Messy Data? Clean it up with OpenRefine!

CARLI Created Content Committee workshop
Session 2

Greer Martin
Metadata Technologies Librarian
Loyola University Chicago
February 17, 2021

Agenda

- OpenRefine review
- Reconciliation
- APIs
- Exporting data

Create a new project

1. Download and install OpenRefine 3.4.1: https://openrefine.org/download.html

2. Double click on OpenRefine icon, then go to http://127.0.0.1:3333/

3. Download repository_data.txt from http://bit.ly/carli-openrefine2

4. Create new project with repository_data.txt

OpenRefine Review

- OpenRefine runs in your browser and saves automatically. “Open Project” to return to it.
- OpenRefine is good for editing **complete** datasets. Bad for adding to or reorganizing datasets (use Excel).
- You get infinite undos!
- Import and export a variety of formats (tabular, XML, JSON, etc.).
- Exporting is required to get data “out.”
- OpenRefine documentation: https://docs.openrefine.org/

Reconciliation

Reconciliation

Reconciliation is a **semi-automated** process that matches your data in OpenRefine against an external dataset

Why use it?

It’s another way to normalize your data

Match data to authoritative headings or labels (VIAF, ORCID, Getty authorities)

Add external data to your dataset (URIs, geographic coordinates, etc.)

Reconciliation Service

A web service that conforms to the Reconciliation Service API standards

Not all data sources provide a reconciliation web service (LC currently does not)

If they have an open API, anyone can create one.

Some reconciliation services created by the organization themselves (such as Getty), some created by volunteers.

List of reconciliation services: https://reconciliation-api.github.io/testbench/

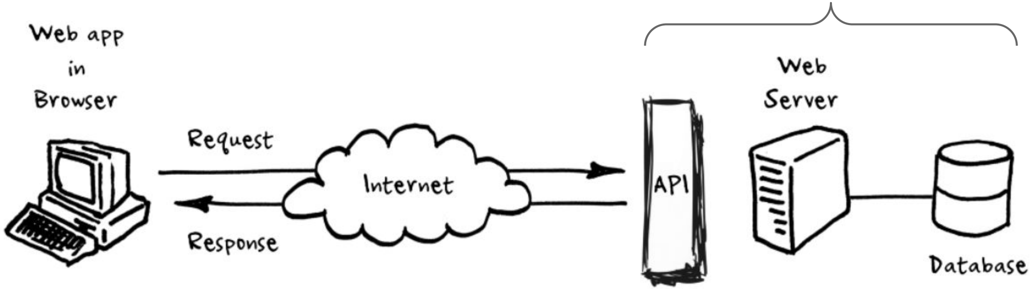
APIs

Application Programming Interface

An API is the **code** that governs access to data on a server.

Designed for applications to communicate with one another.

Returns raw, machine-readable data



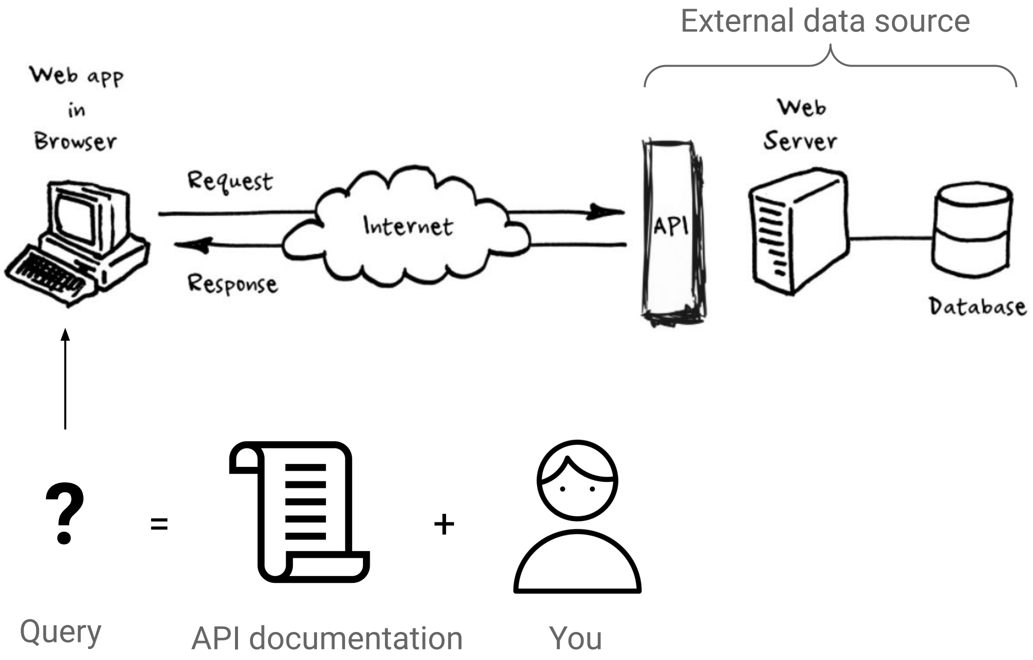
External data source

Application Programming Interface

An API is the **code** that governs access to data on a server.

Designed for applications to communicate with one another.

Returns raw, machine-readable data



External data source

Query

API documentation

You

Forming an HTTP API query

http://fast.oclc.org/searchfast/fastsuggest?&query=Driver&queryIndex=suggestall&suggest=autoSubject&queryReturn=suggestall%2Cauth%2Ctype%2Cidroot%2Ctag&rows=1

Base URL	http://fast.oclc.org/searchfast/fastsuggest?	
Query parameters	query=Driver	Search for the term “Driver”
	queryIndex=suggestall	Search against all FAST headings
	queryReturn=suggestall, tag, auth, type, idroot	Return these data fields
	rows=1	Return max 1 heading

Putting it into OpenRefine

Our HTTP API request:

http://fast.oclc.org/searchfast/fastsuggest?&query=Driver&queryIndex=suggestall&suggest=autoSubject&queryReturn=suggestall%2Cauth%2Ctype%2Cidroot%2Ctag&rows=1

Becomes:

[Keyword1 column] Edit column > Add column by fetching URLs

Enter:

"http://fast.oclc.org/searchfast/fastsuggest?&query="+value+"&queryIndex=suggestall&suggest=autoSubject&queryReturn=suggestall%2Cauth%2Ctype%2Cidroot%2Ctag&rows=1"

Now, query=value parameter will automatically populate with the value in each row!

Parsing data

Most APIs will return data as JSON or XML

Edit Column > Add column based on this column

Enter GREL expression:
value.parseJson()["response"]["docs"][0]["auth"]

Exporting data

Templating Export

Use Export > Template to export JSON, modify template to export as XML

JSON template

```
{
    "Date submitted" : {{jsonize(cells["Date submitted"].value)}},
}
```

XML template

```
<record>
  <dc:date>{{jsonize(cells["Date submitted"].value)}}</dc:date>
</record>
```

Thanks!

OpenRefine documentation: https://docs.openrefine.org/

API diagram, “What exactly is an APi?” by Perry Eisign
https://medium.com/@perrysetgo/what-exactly-is-an-api-69f36968a41f

person by Jimi Lim from the Noun Project

Document by Marek Polakovic from the Noun Project