Introduction to OpenRefine

Owen Stephens
Felix Lohmeier

Using these slides

These slides were developed by Owen Stephens (owen@ostephens.com) on behalf of the British Library.

Introductions

Outline for today

- Introductions and outline (~10 minutes)
- The basics (~60 minutes)

BREAK (approx. 14:10)

- Transforming data (~30 minutes)
- Exporting data (~10 minutes)
- Introduction to arrays and comparators (~30 minutes)

BREAK (approx. 15:40)

- Linking to other data pt 1 (~45 minutes)

BREAK (approx. 16:45)

- Linking to other data pt 2 (~45 minutes)
- Contributing to OpenRefine (~30 minutes)

BREAK (approx. 18:00)

- Free time to experiment and ask questions (dependent on time remaining)

Finish (approx. 19:00)

“a tool for working with messy data”

http://openrefine.org

OpenRefine can help when…

- you have data in a simple tabular format
- there are inconsistencies in how the data is formatted
- there are inconsistencies in where data appears
- there are inconsistencies in terminology used in the data

OpenRefine can help you…

- Get an overview of a data set
- Resolve inconsistencies in a data set
- Help you split data up into more granular parts
- Match local data up to other data sets
- Enhance a data set with data from other sources

Getting help

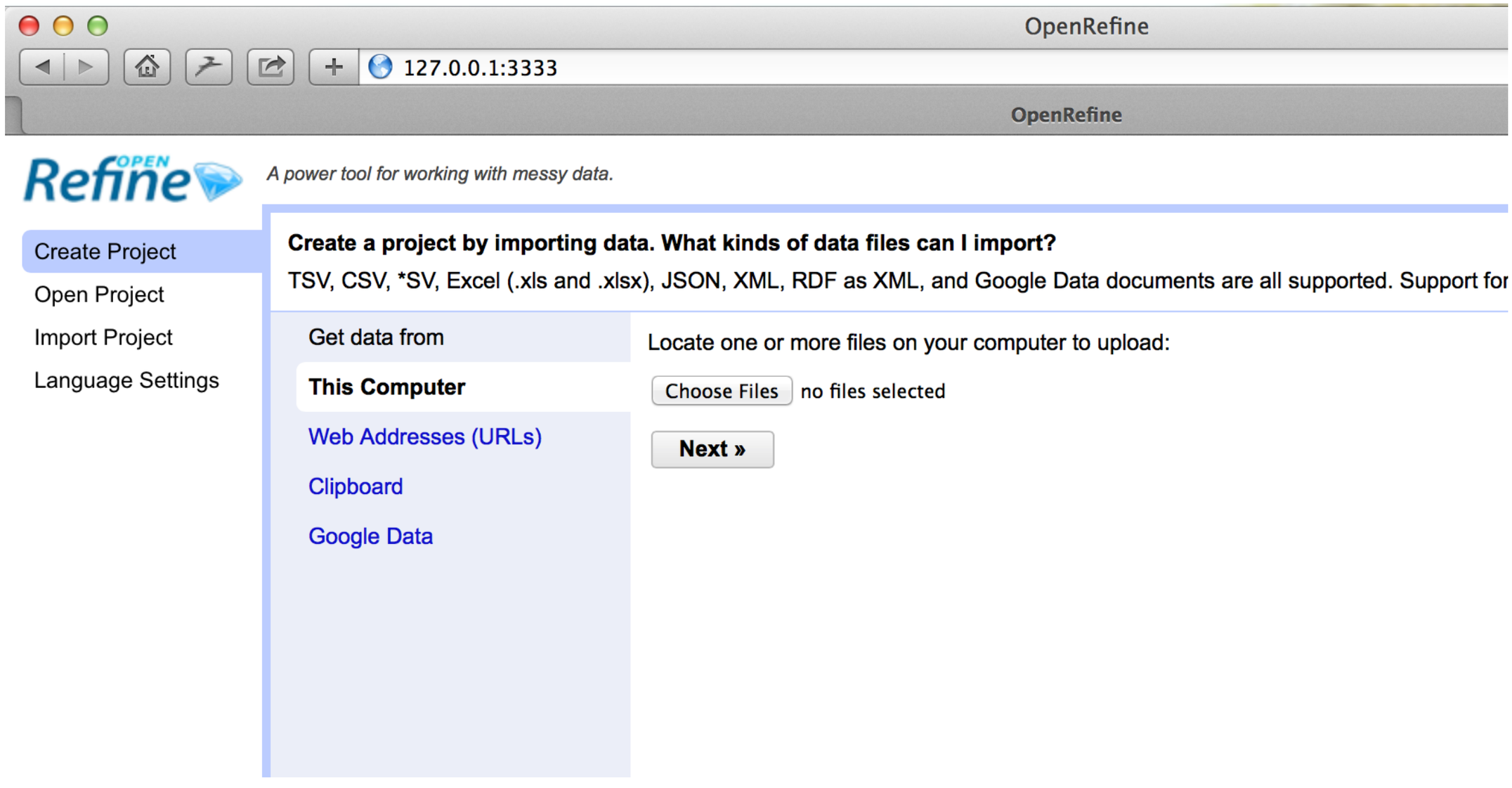
- The OpenRefine Wiki: https://github.com/OpenRefine/OpenRefine/wiki
- The OpenRefine mailing list and forum: http://groups.google.com/d/forum/openrefine
- LibraryCarpentry OpenRefine Lesson: https://librarycarpentry.org/lc-open-refine/
- The ‘Free your metadata’ site: http://freeyourmetadata.org/

http://bit.ly/training-data

download

doaj-article-sample.csv

Start using OpenRefine



http://127.0.0.1:3333

Hands-on!

Comparators

Operator	Use
<	Less than
>	Greater than
==	Equal to (this can also be used to compare two text strings)
>=	Equal to or Greater than
<=	Equal to or Less than

Boolean operators

Boolean operation	Outcome
and(true,true)	TRUE
and(true,false)	FALSE
and(false,false)	FALSE
or(true,true)	TRUE
or(true,false)	TRUE
or(false,false)	FALSE
xor(true,true)	FALSE
xor(true,false)	TRUE
xor(false,false)	FALSE

JSON


```
{
  "status": "ok",
  "message-type": "journal",
  "message-version": "1.0.0",
  "message": {
    "last-status-check-time": 1574258137944,
    "counts": {
      "total-dois": 4992,
      "current-dois": 2800,
      "backfile-dois": 2192
    },
    "publisher": "MDPI AG",
    "title": "Entropy",
    "subjects": [
      {
        "name": "General Physics and Astronomy",
        "ASJC": 3100
      }
    ],
    "ISSN": [
      "1099-4300"
    ],
    "issn-type": [
      {
        "value": "1099-4300",
        "type": "electronic"
      }
    ]
  }
}
```

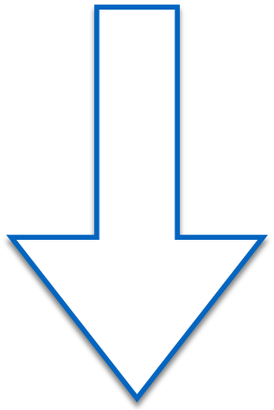
value.parseJson().get("message").get("issn-type")


```
{
  "status": "ok",
  "message-type": "journal",
  "message-version": "1.0.0",
  "message": {
    "last-status-check-time": 1574258137944,
    "counts": {
      "total-dois": 4992,
      "current-dois": 2800,
      "backfile-dois": 2192
    },
    "publisher": "MDPI AG",
    "title": "Entropy",
    "subjects": [
      {
        "name": "General Physics and Astronomy",
        "ASJC": 3100
      }
    ],
    "ISSN": [
      "1099-4300"
    ],
    "issn-type": [
      {
        "value": "1099-4300",
        "type": "electronic"
      }
    ]
  }
}
```


Selects the ‘issn-type’ array in the ‘messages’ object which can contain one or more ISSN value

Filtering arrays

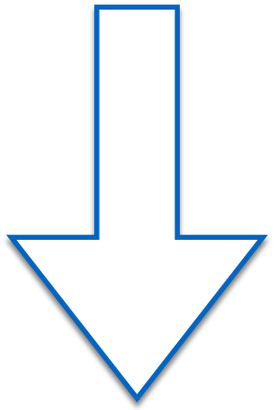
```
filter(["one","two","three"],v,v.startsWith("t"))
```



```
["two","three"]
```

Iterating through arrays with forEach()

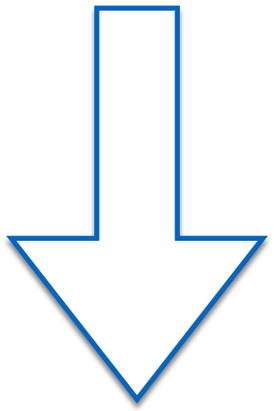
```
forEach(["one","two","three"],v,v.startsWith("t"))
```



```
[false,true,true]
```

Combine filter and forEach

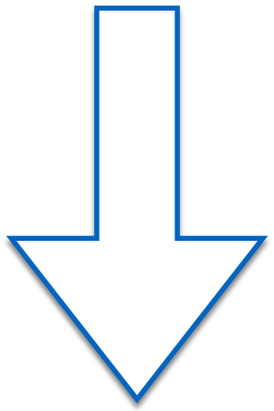
```
([“one|1","two|2","three|3"],v,v.startsWith("t"),w,w.
```



????

Combine filter and forEach

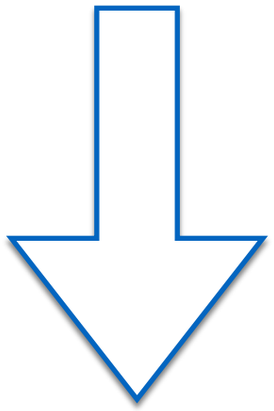
```
filter(["one|1","two|2","three|3"],v,v.startsWith("t"))
```



```
[ "two|2", "three|3" ]
```

Combine filter and forEach

```
forEach(["two|2","three|3"],w,w.split("|")[1])
```



```
[ "2", "3" ]
```

Contributing to OpenRefine

- Join the community at http://groups.google.com/forum/#!forum/openrefine
  - Ask questions, answer questions
- Add to the documentation at https://github.com/openrefine/openrefine/wiki
- Help translate the OpenRefine interface https://hosted.weblate.org/engage/openrefine/
- Report bugs or request enhancements at https://github.com/OpenRefine/OpenRefine/issues/new/choose

Contributing code to OpenRefine

- Documentation for developers https://github.com/OpenRefine/OpenRefine/wiki/Documentation-For-Developers
  - Google Group https://groups.google.com/forum/#!forum/openrefine-dev
- Tackle existing issues (look for the “Good first issues” https://github.com/OpenRefine/OpenRefine/issues?q=is%3Aissue+is%3Aopen+label%3A%22good+first+issue%22)
  - Always feel free to ask for guidance by posting questions on the issue
- Create issues for discussion at https://github.com/OpenRefine/OpenRefine/issues/new/choose
  - Can extend core product
  - Can write an extension to separately extend OpenRefine functionality
- Add a reconciliation service to an existing data source

Reconciliation services

- Reconciliation services consist of one or more APIs to a data source:
  - Reconciliation API (required)
  - Suggest API (optional)
  - Preview API (optional)
  - Data extension API (optional)
- Overview at https://github.com/OpenRefine/OpenRefine/wiki/Reconciliation-Service-API

Implementing a Reconciliation service

- Minimum implementation is simply a URL which can:
  - Return “service metadata” which describes your reconciliation service
  - Support a “query” parameter which contains a batch of queries and provide the results for that query
- API uses JSON for data recieved/returned

Implementing a Reconciliation service

- The Conciliator framework can be used to bolt a reconciliation service on top of an existing API
  - https://github.com/codeforkjeff/conciliator
- Already used to add reconciliation services to:
  - VIAF
  - ORCID
  - OpenLibrary
- The Wikidata reconciliation endpoint is implemented with code that can be used with other wikibase installations https://github.com/wetneb/openrefine-wikibase

Improving and growing the reconciliation API

- A W3C group has been set up to discuss how the API can be improved
  - https://www.w3.org/community/reconciliation/
- A “test bench” has been setup to automatically query existing reconciliation services and assess what services they support
  - https://reconciliation-api.github.io/testbench/

Upcoming OpenRefine developments

- OpenRefine 3.3 beta release available
  - The more testing the better!
- Chan Zuckerberg Initiative (CZI) grant $200,000 to:
  - grow the community of OpenRefine contributors by reaching out to seasoned users and helping them get involved more closely in the project
  - revamp the core architecture of the tool to handle larger datasets and improve workflows
  - Owen Stephens (http://twitter.com/ostephens) and Anton Delpeuch (https://www.cs.ox.ac.uk/people/antonin.delpeuch/) will be working on this in 2020