Fusepool P3 Cookbook

A User's Guide to the Fusepool P3 Platform

Oskari Romanoff

Bern University of Applied Sciences
E-Government Institute

Table of Contents

1 What is the Fusepool P3 Platform?

Fusepool P3 project provides tools to work with Linked Open Data (LOD). It was created to make the publishing and processing of Linked data easy. A brief explanation about what Fusepool P3 platform is about, what it is used for and for whom it is intended can be found from their factsheet:

http://p3.fusepool.eu/fp3-factsheet.pdf?attredirects=0

For more details, it is recommended to get familiar with their website:

http://p3.fusepool.eu/ - p3

2 How Does the Fusepool P3 Platform Work?

The Fusepool P3 platform is used for storing and retrieving RDF and non-RDF data. You can transform unstructured data to RDF files and extract them. Below you can see a figure which represents the main structure of the platform.
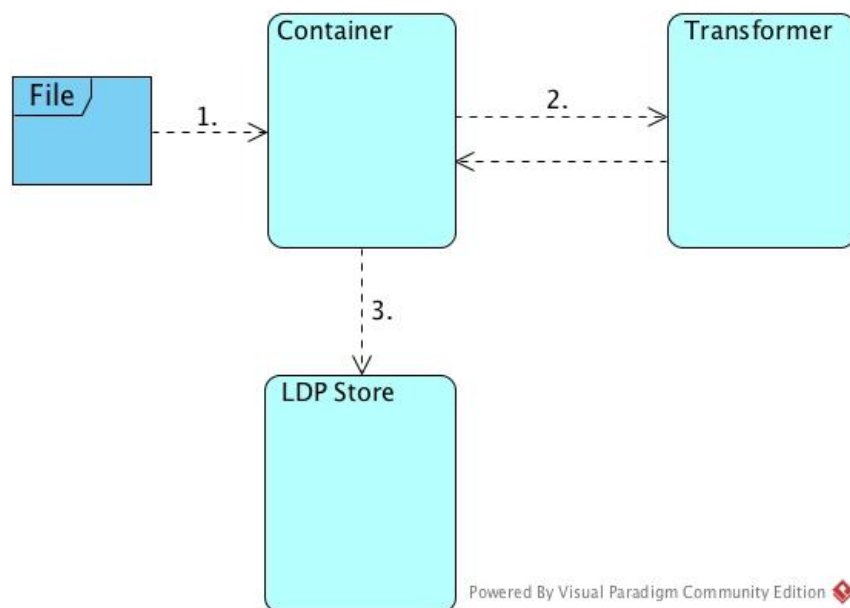
Figure 1 The main structure of the Platform

Here is a brief description how the platform actually works.

1. First user uploads a file to container as URL or locally.
2. Container sends the file to transformer and transformer sends a ready RDF file back to container.
3. The RDF file is stored to the LDP store which can be queried as SPARQL

The container is used to send the file to the transformer. So the container is used only for receiving and sending the data for other components. On the platform the container can be seen as a widget which user generates and links to use the desired transformer.
Transformer receives the file, transforms it into RDF and sends the transformed file back to the

container. Fusepool P3 provides variety of different transformers which user can generate as he/she needs.

Finally the transformed file is stored in the LDP store of the platform and the data can be queried as SPARQL from their own SPARQL UI.

3 Using the Fusepool P3 platform

3.1 Data publication process

Next we are going through the data publication process. Here we describe briefly what is needed to make a ready data set, which you can work with on Fusepool P3 Platform.

Ready data file (CSV)

In order to transform data files, you naturally need a ready data set with some data. You can create your own data or use an existing file.

Database design for a data file (Class diagram)

The database design helps you to perceive the structure of your data. It is good to create for example, a class diagram to represent the structure of your data. This helps on the later phase when mapping is created with OpenRefine.

Ontology/Vocabulary

Once the dataset and the database design with classes and properties are ready, you need vocabularies for it. Vocabularies are used for example, to display knowledge about something or they can be used for organizing knowledge. On our tutorial below, we are using vocabularies from business and relation fields. There exist already many vocabularies from different fields but sometimes you need to create your own vocabulary so you can be able to determine every item on your data set. For searching vocabularies, it is a good way to list all the classes with properties and then search for vocabularies for each individual item.

- Vocabulary is used for mapping
- If there are no existing vocabularies for your data, you have to create your own.
- Search from internet for existing vocabularies

3.2 How to Use the Fusepool P3 Platform?

For transformation you need two files:
1. A CSV file you want to transform
2. Mapping file as JSON

Before you continue, download a CSV example file. Example file is about persons with high income in UK.

File: https://github.com/OskariRomanoff/Tutorial-Fp3

Click clone or download and download a ZIP file.

Next let´s go to the Fusepool P3 platform
http://sandbox.fusepool.info:8200/?platformURI=http://sandbox.fusepool.info/

When you open the website it brings you to the Fusepool P3 platform. On the left you can see publishing, transformers and configuration pages. We go through them briefly.

Publishing: From publishing page you can find the containers which are called "Widgets". This is the page where you can manage your files that you want to transform and also manage already transformed files. The file is "sent" to the transformer through the container and the transformed file appears on the widget. You can also access the datasets of the transformed files which are called triples.

Transformers: On transformer page are the transformers. From here you can create your own transformers or use existing ones. In the middle you can see a list of existing ones. On the right side section "Factories" you can find the transformer factories provided by the Fusepool P3. When you create a transformer it will appear in the middle on the transformer list. It can be renamed, tested etc.

Configurations: From configurations page you can create your own configurations for your widgets. Configuration is like your own "workspace" for publishing page. That means that you can only see your own widgets on the publishing page so it is easier to work if there are many widgets.

The pages relevant for this work are publishing and transformer pages.

3.3 Using the OpenRefine to generate a BatchRefine transformer

We are working with Fusepool´s p3-batchrefine distribution for OpenRefine. Distribution adds batch processing capabilities to OpenRefine supporting multiple back ends. OpenRefine (before GoogleRefine) is an open source application for data clean up and transformation to other forms.
On this section you will learn how to create a BatchRefine transformer by using OpenRefine.

The process includes these phases:

Generating a transformer:

→ Creating a project
→ Creating a mapping file by using OpenRefine
→ Creating a transformer by using OpenRefine with Fusepool P3 distribution

Creating project	Mapping	Creating transformer	Testing transformer

Figure 2 Process of generating a transformer

Using the transformer:

→ Creating a container for the data
→ Uploading the data to the container
→ Transforming the data

Creating container	Uploading data	Transforming data

Figure 3 Process of using the transformer

3.3.1 Generating the transformer

3.3.1.1 Creating the project
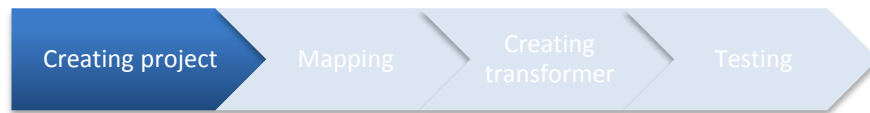
Figure 4 Creating project

Go to Fusepool P3 platform page and select "Transformers" page from the left. On the right side you can see a list of Factories. Choose OpenRefine UI and click "Open factory GUI".

Now you get to the OpenRefine´s page. If you don't have existing projects, you can use the CSV file we provided and create a new one.

→ Choose your project from the "Open project" tab.
→ Now upload the high-earners-pay csv file and click "Next"
→ There are two things to care about before the creating project:

Make sure that columns are separated by commas

Figure 5 Column separation

Above that is Character encoding. Click the text field and choose UTF-8
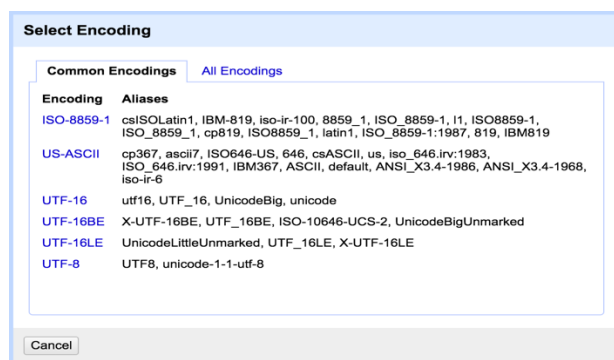
Figure 6 Encoding

The page should look like this:

	Name	Job Title	Grade	Organisation	Annual pay rate - including taxable benefits and allowances	Notes
1.	Stephan Wilcke	Chief Executive Officer		Asset Protection Agency	£150,000 - £154,999	
2.	Jens Bech	Chief Risk Officer		Asset Protection Agency	£165,000 - £169,999	No pension
3.	Ion Dagtoglou	Chief Invesment Officer		Asset Protection Agency	£165,000 - £169,999	No pension
4.	Brian Scammell	Chief Credit Officer		Asset Protection Agency	£130,000 - £134,999	4 days per week
5.	Hamish Buckland	Special Advisor		Asset Protection Agency	£65,000 - £69,999	2 days per week
6.	Stephen Lovegrove	Chief Executive, Shareholder Executive	Director General	BIS	£185,000 - £189,999	
7.	Philip Rutnam	Director General, Business Group	Director General	BIS	£180,000 - £184,999	
8.	Vicky Pryce	Director General, Economics and Joint Head, UK Government Economic Service	Director General	BIS	£175,000 - £179,999	
9.	Prof. John	Government Chief Scientific Adviser	Permanent	BIS	£165,000 - £169,999	

Figure 7 Overview of the upcoming project

Now click Create Project from upper right corner.

3.3.1.2 Creating the mapping file

Figure 8 Mapping

Now we start editing the RDF skeleton to create the mapping file. This is only an example to show how to edit the skeleton. You can display data as you want.

On the upper right corner, you can see the Extensions.

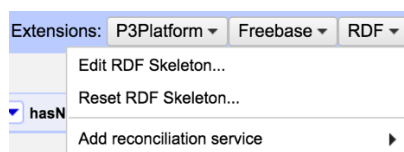→ Click the RDF dropdown menu and choose "Edit RDF Skeleton"

Figure 9 Edit rdf skeleton
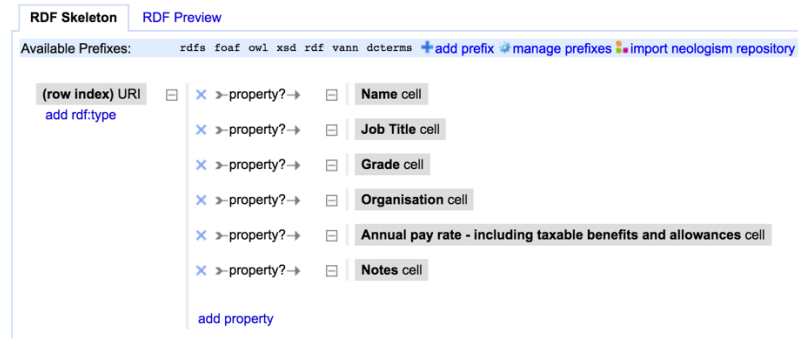
It will open a window like this:

Figure 10 Unedited rdf skeleton

On the picture above you can see the Available Prefixes section. Prefixes are vocabularies which include already defined terms from some field e.g. Business or Geography. Vocabularies are used for example, to display knowledge about something or they can be used for organizing knowledge.

For more information about vocabularies:
https://www.w3.org/standards/semanticweb/ontology

There are already some prefixes but we will add few to have more vocabularies.

→ Click "add prefix" and on the prefix field type: foaf.

Figure 11 Adding prefix

Next add these prefixes: dcat, dct, myns, skos

NOTE: for myns you need to add URI by yourself. It is:
http://myhost.com/myns/

Once you have added prefixes, we start to edit the RDF skeleton from left to right. On this example we want to represent person´s name, title, annual pay rate and the organization where he/she is working.

→ click "(row index URI) and it opens a window. Choose content "Organization" and how the contact is used "as a URI". Then click Ok.
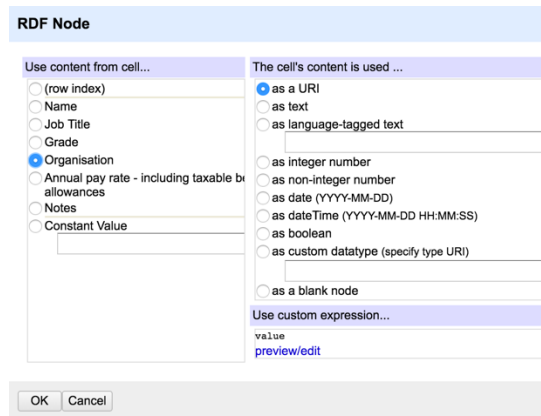
Figure 12 Defining the root node

→ Now below "Organization", you can see "add rdf type". Click it and you can search for a class. First type the prefix name "foaf:" Then it will start provide suggestions. Start to type Organization and choose it from the list.
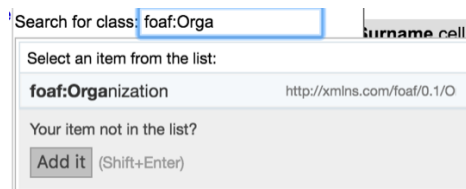

Figure 13 Defining class

→ Next we move to the second row. You can see that there is a long list of properties. Now delete all by clicking the cross next to them, except the one which is linked to Organization cell. Next we edit and add properties.

> → Click the property and type foaf:name and click it. Now the property "name" is linked to organization.
> → Then click add property from the same row. A new property will appear. Click it and type foaf:member

Figure 14 Defining properties

Now the field should look like this

→ Then we edit the third row. Now click the "Configure?" and add content from the cell as (row index) and the content is used "as a URI". Click OK and below add the rdf type "foaf:person". Next to the (row index) is a plus sign, click it to add fourth row.

→ Move to the fourth row and add three properties same way as before.

Properties: foaf:name, foaf:title and :annualPayRate

Note: The vocabularies don't include the annualPayRate so just type it and add it and click apply.

→ Then the fifth and the last row. Modify it like the list below:

foaf:name → Name as text
foaf:title → Job title as text
:annualPayRate → Annual pay rate - including taxable benefits and allowances as text

The skeleton should look like this:

Figure 15 Ready RDF skeleton

We have defined organization as a root node. What we have done here, is that we have defined two properties for organization, its name and members. They both have their own value "Organization cell" and "(row index URI)" which uses "foaf:Person" as a class. It is defined as URI so we can use properties which are defined for Person class. Person class has properties name, title and annual pay rate.

Now we have ready RDF skeleton and it can be saved. Click "Save" from lower right corner

3.3.1.3 Creating the transformer

Figure 16 Creating transformer

Now we are ready to create the transformer. We are using Fusepool P3 platform distribution which includes feature to create the BatchRefine transformer.

→Go to the part "Extensions". Extensions bar should look like this:
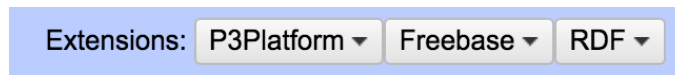
Figure 17 Extension bar

→Create the transformer.
→ Click the drop down menu "P3Platform"
→ Choose "Create transformer"

A window like this should open. On the left you can see the versions of your work. Every time you click save while editing the RDF skeleton, OpenRefine stores the changes and you can choose the version you want. We are going to choose the latest one.
On the right side you can see the JSON file of your work.

Figure 18 Creating the transformer

→ You don't have to care about the first section "Post transformation rules"
→ On the "Register Transformer" section. Change "Transformer name to more individual so you can recognize it
→ Click "Register transformer

Now if everything went right, you should be able to see the transformer you created at the Fusepool platform (where we started creating the transformer). Refresh the page and it should appear on the transformers list.

3.3.1.4 Testing the transformer

Figure 19 Testing the transformer

To make sure your transformer works, it is good to test it before starting to transform files.

Now you can see your transformer on the transformer list. It is usually the last one.

You can test that it is working by clicking "Test"

→ Click "Browse", choose your CSV file
→ Once you have the file uploaded, select input format as "text/csv" and preferred output format as "text/turtle".
→ Click test transformer and you can see the transformation as a result.

After these steps it should look like this:

Figure 20 Creating a test

NOTE: This is just a test that transformer works. Not the transformed file, for that you need to create a widget which is explained on the next phase.

Now once we are sure the transformer works. We can start transforming files.
You can close this window and continue to next section.
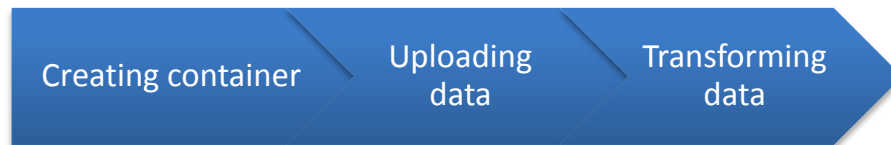
3.3.2 Transforming files

Figure 21 Transforming files

In order to use the transformer, you have to create a container. Container sends the data to the transformer as described on the section 2. Containers are called "Widgets" on the Fusepool P3 platform and using them is really simple. Next we are going to create a container for our transformer.
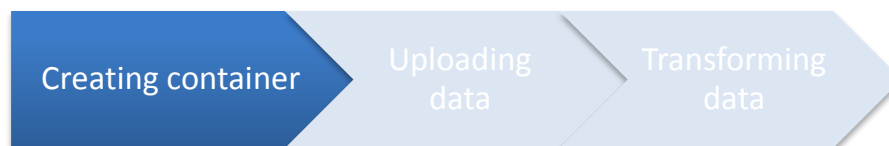
3.3.2.1 Creating container

Figure 22 Creating a container

Go to "Publishing" page. You need to create a widget in order to use your transformer. Click "New widget".

→ You can leave the "Widget type" as T-LDPC
→ Give it a name you desire
→ Select your transformer from the dropdown list
→ Transformer URI will be filled automatically
→ You can give description if needed
→ Save widget

Figure 23 Creating a widget

Your widget should appear now on the page. Now you can use your widget to transform files.

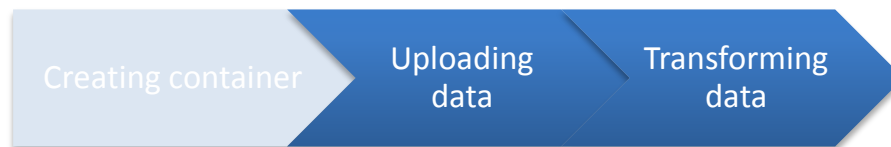### 3.3.2.2 Uploading and transforming data

Figure 24 Uploading and transforming the data

 Note: Uploading and transforming data are included on the same phase because user doesn't have to do anything regarding the transformation.
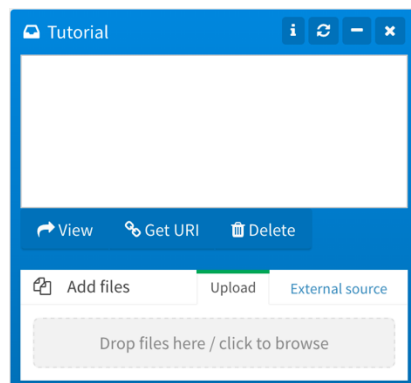
The widget looks like this:

Figure 25 Empty widget

Next you need to upload a file to the container to get it transformed.

→ Click section "Drop files here / click browse"
→ Select a file you want to transform and click "Open". The file should now appear on the list.
→ Click refresh from upper right corner of the widget and your transformed file should appear there. Sometimes the whole page has to be refreshed in order to see the file. Transformation may take a while depending on the size of the data set.
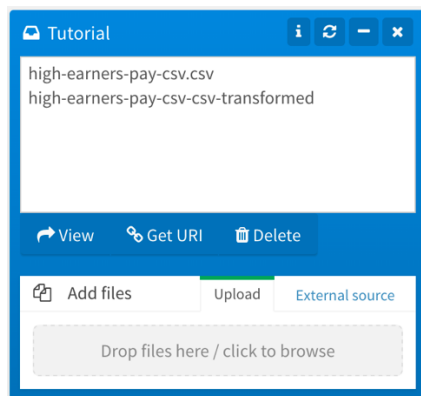
Figure 26 Transformed file

If everything went right, you should be able to see the same as above. There you can see the original file and the transformed file.

You can view, copy the URI of your file or delete your file.
To make sure that your data is now stored as triples in the data set, choose transformed file and click "View". It takes you to the data set.

3.4 Batchrefine transformer

3.4.1 Creating the transformer

We can create the BatchRefine transformer without OpenRefine. Exception here is that you need ready mapping file which you link to the transformer manually. You can use the same mapping file that we created on the previous part.

→ Go to the transformers page
→ On the lower right side choose Batchrefine UI from the Factories section and click Open factory GUI
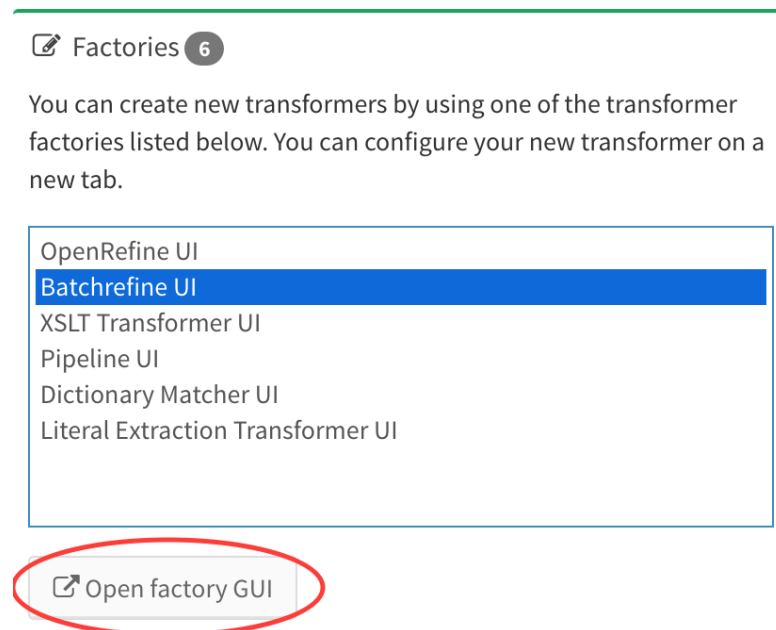
→ Then you need to copy the URI of your JSON file that we generated from the csv file on the previous chapter. In order to use the JSON file, you need to upload it to some server. We are using GitHub. If you don't want to upload it by yourself, you can use this: https://raw.githubusercontent.com/OskariRomanoff/Tutorial-Fp3/master/mapping.json

→ Copy it on the "URI of the Refine JSON" field. Click "Generate transformer" to generate the transformer and then click "Register". You don't have to care about other links.
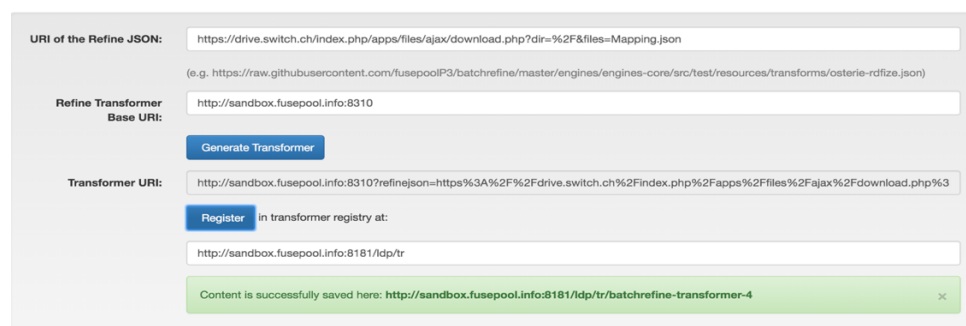

Figure 27 Generating BatchRefine transformer

The page should look like this when you have successfully generated transformer. Now you can close this tab and go back to transformers page.

3.4.2 Naming and testing the BatchRefine transformer

NOTE: The testing of the BatchRefine transformer doesn't work at the moment. But if everything is alright on your configurations you can transform files using widgets. Anyway it is recommended to rename it in order to recognize your own transformer.

1. Now you can see your transformer on the transformer list named "Batchrefine transformer". It is usually the last one.

   a. Change its name to recognize it later by clicking Rename and typing new name.
   b. You can test that it is working by clicking "Test"
      → Click "Browse", choose your file that you would like to transform and click Open.
      → Once you have the file uploaded, select input format as "text/csv" and preferred output format as "text/turtle".
      → Click test transformer and you can see as a result the transformed file. Now you can close this window.
      After these steps it should look like this:

Figure 28 Test phase of the BatchRefine transformer

3.4.3 Transforming files

The transformation works in the same way as on chapter 3.2.2. You just need to make sure that you choose the right transformer when creating a widget.

4 Debugging

4.1 Introduction

We have encountered some problems while using the platform and here are few ways to try to avoid and solve them.

4.2 Transformer problems

We have faced a problem when you upload a file to the widget and it won´t get transformed but the original file is still there. So you can´t be sure where the problem is because the platform doesn't inform you if there has been an error. To make sure that the transformer has the right configurations, you can check, if it has received the information from widget by checking the network calls. Below is tutorial how to check the calls.

We are using Google chrome as a web browser.

1. Open The Chrome menu ☰ on upper right corner
2. Select **Tools → Developer tools,** it opens developer´s tool window
3. Then open **Network** tab in order to see network calls etc.
4. From name column you need to find right call then click it and choose **Response** tab to make sure that it is the right call. Response should look like this.
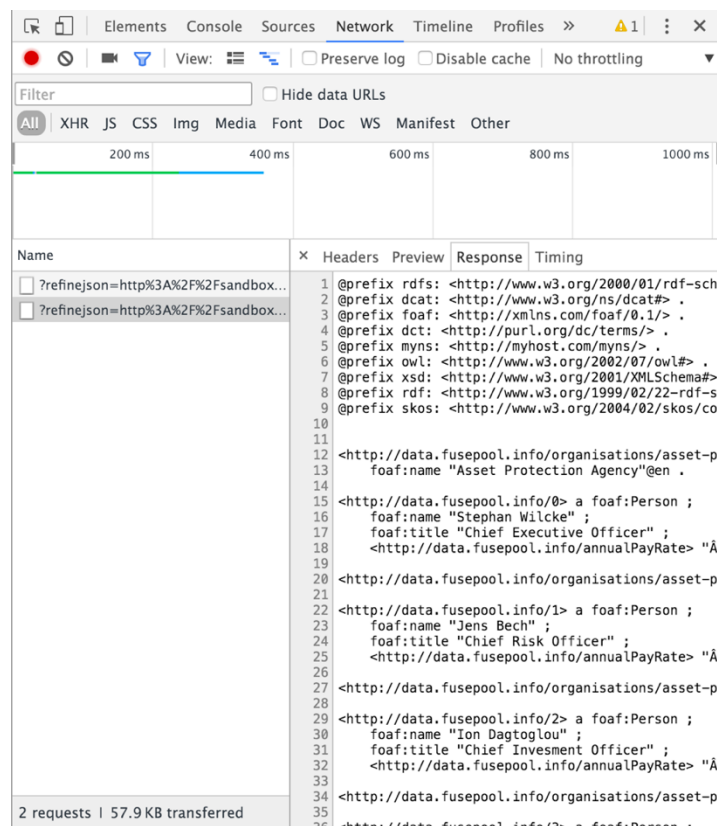
5. When you have found the right call you need to get its cURL.
   First click the right call with your second mouse button. Then choose from dropdown list **Copy as cURL.**
6. Copy and paste the cURL to your console → Press **Enter**.

7. Now you should have the file listed as RDF

4.3 Browser problems

When uploading a file to the widget the file format changed by itself. For example, a csv file turned into xls and on some cases to bin format. It is a browser problem and we recommend using google chrome as a browser.