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.

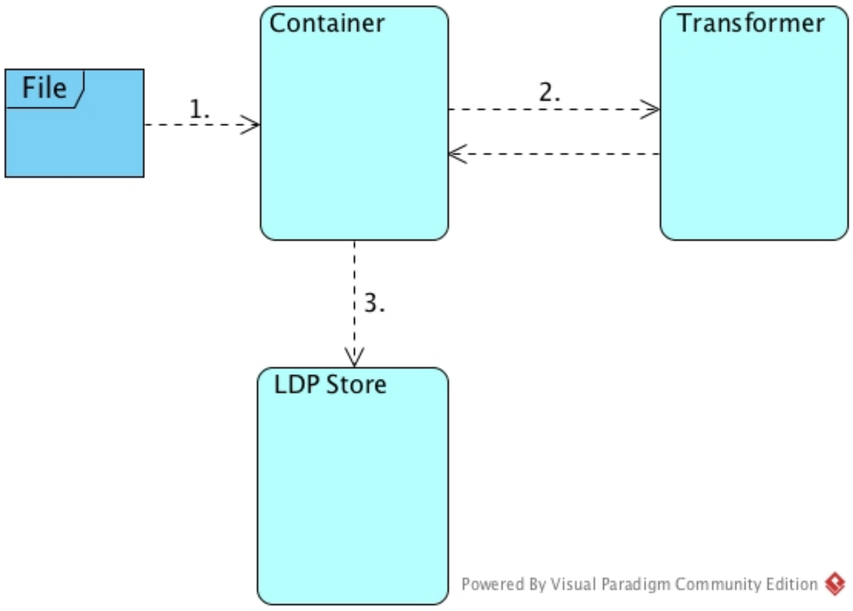


Figure 1 The main structure of the Platform

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?

The purpose of this tutorial example is, that you learn how to transform files by using the Fusepool P3 platform. In the end you will have a CSV file transformed into RDF and you can query the data by using SPARQL query language. To transform files, we are going to generate a BatchRefine transformer. There are two ways to generate the BatchRefine transformer. We will go through both ways.
To put it briefly, in general, the process includes these steps:

1. Get a data file you want to transform (CSV)
2. Generate a transformer (BatchRefine)
3. Generate a container and link it to your transformer (Widgets)
4. Transform the file and store it in the triple store
5. Use SPARQL to query the data

There is a more detailed description of the process as you go further on this tutorial. If you want to check it now you can find it from chapter: 3.3

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. The example file "high earners pay", is about persons with high income in UK. The data file describes employees name, job title, grade, organization he/she is working for and their annual pay rate.
On this example we want to be able to query employees name, job title, organization and annual pay rate. To do this, we choose these fields for our mapping file. The idea of the mapping file is, that we can tell which data rows we want to display as triples in the data store.
We will get on these in more detail as you proceed on the tutorial.

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.

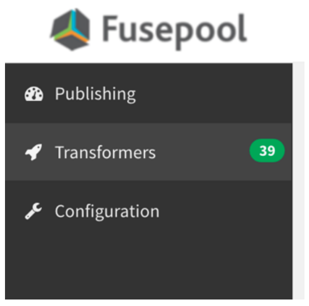
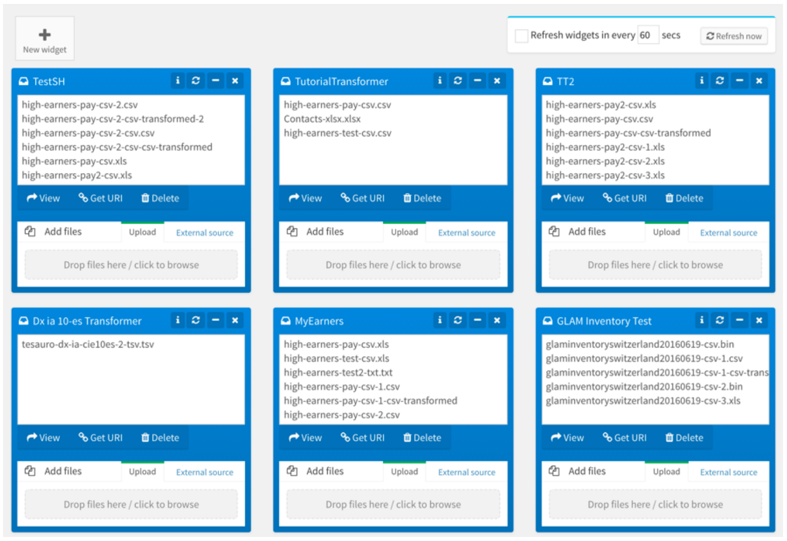


Figure 2 The main pages of the Fusepool P3 Platform

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.



TestSH
high-earners-pay-csv-2.csv
high-earners-pay-csv-2-csv-transformed-2
high-earners-pay-csv-2-csv.csv
high-earners-pay-csv-2-csv-csv-transformed
high-earners-pay-csv.xls
high-earners-pay2-csv.xls

TutorialTransformer
high-earners-pay-csv.csv
Contacts-xlsx.xlsx
high-earners-test-csv.csv

TT2
high-earners-pay2-csv.xls
high-earners-pay-csv.csv
high-earners-pay-csv-csv-transformed
high-earners-pay2-csv-1.xls
high-earners-pay2-csv-2.xls
high-earners-pay2-csv-3.xls

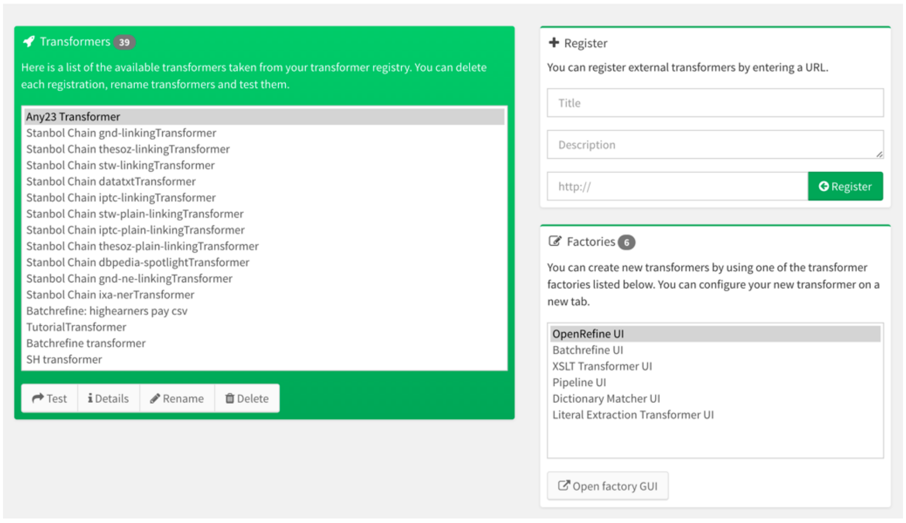
Dx ia 10-es Transformer
tesauro-dx-ia-cie10es-2-tsv.tsv

MyEarners
high-earners-pay-csv.xls
high-earners-test-csv.xls
high-earners-test2-txt.txt
high-earners-pay-csv-1.csv
high-earners-pay-csv-1-csv-transformed
high-earners-pay-csv-2.csv

GLAM Inventory Test
glaminventoryswitzerland20160619-csv.bin
glaminventoryswitzerland20160619-csv-1.csv
glaminventoryswitzerland20160619-csv-1-csv-trans
glaminventoryswitzerland20160619-csv-2.bin
glaminventoryswitzerland20160619-csv-3.xls

Figure 3 Publishing page

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.



Transformers 39
Here is a list of the available transformers taken from your transformer registry. You can delete each registration, rename transformers and test them.

Any23 Transformer
Stanbol Chain gnd-linkingTransformer
Stanbol Chain thesoz-linkingTransformer
Stanbol Chain stw-linkingTransformer
Stanbol Chain datatxtTransformer
Stanbol Chain iptc-linkingTransformer
Stanbol Chain stw-plain-linkingTransformer
Stanbol Chain iptc-plain-linkingTransformer
Stanbol Chain thesoz-plain-linkingTransformer
Stanbol Chain dbpedia-spotlightTransformer
Stanbol Chain gnd-ne-linkingTransformer
Stanbol Chain ixa-nerTransformer
Batchrefine: highearners pay csv
TutorialTransformer
Batchrefine transformer
SH transformer

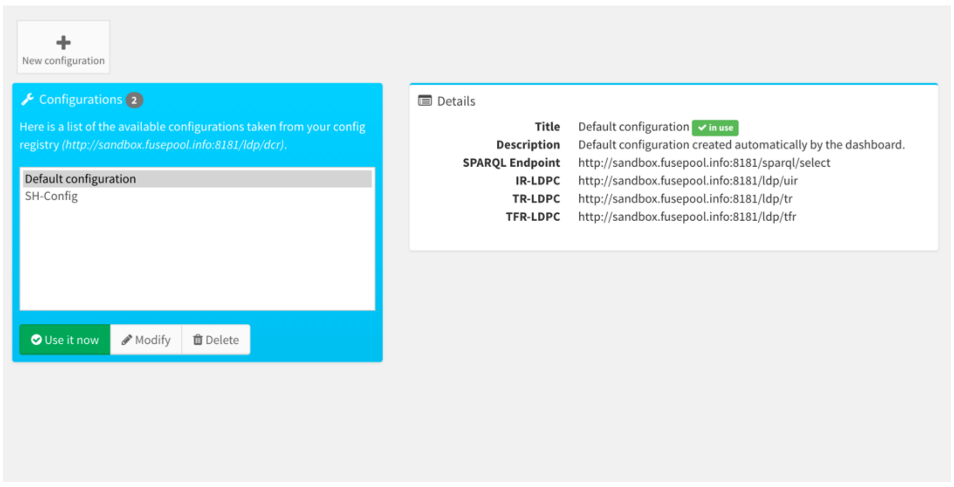
Register
You can register external transformers by entering a URL.

Factories 6
You can create new transformers by using one of the transformer factories listed below. You can configure your new transformer on a new tab.

OpenRefine UI
Batchrefine UI
XSLT Transformer UI
Pipeline UI
Dictionary Matcher UI
Literal Extraction Transformer UI

Figure 4 Transformers page

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.



Here is a list of the available configurations taken from your config registry *(http://sandbox.fusepool.info:8181/ldp/dcr)*.

Default configuration
SH-Config

Title	Default configuration
Description	Default configuration created automatically by the dashboard.
SPARQL Endpoint	http://sandbox.fusepool.info:8181/sparql/select
IR-LDPC	http://sandbox.fusepool.info:8181/ldp/uir
TR-LDPC	http://sandbox.fusepool.info:8181/ldp/tr
TFR-LDPC	http://sandbox.fusepool.info:8181/ldp/tfr

Figure 5 Configuration page

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

Other tutorials:

Before you continue on this tutorial, Fusepool p3 Platform has many other features. If you want to check for the other tutorials of Fusepool P3 platform, please check-out their YouTube channel:

https://www.youtube.com/channel/UC_WGjcr81uETsEk6jmA8ChQ

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
→ Testing transformer

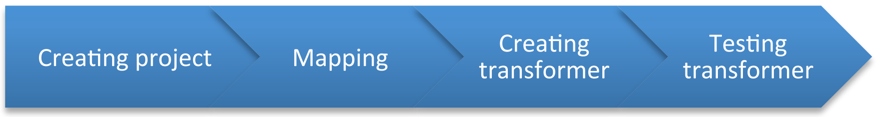


Figure 6 Process of generating a transformer

Using the transformer:

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

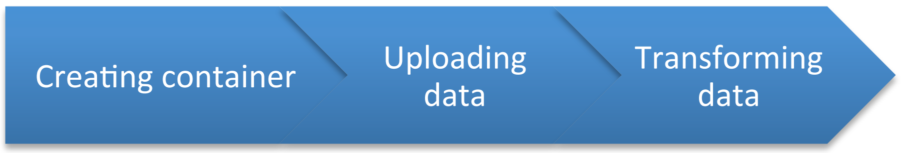


Figure 7Process of using the transformer

3.3.1 Generating the transformer

3.3.1.1 Creating the project

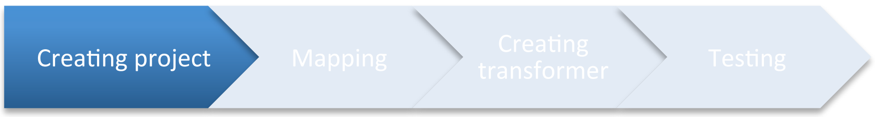


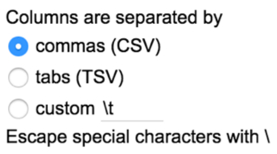
Figure 8 Creating project

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:

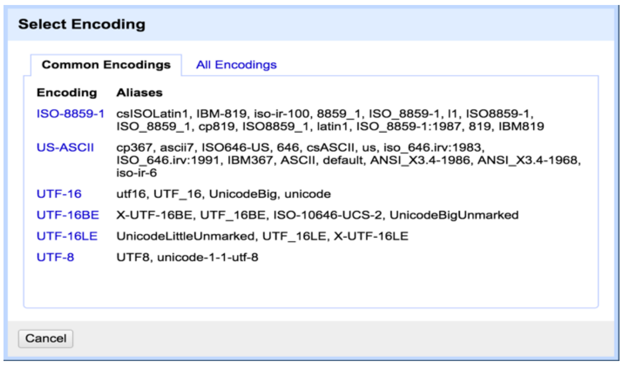
Make sure that columns are separated by commas



Columns are separated by
[✓ commas (CSV)]
tabs (TSV)
custom \t
Escape special characters with \

Figure 9 Column separation

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



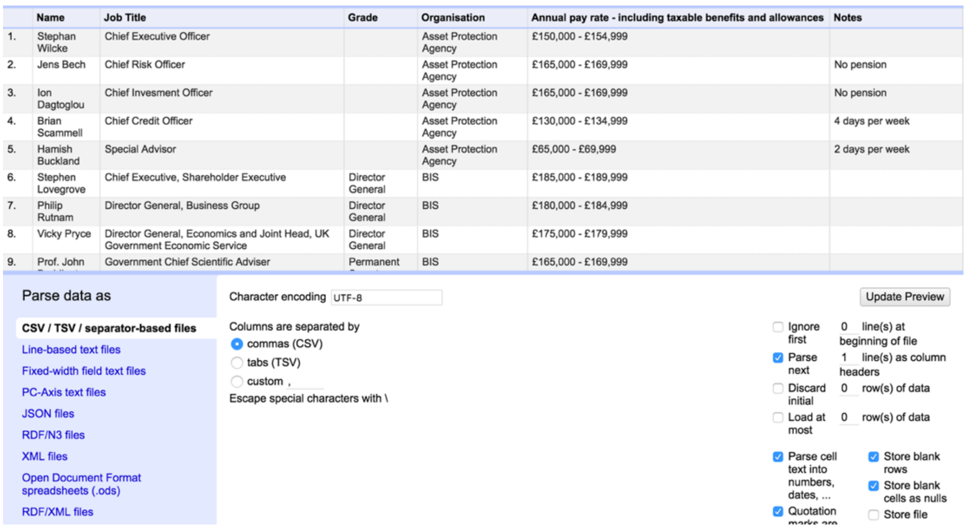
Select Encoding

Common Encodings | All Encodings

Encoding	Aliases
ISO-8859-1	csISOLatin1, IBM-819, iso-ir-100, 8859_1, ISO_8859-1, l1, ISO8859-1, ISO_8859_1, cp819, ISO8859_1, latin1, ISO_8859-1:1987, 819, IBM819
US-ASCII	cp367, ascii7, ISO646-US, 646, csASCII, us, iso_646.irv:1983, ISO_646.irv:1991, IBM367, ASCII, default, ANSI_X3.4-1986, ANSI_X3.4-1968, iso-ir-6
UTF-16	utf16, UTF_16, UnicodeBig, unicode
UTF-16BE	X-UTF-16BE, UTF_16BE, ISO-10646-UCS-2, UnicodeBigUnmarked
UTF-16LE	UnicodeLittleUnmarked, UTF_16LE, X-UTF-16LE
UTF-8	UTF8, unicode-1-1-utf-8

Figure 10 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	

Parse data as

CSV / TSV / separator-based files
Line-based text files
Fixed-width field text files
PC-Axis text files
JSON files
RDF/N3 files
XML files
Open Document Format spreadsheets (.ods)
RDF/XML files

Character encoding UTF-8

Update Preview

Columns are separated by
● commas (CSV)
○ tabs (TSV)
○ custom ,
Escape special characters with \

☐ Ignore first 0 line(s) at beginning of file
✓ Parse next 1 line(s) as column headers
☐ Discard initial 0 row(s) of data
☐ Load at most 0 row(s) of data
✓ Parse cell text into numbers, dates, ...
✓ Quotation
✓ Store blank rows
✓ Store blank cells as nulls
☐ Store file

Figure 11 Overview of the upcoming project

Now click Create Project from upper right corner.

3.3.1.2 Creating the mapping file

Creating project
Mapping
Creating transformer
Testing

Figure 12 Mapping

As described in the beginning of this chapter, with a mapping file, we tell the transformer which data we want to display in the triple store. 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”

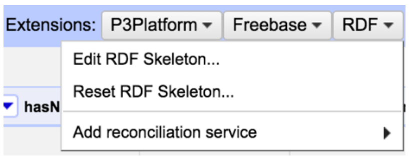

Extensions: P3Platform Freebase RDF
Edit RDF Skeleton...
Reset RDF Skeleton...
hasN
Add reconciliation service

Figure 13 Edit rdf skeleton

It will open a window like this:

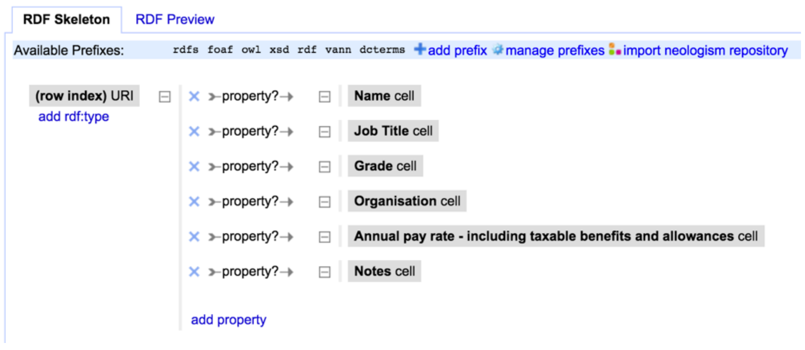

RDF Skeleton
RDF Preview
Available Prefixes: rdfs foaf owl xsd rdf vann dcterms add prefix manage prefixes import neologism repository
(row index) URI
add rdf:type
property?
Name cell
Job Title cell
Grade cell
Organisation cell
Annual pay rate - including taxable benefits and allowances cell
Notes cell
add property

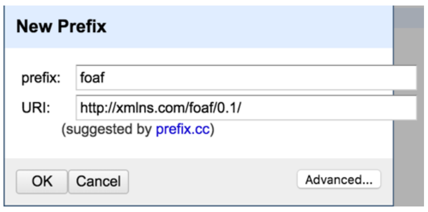
Figure 14 Unedited rdf skeleton

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.


New Prefix

prefix: foaf

URI: http://xmlns.com/foaf/0.1/

(suggested by prefix.cc)

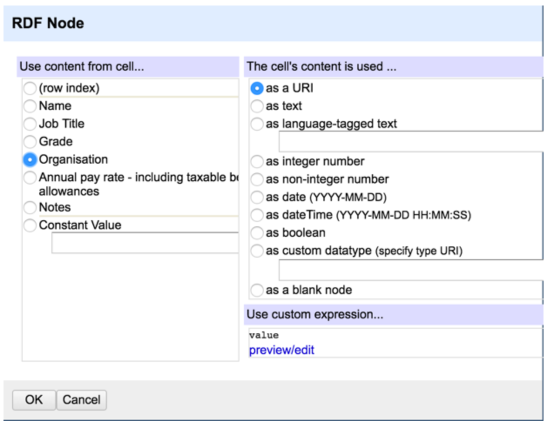
Figure 15 Adding prefix

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.



RDF Node

Use content from cell...
- (row index)
- Name
- Job Title
- Grade
- [✓ Organisation]
- Annual pay rate - including taxable be allowances
- Notes
- Constant Value

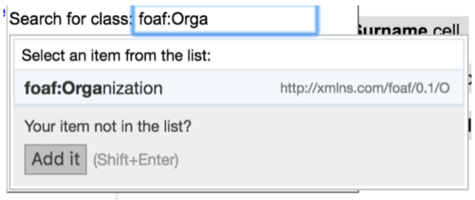
The cell's content is used ...
- [✓ as a URI]
- as text
- as language-tagged text
- as integer number
- as non-integer number
- as date (YYYY-MM-DD)
- as dateTime (YYYY-MM-DD HH:MM:SS)
- as boolean
- as custom datatype (specify type URI)
- as a blank node

Use custom expression...
value
preview/edit

OK Cancel

Figure 16 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.



Search for class: foaf:Orga

Select an item from the list:

foaf:Organization http://xmlns.com/foaf/0.1/O

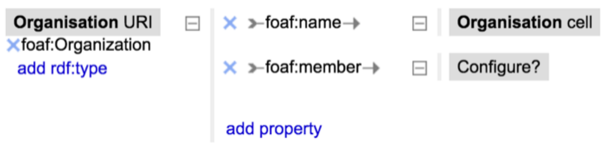
Your item not in the list?

Add it (Shift+Enter)

Figure 17 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



Organisation URI
foaf:Organization
add rdf:type

foaf:name → Organisation cell
foaf:member → Configure?
add property

Figure 18 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.

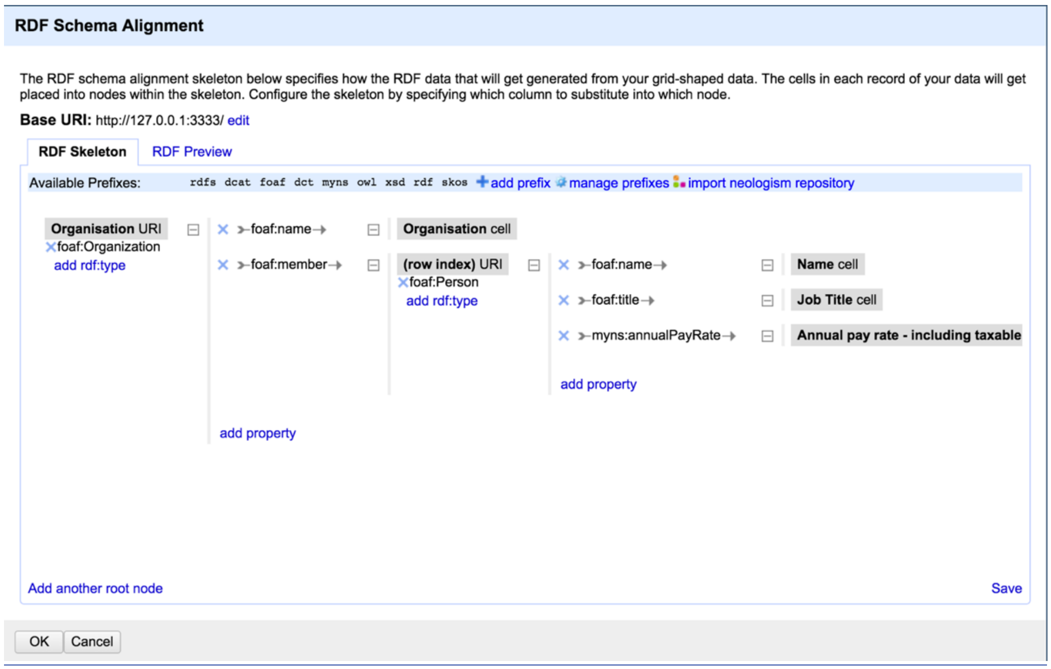
Properties: foaf:name, foaf:title and myns:annualPayRate

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
myns:annualPayRate → Annual pay rate - including taxable benefits and allowances as text

The skeleton should look like this:



RDF Schema Alignment

The RDF schema alignment skeleton below specifies how the RDF data that will get generated from your grid-shaped data. The cells in each record of your data will get placed into nodes within the skeleton. Configure the skeleton by specifying which column to substitute into which node.

Base URI: http://127.0.0.1:3333/ edit

Figure 19 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

Creating project
Mapping
Creating transformer
Testing

Figure 20 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:

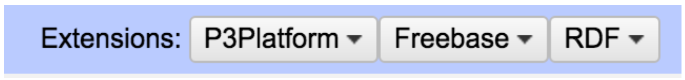
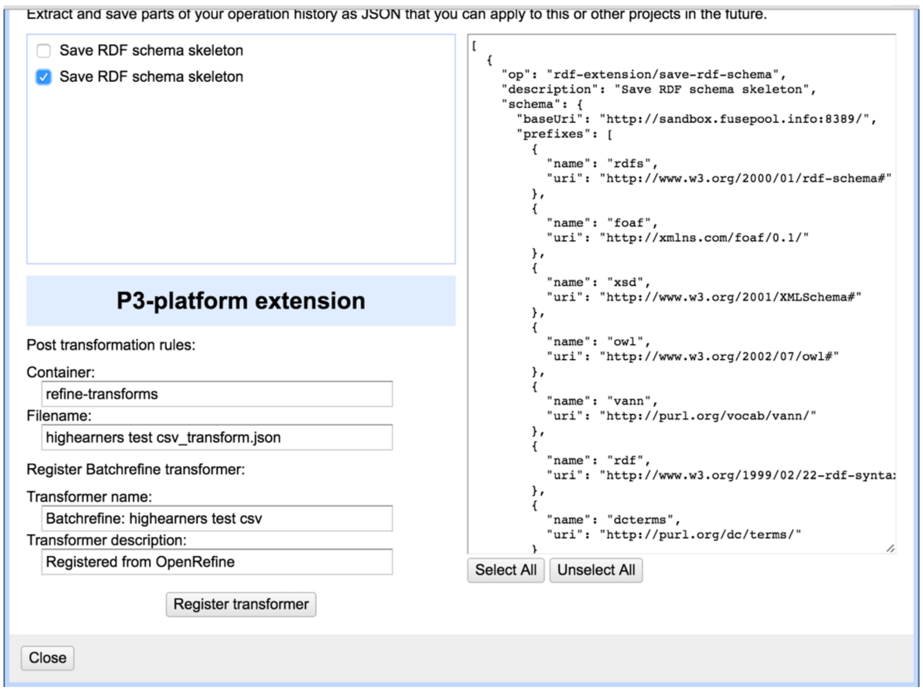


Figure 21 Extension bar

→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.



Extract and save parts of your operation history as JSON that you can apply to this or other projects in the future.

☐ Save RDF schema skeleton
☑ Save RDF schema skeleton

P3-platform extension

Post transformation rules:

Container: refine-transforms

Filename: highearners test csv_transform.json

Register Batchrefine transformer:

Transformer name: Batchrefine: highearners test csv

Transformer description: Registered from OpenRefine

```
[
  {
    "op": "rdf-extension/save-rdf-schema",
    "description": "Save RDF schema skeleton",
    "schema": {
      "baseUri": "http://sandbox.fusepool.info:8389/",
      "prefixes": [
        {
          "name": "rdfs",
          "uri": "http://www.w3.org/2000/01/rdf-schema#"
        },
        {
          "name": "foaf",
          "uri": "http://xmlns.com/foaf/0.1/"
        },
        {
          "name": "xsd",
          "uri": "http://www.w3.org/2001/XMLSchema#"
        },
        {
          "name": "owl",
          "uri": "http://www.w3.org/2002/07/owl#"
        },
        {
          "name": "vann",
          "uri": "http://purl.org/vocab/vann/"
        },
        {
          "name": "rdf",
          "uri": "http://www.w3.org/1999/02/22-rdf-synta
        },
        {
          "name": "dcterms",
          "uri": "http://purl.org/dc/terms/"
        }
```

Figure 22 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

Creating project
Mapping
Creating transformer
Testing transformer

Figure 23 Testing the transformer

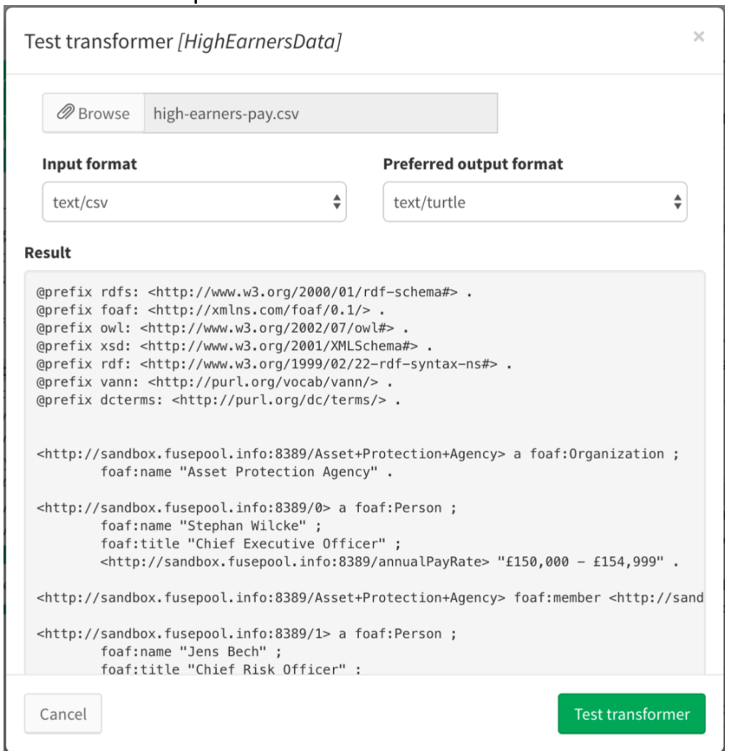
To make sure your transformer works, it is good to test it before starting to transform files.

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:


Test transformer *[HighEarnersData]*

Browse high-earners-pay.csv

Input format text/csv

Preferred output format text/turtle

Result

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix vann: <http://purl.org/vocab/vann/> .
@prefix dcterms: <http://purl.org/dc/terms/> .


<http://sandbox.fusepool.info:8389/Asset+Protection+Agency> a foaf:Organization ;
        foaf:name "Asset Protection Agency" .

<http://sandbox.fusepool.info:8389/0> a foaf:Person ;
        foaf:name "Stephan Wilcke" ;
        foaf:title "Chief Executive Officer" ;
        <http://sandbox.fusepool.info:8389/annualPayRate> "£150,000 - £154,999" .

<http://sandbox.fusepool.info:8389/Asset+Protection+Agency> foaf:member <http://sand

<http://sandbox.fusepool.info:8389/1> a foaf:Person ;
        foaf:name "Jens Bech" ;
        foaf:title "Chief Risk Officer" ;
```

Figure 24 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

Creating container
Uploading data
Transforming data

Figure 25Transforming files

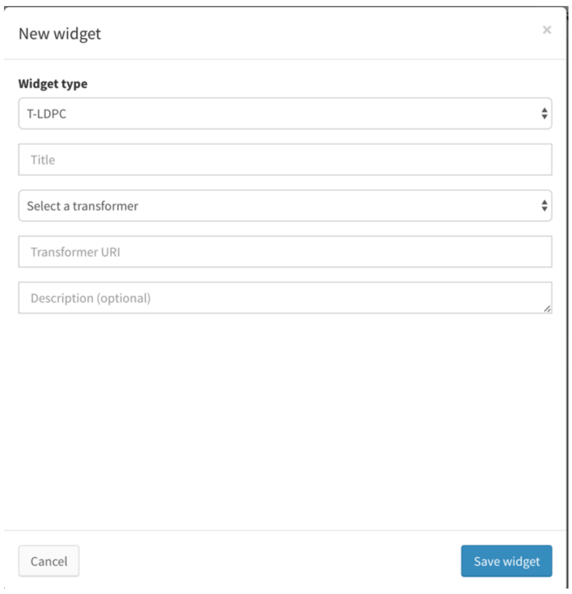
In order to use the transformer, you have to create a container. Container sends the data to the transformer as described on the section 2. Containers are called “Widgets” on the Fusepool P3 platform and using them is really simple. Next we are going to create a container for our transformer.

3.3.2.1 Creating container

Figure 26 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


New widget

×

Widget type

T-LDPC

Title

Select a transformer

Transformer URI

Description (optional)

Cancel

Save widget

Figure 27 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

Creating container
Uploading data
Transforming data

Figure 28 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:

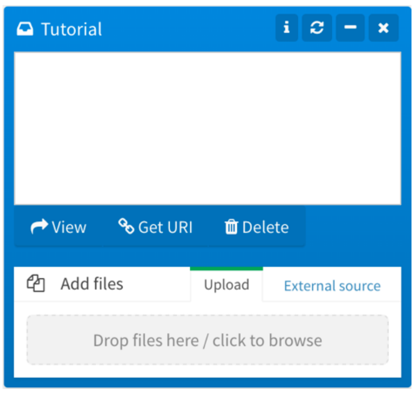
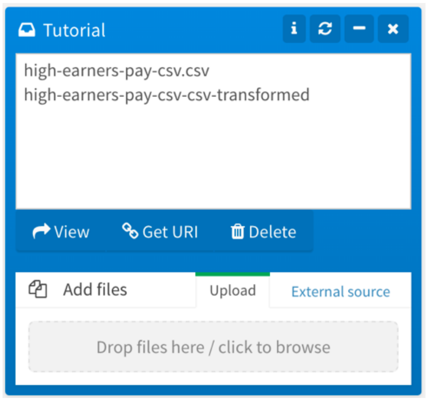


Figure 29 Empty widget

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.



high-earners-pay-csv.csv
high-earners-pay-csv-csv-transformed

Figure 30 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.

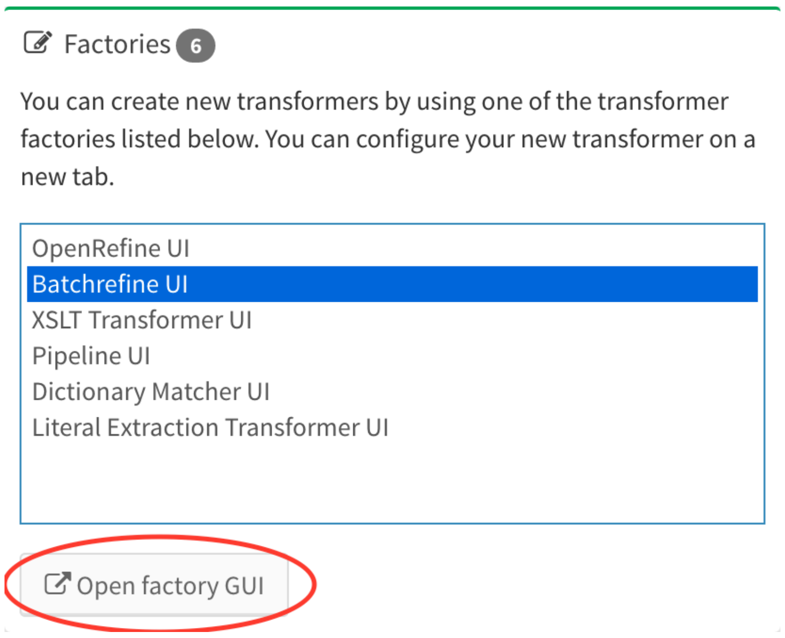
You can view, copy the URI of your file or delete your file.
To make sure that your data is now stored as triples in the data set, choose transformed file and click "View". It takes you to the data set.

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.

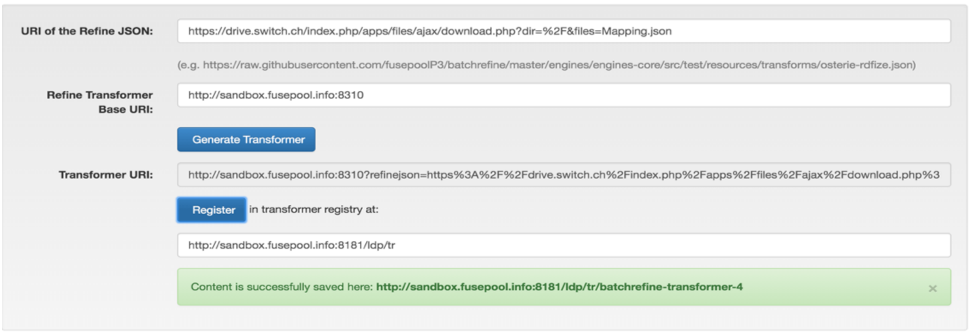
→ Go to the transformers page
→ On the lower right side choose Batchrefine UI from the Factories section and click Open factory GUI



You can create new transformers by using one of the transformer factories listed below. You can configure your new transformer on a new tab.

→ 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.


URI of the Refine JSON: https://drive.switch.ch/index.php/apps/files/ajax/download.php?dir=%2F&files=Mapping.json

(e.g. https://raw.githubusercontent.com/fusepoolP3/batchrefine/master/engines/engines-core/src/test/resources/transforms/osterie-rdfize.json)

Refine Transformer Base URI: http://sandbox.fusepool.info:8310

Transformer URI: http://sandbox.fusepool.info:8310?refinejson=https%3A%2F%2Fdrive.switch.ch%2Findex.php%2Fapps%2Ffiles%2Fajax%2Fdownload.php%3

Register in transformer registry at: http://sandbox.fusepool.info:8181/ldp/tr

Content is successfully saved here: **http://sandbox.fusepool.info:8181/ldp/tr/batchrefine-transformer-4**

Figure 31 Generating BatchRefine transformer

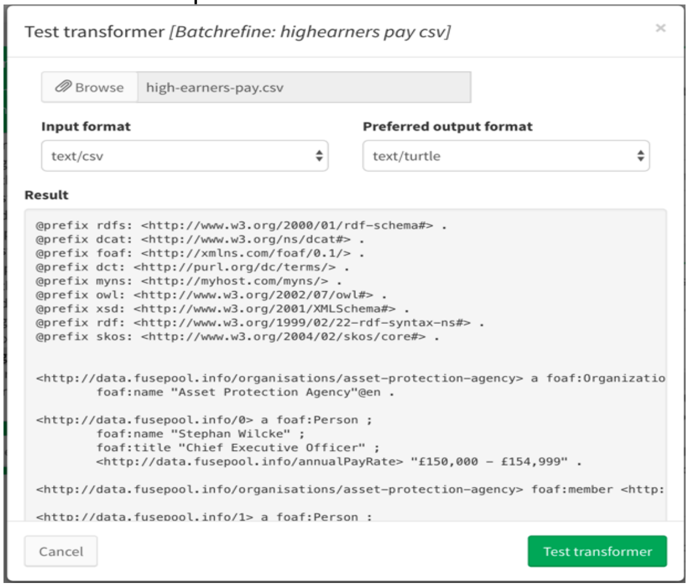
The page should look like this when you have successfully generated transformer. Now you can close this tab and go back to transformers page.

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:


Test transformer *[Batchrefine: highearners pay csv]*

Browse high-earners-pay.csv

Input format: text/csv

Preferred output format: text/turtle

Result

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix myns: <http://myhost.com/myns/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .


<http://data.fusepool.info/organisations/asset-protection-agency> a foaf:Organizatio
        foaf:name "Asset Protection Agency"@en .

<http://data.fusepool.info/0> a foaf:Person ;
        foaf:name "Stephan Wilcke" ;
        foaf:title "Chief Executive Officer" ;
        <http://data.fusepool.info/annualPayRate> "£150,000 - £154,999" .

<http://data.fusepool.info/organisations/asset-protection-agency> foaf:member <http:

<http://data.fusepool.info/1> a foaf:Person ;
```

Figure 32 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.

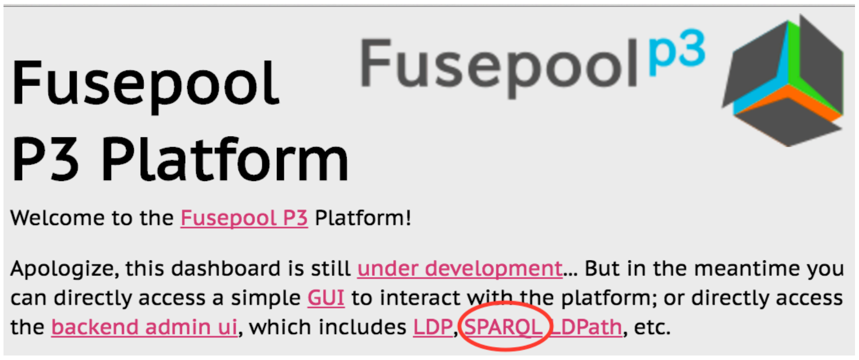
3.5 Querying data by using SPARQL

Now we have the data stored as triples in the database. The next step is to query this data and display it as tables. If you are not familiar with SPARQL, it is recommended to learn about it before continuing. You can find a good tutorial from here:

http://www.cambridgesemantics.com/semantic-university/sparql-by-example

For querying we are using Fusepool´s SPARQL query endpoint. Which you can find from here:

http://sandbox.fusepool.info:8181/fusepool/public/dashboard.html and click SPARQL link.



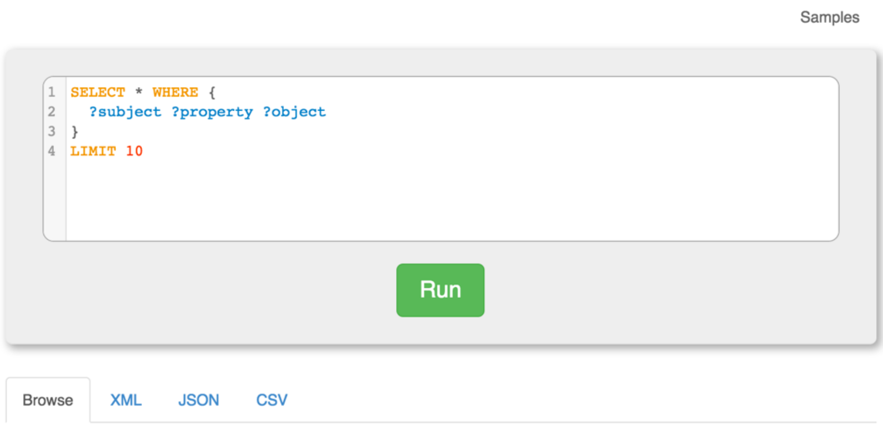
Fusepool
P3 Platform

Welcome to the Fusepool P3 Platform!

Apologize, this dashboard is still under development... But in the meantime you can directly access a simple GUI to interact with the platform; or directly access the backend admin ui, which includes LDP, SPARQL, LDPath, etc.

Figure 33 Link to SPARQL endpoint

When you enter the page you can see the field were we add the query. By default, it looks like this



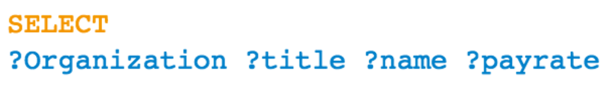
```
SELECT * WHERE {
  ?subject ?property ?object
}
LIMIT 10
```

Figure 34 SPARQL query endpoint

So as we determined on the mapping file, our triple store includes organization, employees name, job title and annual pay rate. On this example we are going to query and display them all.

So to get started with the query. We need the prefixes from our mapping file. You can copy them by going to your project in OpenRefine, then choose “Edit the RDF skeleton” like when we edited the RDF skeleton and from there “RDF Preview” tab. Now copy the prefixes to the SPARQL endpoint. Before continuing delete the “@” marks from every prefix.

1. So first we determine the objects for our query. We choose every data row from our triple store, so the SELECT section looks like this:

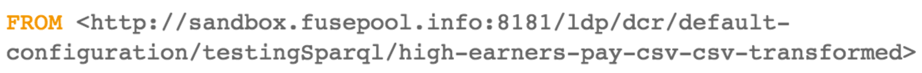


```
SELECT
?Organization ?title ?name ?payrate
```

Figure 35 Select

These can be named anyway you want but it is good to give them a describing name in order to recognise the data.

2. Then we determine the source of the data we query by using FROM:

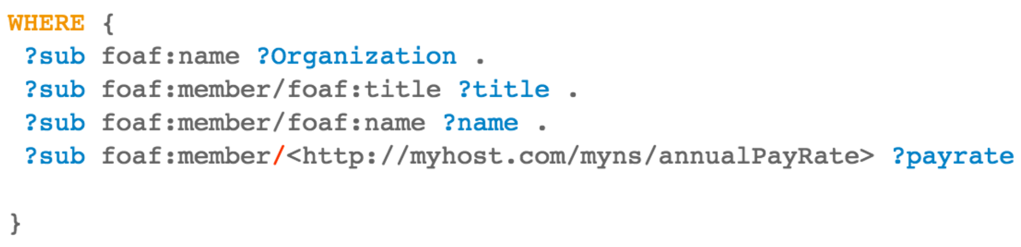


```
FROM <http://sandbox.fusepool.info:8181/ldp/dcr/default-
configuration/testingSparql/high-earners-pay-csv-csv-transformed>
```

Figure 36 From

You can get the source by going to your widget and opening the transformed file. Then copy the URL of your triple store (URL of the web page).

3. Next we determine the actual data we want to display. Under the WHERE, we add the triples “?sub” “?pred” “?obj” . It should look like this:



```
WHERE {
 ?sub foaf:name ?Organization .
 ?sub foaf:member/foaf:title ?title .
 ?sub foaf:member/foaf:name ?name .
 ?sub foaf:member/<http://myhost.com/myns/annualPayRate> ?payrate

}
```

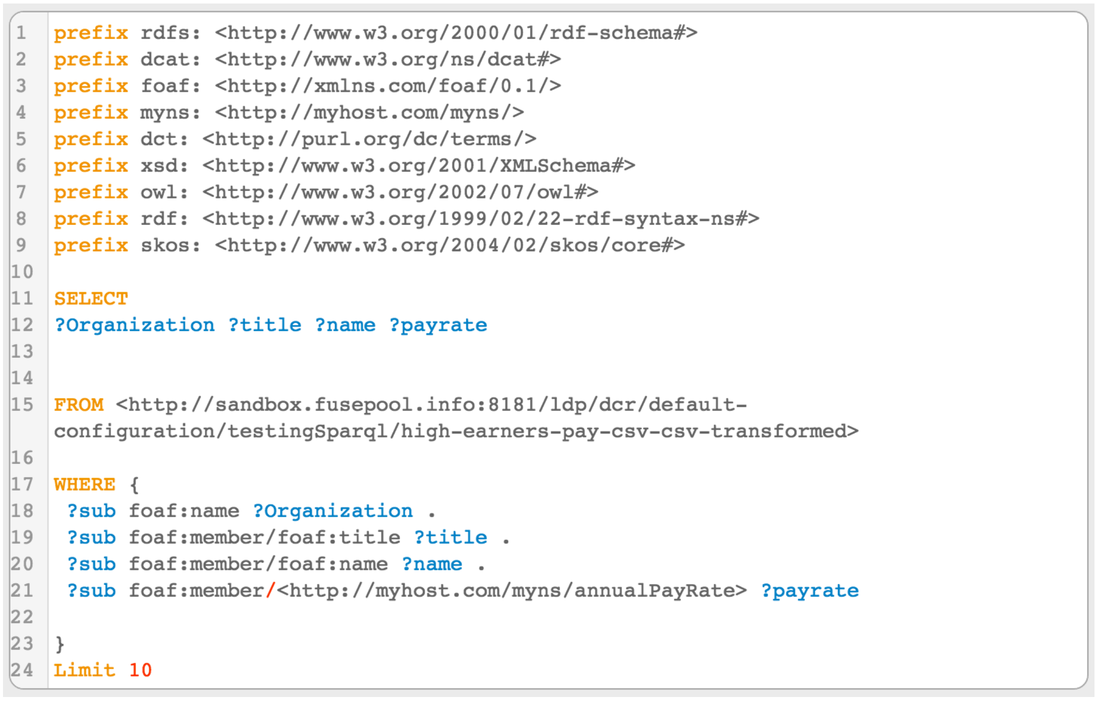
Figure 37 Where

We don’t need to touch the subject so we leave it just “?sub”. For predicate you need to determine which value (property) your subject has “foaf:name”. Then we name it as “?Organization”.

Because the title, name and annual pay rate are properties of the sub class Person, we need to determine the “path” from Organization to the properties of the Person class. It happens by adding “/” between the properties.

Don’t forget to add the dot “.” after every triple (except the last one).

Now the query is ready and it should look like this:



```
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix dcat: <http://www.w3.org/ns/dcat#>
prefix foaf: <http://xmlns.com/foaf/0.1/>
prefix myns: <http://myhost.com/myns/>
prefix dct: <http://purl.org/dc/terms/>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix skos: <http://www.w3.org/2004/02/skos/core#>

SELECT
?Organization ?title ?name ?payrate


FROM <http://sandbox.fusepool.info:8181/ldp/dcr/default-
configuration/testingSparql/high-earners-pay-csv-csv-transformed>

WHERE {
 ?sub foaf:name ?Organization .
 ?sub foaf:member/foaf:title ?title .
 ?sub foaf:member/foaf:name ?name .
 ?sub foaf:member/<http://myhost.com/myns/annualPayRate> ?payrate

}
Limit 10
```

Figure 38 Ready query

In the end you can add limitation for your results. So only 10 results are displayed.

No you can run the query and the results are displayed below of the web page.

This example query is really simple. You can try more complex ones as you like.

4 Debugging

4.1 Introduction

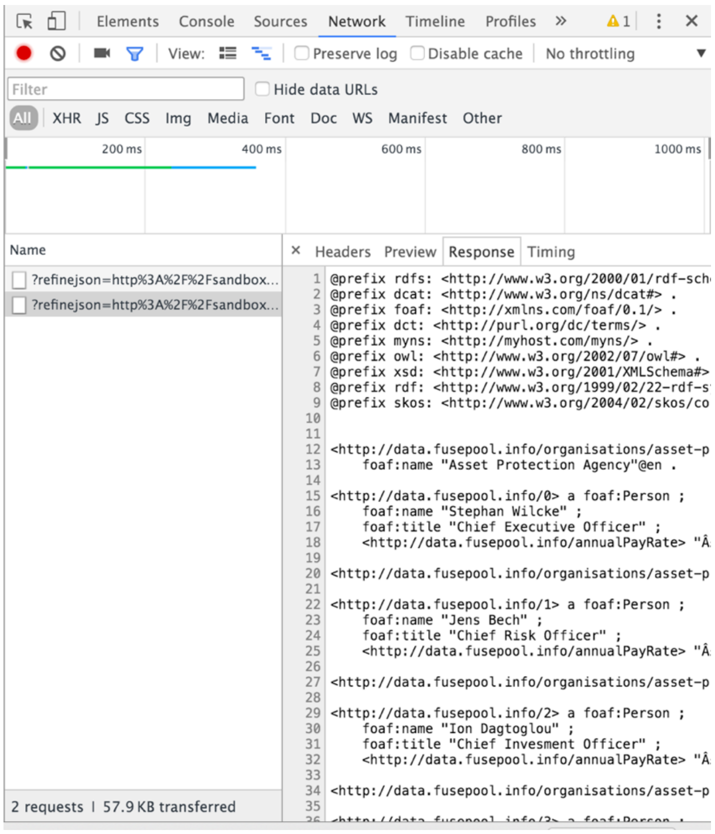
We have encountered some problems while using the platform and here are few ways to try to avoid and solve them.

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.



```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-sch
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix myns: <http://myhost.com/myns/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-s
@prefix skos: <http://www.w3.org/2004/02/skos/co


<http://data.fusepool.info/organisations/asset-p
    foaf:name "Asset Protection Agency"@en .

<http://data.fusepool.info/0> a foaf:Person ;
    foaf:name "Stephan Wilcke" ;
    foaf:title "Chief Executive Officer" ;
    <http://data.fusepool.info/annualPayRate> "Â

<http://data.fusepool.info/organisations/asset-p

<http://data.fusepool.info/1> a foaf:Person ;
    foaf:name "Jens Bech" ;
    foaf:title "Chief Risk Officer" ;
    <http://data.fusepool.info/annualPayRate> "Â

<http://data.fusepool.info/organisations/asset-p

<http://data.fusepool.info/2> a foaf:Person ;
    foaf:name "Ion Dagtoglou" ;
    foaf:title "Chief Invesment Officer" ;
    <http://data.fusepool.info/annualPayRate> "Â

<http://data.fusepool.info/organisations/asset-p

<http://data.fusepool.info/3> a foaf:Person ;
```

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 and a platform problem. We have faced this issue with every browser on Windows computers.
The problem is caused by the Internet media type (MIME) standard. It is an identifier for file formats and format contents transmitted on the internet. To put it simple and briefly, the transformation of the file type happens, because browsers use “hard coded lists”, which determine the media types. CSV is not specified on this list, so this error occurs.

5 Abbildungsverzeichnis

No table of figures entries found.

6 Tabellenverzeichnis

7 Glossar

Auinweon Et ut aut isti repuditis qui ium	7
Batnwpe Et ut aut isti repuditis qui ium	9
Cowoll	

Et ut aut isti repuditis qui ium 11

8 Literaturverzeichnis

Literatureintrag
Autorname, Autorvorname, Buchtitel, Verlag, Ort, Ausgabe, Jahr 7

Literatureintrag
Autorname, Autorvorname, Buchtitel, Verlag, Ort, Ausgabe, Jahr 9

Literatureintrag
Autorname, Autorvorname, Buchtitel, Verlag, Ort, Ausgabe, Jahr 11

9 Anhang

Et ut aut isti repuditis qui ium nonsecturia quis incientiae laborem elliquis et quatur, sitiur aut od moluptatur aut ea conseque peri sim erro essequisit remporia dem et landi dest, cone poris quunt volecab ipidero quatur ad quibusamus.

10Versionskontrolle

Version	Datum	Beschreibung	Autor
0.1	26.02.2013	Dokument erstellt	Peter Muster
0.2	13.03.2013	Dokument überarbeitet	Anna Meier
1.0	21.05.2013	Dokument fertiggestellt	Peter Muster