transformation service instead of the mapping service.

Following diagram shows the workflow of both mapping and transformation service. It can be noticed that no mapping rules file is needed for transformation service.



1. Mapping Service

Maps records from one format to another (currently from LIDO to EDM) based on the provided mapping rules.

1.1 Use of the service

You can send POST and GET requests to the server. The POST request is for sending the data, the GET request is for getting information about the data and fetching the data.

1.1.1 POST request

The POST request can be used to transform a record or records with the defined mapping rules.

URL

Following url along with the parameters in the body can be used to access this service:

http://www.heron-net.be/einside_test/dmt.php/DataMapping/<provider>/<batch>/Transform

- DataMapping: is the name of the module therefore it should remain the same.
- Provider: This is the name of the organization making this request.
- Batch: This is the name of the batch submitted for transformation.
- Parameters: This service accepts following three parameters:
    - records or record:

      A ZIP file with XML records in them (records parameter) or an XML file with one or more records in the XML file (record parameter). You also have to provide the correct Content-type: ZIP: application/zip. XML: application/xml or text/xml (the XML content-type is not yet validated) You have to choose record or records. Supplying both results in an error.

    - mappingRulesFile:

      A csv file (Content-type:text/csv) with mapping rules (See section 1.1.3 for an explanation about this file).

    - sourceFormat:

      This is the source metadata format. At the moment only LIDO is supported.

    - targetFormat:

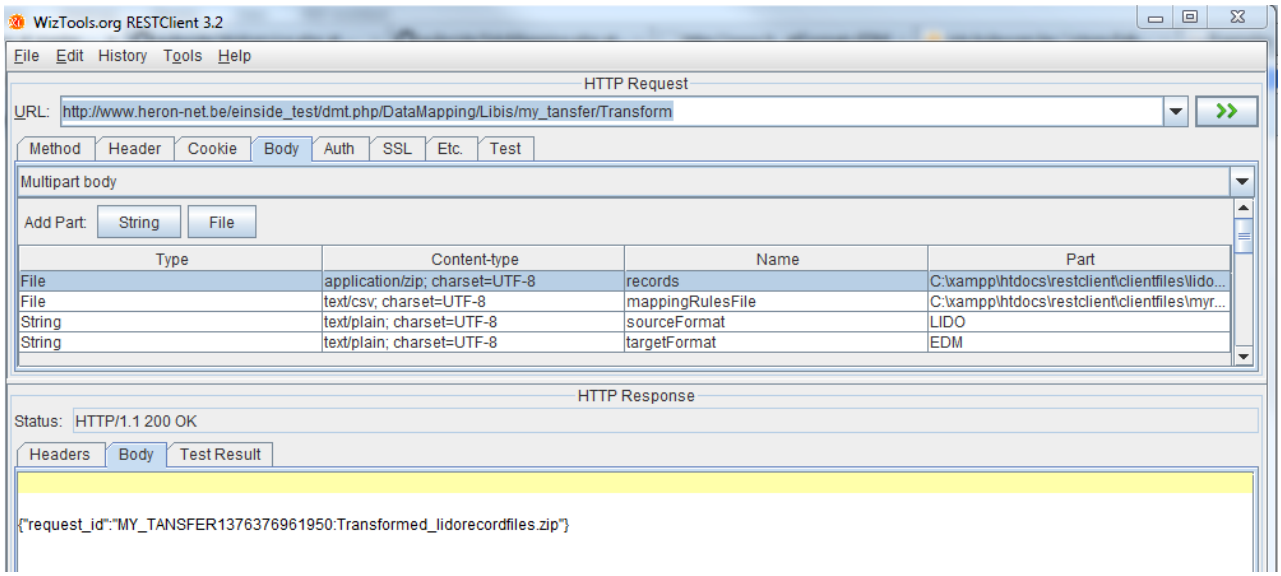
      This is the target metadata format. At the moment only EDM is supported.

An example url is:

http://www.heron-net.be/einside_test/dmt.php/DataMapping/Libis/my_tansfer/Transform

Where 'Libis' is the name of the provider of the record(s) and 'my_tansfer' is the name of the batch.

The example url with required parameters is also shown in the following diagram.



POST response

The response of the service is a request_id. E.g.:

{"request_id":"MY_TANSFER1376376799092:Transformed_lidorecordfiles.zip"}.

With this request_id you can get the status and download the Transformed files. (See further in the Get request). The MY_TRANSFER in the response comes from the my_transfer in the request url.

1.1.2 GET-request

You can use the GET request to get information from the service:

1.1.2.1 Status

Status gives back a status code about the processing of the records:

- 0: no request exist with this id
- 1: not yet mapped/transformed
- 2: ready to be fetched

URL

Example url: http://www.heron-net.be/einside_test/dmt.php/DataMapping/Libis/my_transfer/status?request_id=MY_TANSFER1376036187653:Transformed_lidorecordfiles.zip

The url is the same as the POST request, only the transform will become status and the required parameter needs to be added. The Url also needs to consist of Datamapping, organisation, batch name and action. Currently only the 'action' value is processed but other values are also needed in the url to make it a valid url.

Parameters:

- Request_id: is the request_id from the POST-response

1.1.2.2 Fetch

Fetch can be used to download the mapped/transformed files.

URL

Example url: http://www.heron-net.be/einside_test/dmt.php/DataMapping/Libis/my_transfer/Fetch?request_id=MY_TANSFER1376036187653:Transformed_lidorecordfiles.zip

Url for fetch request is the same as the status request only status becomes fetch. The parameter is also request_id.

1.1.2.3 List

This returns a list of supported metadata formats (currently only EDM and LIDO)

URL

Example url: http://www.heron-net.be/einside_test/dmt.php/DataMapping/Libis/my_transfer/List

The url is the same as the other GET requests, the action is List. There are no parameters.

1.1.3 Mapping Rules

The mapping rules are used to define which source field goes to which target field. You can also define an action to be executed on the source field before it is added to the ~~source~~ target field.

The different components are separated by a comma, but the file is not a valid csv file as some actions need to have new lines in them. Commas are allowed in the values. The

rules are processed from top to bottom. The supported mapping rules with appropriate examples are given at the end of this document in annex 1.

Source field

The source field (see table below) contains the field from the source metadata format (sourceFormat parameter in the POST request). The mapping service supports only LIDO as source format. MARC support will be added. For the LIDO format we use a XPath-like format. We didn't find a more condense way to select the correct source field in LIDO. You can also select an attribute with an @. In MARC we will have a much shorter syntax.

Target field

The target field (see table below) contains the field from the target metadata format (targetFormat parameter in the POST request). The mapping service supports only EDM as target format.

There are some limitations in the current EDM support. The edm:isShownBy is not supported. You have to use edm:object instead. IsShownBy will be supported in some weeks. Also not yet supported are edm:Place, edm:Agent, edm:TimeSpan, and skos:Concept. This means if you supply a value that are places, persons, organisations or concepts they will be added to the dc or edm elements in for example ProvidedCHO, but we won't create an extra edm:Place, edm:Agent or skos:Concept element yet. This won't be implemented in the first release, because it is not required and we will need to think about how we can implement this in a user-friendly way without making the target field too complex.

Edm:rights and dc:rights are not processed correctly at the moment. This means that we create the elements, but it is possible they are not in the right element. Some values are still empty. The reason is this needs some further testing, because it depends on the context if the elements need to go to providedCHO, aggregation or webResource element.

2. Transformation Service

Transforms records from one format to another (currently from LIDO to EDM) based on a style sheet provided by Linked Heritage.

2.1 Use of the service

Like the mapping service you can send POST and GET requests for transformation service. The POST request is for sending the data (without mapping rules file), the GET requests are for getting information about the data and fetching the transformed data.

2.1.1 POST request

The POST request can be used to submit record(s) for transformation (without mapping rules).

URL

Following url along with the parameters in the request body can be used to access this service:

http://services.libis.be/euInside/dmt.php/DataMapping/<provider>/<batch>/Transform

- DataMapping: is the name of the module therefore it should remain the same.
- Provider: This is the name of the organization making this request.
- Batch: This is the name of the batch submitted for transformation.
- Parameters: This service accepts following three parameters:
    - Records or record:

      A ZIP file with XML records in them (records parameter) or an XML file with one or more records in the XML file (record parameter). You also have to provide the correct Content-type: ZIP: application/zip. XML: application/xml or text/xml (the XML content-type is not yet validated) You have to choose record or records. Supplying both results in an error.
    - sourceFormat:

      This is the source metadata format. At the moment only LIDO is

      supported.
    - targetFormat:

      This is the target metadata format. At the moment only EDM is supported.

Following is an example url:

http://services.libis.be/euInside/dmt.php/DataMapping/Libis/LeuvenLibData/Transform

Where 'Libis' is the name of the provider of the record(s) and 'LeuvenLibData' is the name of the batch.

Note that no mapping file (mappingRulesFile) is need for this service, a style sheet provide by Linked Heritage will be used to transform the record(s).

POST response

The response of the transformation service is similar to the response of the mapping service, that is a request_id in json format. An example response is:

{"request_id":"MY_TANSFER1378883070287:Transformed_lido1.xml"}

Subsequently, this request_id can be used to check the status of the submitted records and to fetch the transformed records. For this purpose following given GET requests can be used.

2.1.2 GET-request

You can use the GET request to get information about your submitted record(s):

2.1.2.1 Status

Status gives back a status code about the processing of the records. The status GET request is the same as described in section 1.1.2.1 except the base url which points to a different server.

URL

An example url is:

http://services.libis.be/euInside/dmt.php/DataMapping/Libis/LeuvenLibData/status?request_id=MY_TANSFER1378883070287:Transformed_lido1.xml

- Status: is the action which returns the status of the submitted records.
- request_id: is id of the transformation request.

2.1.2.2 Fetch

Fetch can be used to download the transformed records.

URL

An example url is:

http://services.libis.be/euInside/dmt.php/DataMapping/Libis/LeuvenLibData/fetch?request_id=MY_TANSFER1378883070287:Transformed_lido1.xml

The url is similar to the status request url except that action status changes to fetch.

Annex 1-Supported mapping rules

Action	Definition	Input	Output	Example
COPY	Copies the value from a source field to a target field. If different source fields are defined to be copied to the same target field a new occurrence of the target field will be created	Source metadata field	Target field	COPY,/lidoRecID,dc:identifier
APPEND	Appends text to the value of a source field and assigns it to a target field. The specified text is appended at the end of the value. In case no extra information is to append an empty space can be left.	Source metadata field	Target field	APPEND,/category/term, some text ,dc:title or APPEND,/category/term, ,dc:title
PREPEND	Appends text to the value of a source field and assigns it to a target field. The specified text is appended at the beginning of the value. In case no extra information is to prepend an empty space can be left.	Source metadata field, Value to prepend	Target field	PREPEND,marc300b,Red,dcterms:extent3 Or PREPEND,marc300b,,dcterms:extent3

Action	Definition	Input	Output	Example
LIMIT	Limit the number of characters in a source field	Source field and number of characters	Target field	LIMIT,/descriptiveMetadata/objectIdentificationWrap/objectMeasurementsWrap/objectMeasurementsSet/displayObjectMeasurements,9,edm:description
COMBINE	Multiple source fields can be combined in one target field. The fields are combined with a space. Additionally, value of the last item can be surrounded by brackets.	Source fields, separate by ;	Target field	LIDO Example COMBINE, /category/term;/lidoRecID, dc:title MARC Examples COMBINE,(,marc245a;marc245b;marc001;marc340a,), dc:title or COMBINE,,marc245a;marc245b;marc001;marc340a,,dc:title
SPLIT	The source field will be split on the defined character and put in the target field. (In the example it will be split on space)	Source field and character to split on	Target field	SPLIT,/descriptiveMetadata/eventWrap/eventSet/event/eventMaterialsTech/displayMaterialsTech,,dcterms:medium
PUT	Add a value to a target field. The value can contain commas, but no 2 pipes (||), because this is used as replacement character	Value	Target field	PUT,"Koninklijke Musea voor Kunst en Geschiedenis, Brussel",edm:dataProvider

Action	Definition	Input	Output	Example
REPLACE	Replaces a value in the source field with a replace string in the target Field. In the example d'Histoire will be replaced by History in the edm:provider element	Source field, value to be replace and replace value	Target field	REPLACE,/administrativeMetadata/rightsWorkWrap/rightsWorkSet/rightsHolder/legalBodyName/appellationValue,d'Histoire,History,edm:provider

CONDITION	With condition you can combine different actions and use a conditional flow.
	Nested conditions are not yet implemented. IF can be used to define conditional action. With nested conditions also ELSE IF can be used.
	The CONDITION actions starts and ends with curly brackets '{ '. A new line is also mandatory.
	See the Condition action table for a breakdown of the condition syntax
				CONDITION,{
				
				IF[/descriptiveMetadata/eventWrap/eventSet/event/eventType/term,EQUALS,Production,DO(COMBINE, /category/term;/lidoRecID, dc:title)]
				
				}
				
				CONDITION,{
				
				IF[marc340a,NOT EQUAL,"Poster",DO(COPY,marc001,dc:identifier)]
				
				ELSE[APPEND,marc245a,nm,dc:title]
				
				}

Condition action

Part	Syntax	Definition	Example
IF and source field	IF[source field, comparator, value to compare, action]	The IF has the condition between squared brackets '[ ]'	IF[/descriptiveMetadata/eventWrap/eventSet/event/eventType/term,…]
Comparator	EQUALS	Available for LIDO and MARC to EDM	IF[/descriptiveMetadata/eventWrap/eventSet/event/eventType/term,EQUALS,…]
	NOTEEQUAL	Available for MARC to EDM	
Value to compare		The value is a string. Like all values this part can contain commas	IF[/descriptiveMetadata/eventWrap/eventSet/event/eventType/term,EQUALS,Production,…]
Action	DO(action)	The action starts with DO. Between the round brackets an action from the previous table is defined (except of course condition)	IF[/descriptiveMetadata/eventWrap/eventSet/event/eventType/term,EQUALS,Production,DO(COMBINE, /category/term;/lidoRecID, dc:title)]

ELSE	ELSE [A mapping rule.]	In case of condition not fulfilled in IF part, mapping rule given in ELSE part will be executed.
		
		Currently, this is only available for MARC to EDM conversion.
			CONDITION,{
			
			IF[marc340a,NOT EQUAL,"Poster",DO(COPY,marc001,dc:identifier)]
			
			ELSE[APPEND,marc245a,Belgium,dc:title]
			
			}

Example files:

XML Document

Lido file:

Microsoft Excel Macro-Enabled Works

Mapping rules lido:

marcmappingrules.csv

Mapping rules marc: