Enriching Data (Singapore edition)

In this exercise we are going to use data from opencorporates to enrich a dataset containing company data.

The dataset for this exercise was located by using an advanced google search containing the following: “site:sg filetype:xls contracts”. This resulted in locating the following dataset:

www.citygas.com.sg/pdf/AGENT_LISTING.xls

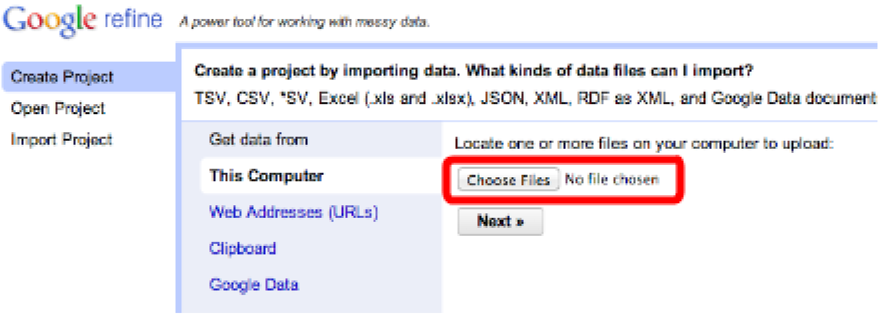
This appears to be a listing of agents of the company that was created in November 2006 (according to the file properties).

The question we are going to try and answer is:

“How many of the agents are still active, how many are dissolved and how many have been liquidated?”

Step 1 - Import into Refine

Simply import this data into refine as a new project. Ensure that the formatting looks correct before clicking create project.



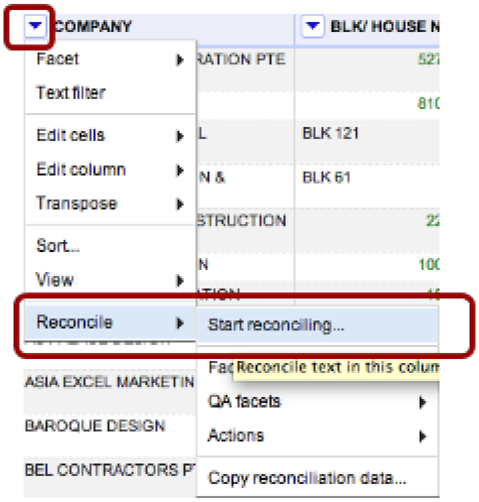
Create a project by importing data. What kinds of data files can I import?
TSV, CSV, *SV, Excel (.xls and .xlsx), JSON, XML, RDF as XML, and Google Data document

Locate one or more files on your computer to upload:

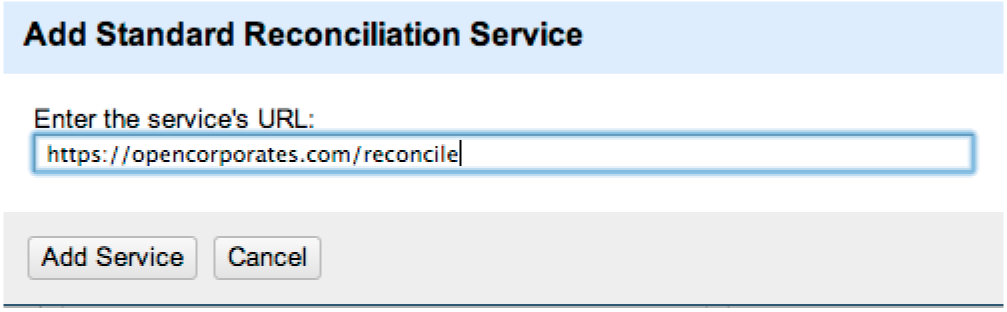
Step 2 - Reconcile and link

With the data now imported we have a column listing company names in plain text.

In order to link the data to publicly available data we are going to use the opencorporates reconciliation API for Singapore. From you company name column drop down select the **Start Reconcile** option from the **Reconcile** menu. You will note that opencorporates is not yet available as an option so a new **standard reconciliation service** will need to be added. The service url is as follows, please ensure you copy this exactly (including the http**s**):



```
https://opencorporates.com/reconcile/sg
```

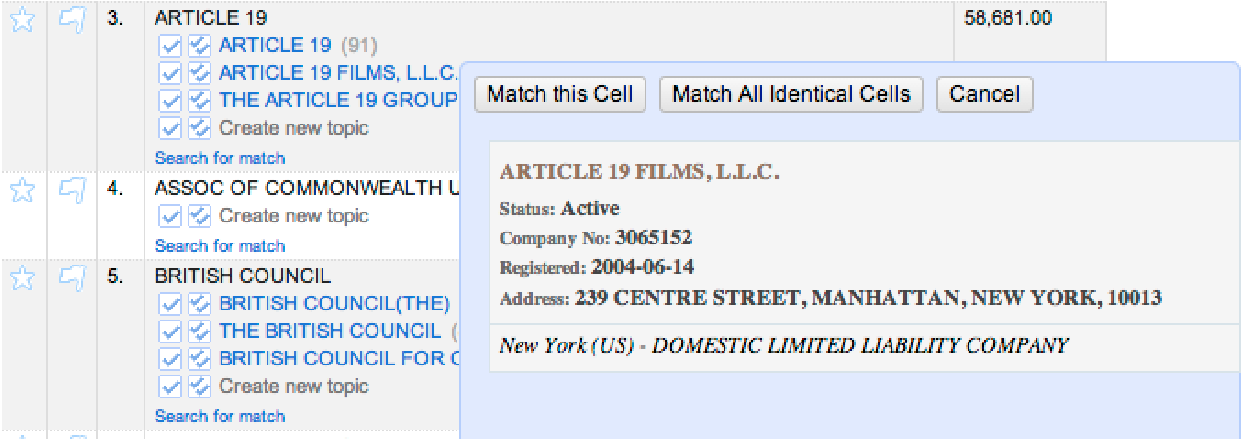


Add Standard Reconciliation Service

Enter the service's URL: https://opencorporates.com/reconcile

Once done, select this service and click **Start Reconciling.**

When complete you will notice that each company may have several matches in opencorporates, with each scored differently. You can click on each option to see a brief overview of that company and then tick which you believe is the correct option.



3. ARTICLE 19 — 58,681.00
- ARTICLE 19 (91)
- ARTICLE 19 FILMS, L.L.C.
- THE ARTICLE 19 GROUP
- Create new topic

4. ASSOC OF COMMONWEALTH U
- Create new topic

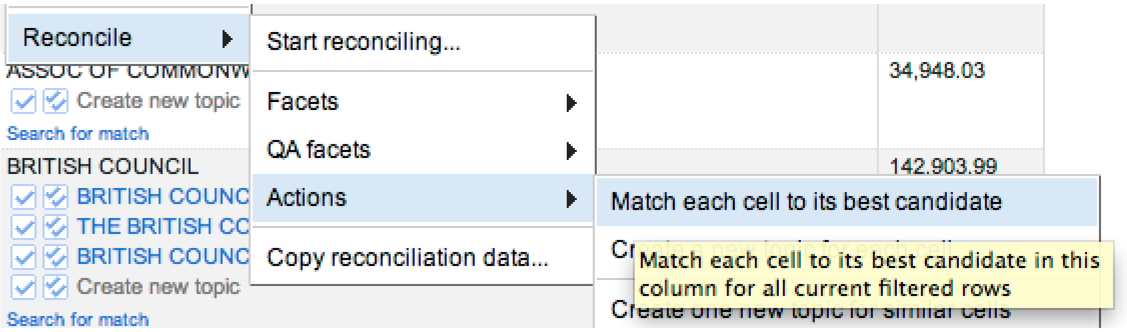
5. BRITISH COUNCIL
- BRITISH COUNCIL(THE)
- THE BRITISH COUNCIL (
- BRITISH COUNCIL FOR C
- Create new topic

ARTICLE 19 FILMS, L.L.C.

Status: Active
Company No: 3065152
Registered: 2004-06-14
Address: 239 CENTRE STREET, MANHATTAN, NEW YORK, 10013

New York (US) - DOMESTIC LIMITED LIABILITY COMPANY

Rather than manually process the entire dataset you can also filter to high or low quality matches using the facet on the left. You can also just choose to match all companies against their best candidate using via the **match** option in the **reconcile** menus **actions**.



ASSOC OF COMMONW — 34,948.03
- Create new topic

BRITISH COUNCIL — 142,903.99
- BRITISH COUNC
- THE BRITISH CC
- BRITISH COUNC
- Create new topic

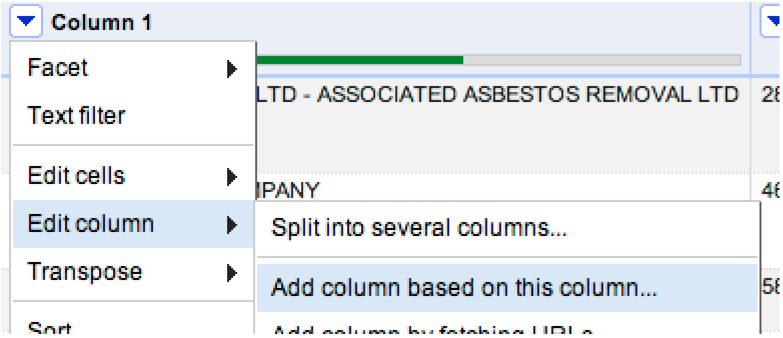
Match each cell to its best candidate in this column for all current filtered rows

Reconcile
Start reconciling...
Facets
QA facets
Actions
Copy reconciliation data...
Match each cell to its best candidate
Create a new topic for each cell
Create one new topic for similar cells
Search for match

Step 4 - Reveal the URI

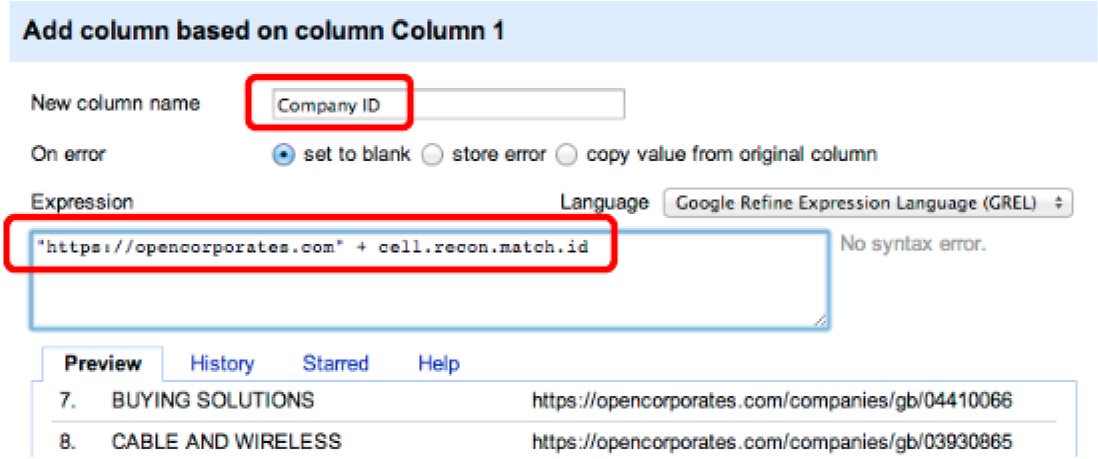
Once you have matched as many companies as possible you will notice that clicking on the company will now take you to that companies page in opencorporates. This URL is also the company URI (identifier) and we can also reveal this in the dataset as follows.

From the company column drop down select **Add column based upon this column**.


Column 1
Facet
Text filter
Edit cells
Edit column
Transpose
Sort
LTD - ASSOCIATED ASBESTOS REMOVAL LTD
28
IPANY
46
Split into several columns...
Add column based on this column...
58
Add column by fetching URLs

In the box that appears, give the column a name of **Company ID** and type the following in the expression box:

```
“https://opencorporates.com” + cell.recon.match.id
```



Add column based on column Column 1

New column name: Company ID

On error: ● set to blank ○ store error ○ copy value from original column

Expression

Language: Google Refine Expression Language (GREL)

```
"https://opencorporates.com" + cell.recon.match.id
```

No syntax error.

Preview | History | Starred | Help

7.	BUYING SOLUTIONS	https://opencorporates.com/companies/gb/04410066
8.	CABLE AND WIRELESS	https://opencorporates.com/companies/gb/03930865

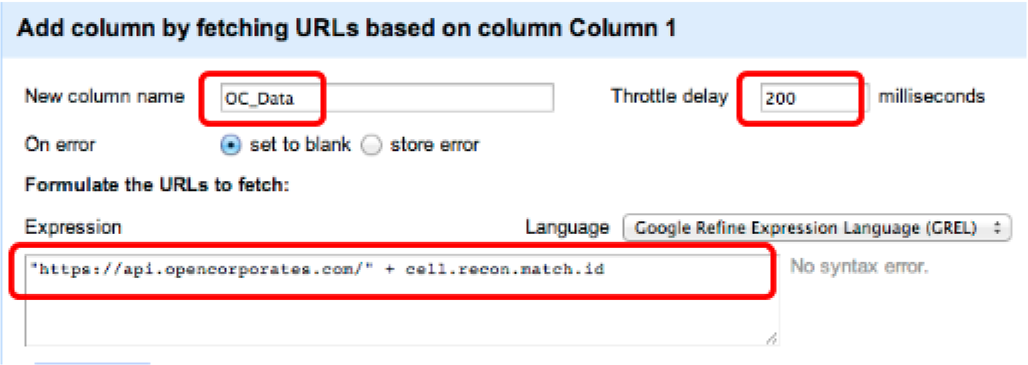
When done click OK. We could now export this dataset as CSV, XML, RDF etc and it would contain a URI from which further data could be obtained. This is a simple but effective example of linked data and linked csv.

Step 5 - Get the data

We now have a link to the opercorporates page for each company that lists data about that company. It is the current status that we need in order to answer the original question. In order to fetch it for the companies we shall use the opencorporates API in order to download the data for each company.

This stage involves adding a **column by fetching URLs**. Although we could fetch the URI from step 4 using content negotiation, refine does not understand HTTP redirection thus we would not get the data back from the URI. For this reason it is necessary to cut a corner and download the data directly from where the redirect would have sent refine.

```
“https://api.opencorporates.com” + cell.recon.match.id
```



Add column by fetching URLs based on column Column 1

New column name: OC_Data

Throttle delay: 200 milliseconds

On error: ● set to blank ○ store error

Formulate the URLs to fetch:

Expression — Language: Google Refine Expression Language (GREL)

```
"https://api.opencorporates.com/" + cell.recon.match.id
```

No syntax error.

Make sure you name the column and turn down the **throttle delay** such that this process completes in reasonable time. The throttle delay is there so we don’t overwhelm an API with requests.

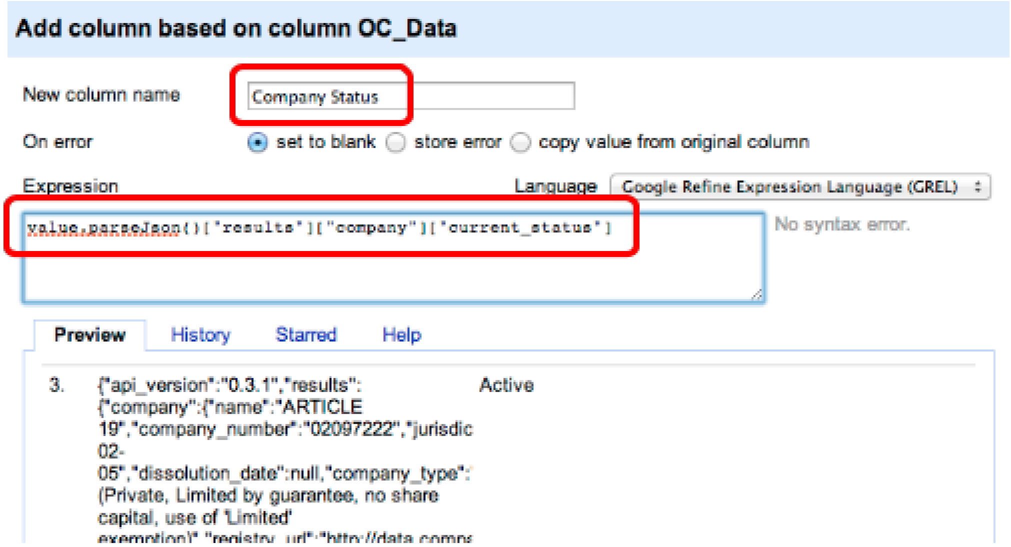
Once done click OK and you should end up with a column full of JSON data from which we need to pick out the **company status**.

In order to see what the JSON data looks like why not take one of the returned values and paste it into the JSON validator at **jsonlint.com**.

Step 6 - Extract the company status

One last column to add based upon the data column this time with the following expression to parse the JSON data:

```
value.parseJson()["results"]["company"]["current_status"]
```



Add column based on column OC_Data

New column name: Company Status

On error: ● set to blank ○ store error ○ copy value from original column

Expression — Language: Google Refine Expression Language (GREL)

```
value.parseJson()["results"]["company"]["current_status"]
```

No syntax error.

```
3.  {"api_version":"0.3.1","results":                Active
    {"company":{"name":"ARTICLE
    19","company_number":"02097222","jurisdic
    02-
    05","dissolution_date":null,"company_type":
    (Private, Limited by guarantee, no share
    capital, use of 'Limited'
    exemption)","registry_url":"http://data.comp
```

Once complete you might want to **collapse** the data column so that you can see more rows on the screen. It is then possible to **apply a text on the new column** to find out how many companies are still active, liquidated, dissolved or other.

Extension Exercises

Why not try extracting data other than current status from the opencorporates data?

Can you link to any other data available from opencorporates or suppliers of other reconciliation endpoints?

How about a visualisation of the data?