Tracking alternate labels for subject headings with Wikibase

LD4 Wikibase Working Hour at the 2022 LD4 Conference on Linked Data
July 2022
Diane Shaw, Jackie Shieh, Jim Hahn, Alexander Whelan, Brian Luna Lucero, Esther Jackson*, Melanie Wacker, Timothy Ryan Mendenhall*, Honor Moody*, Christine Fernsebner Eslao

Session Overview

- Differences between Wikidata, Wikibase, and Wikibase Cloud
- Creating items and properties in Wikibase
- Using the Query service in a Wikibase instance
- Using Quick Statements with a Wikibase instance
- OpenRefine into Wikibase workflows
- For simplicity, we may use “Wikibase” to refer to our wikibase.cloud instance during this workshop

Course materials

Please visit our GitHub page if you'd like to follow along with the demonstrations

https://bit.ly/3aohAft

About the LD4 Wikibase Working Hour

The LD4 Wikibase Working Hour seeks to create a space for GLAM professionals experimenting with Wikibase/WBStack implementation, the software that Wikidata is based on, to learn collaboratively and share tips, tools, and resources. The working group will facilitate identification of areas for collaboration among institutions experimenting with separate institutional Wikibase instances. Topics considered are:

- Shared data modeling of properties common to many GLAM collections
- The development of documentation for the GLAM Wikibase community
- Learning about each other's Wikibase projects, workflows, and tools.
- Explore the development of mechanisms for channeling community feedback to the developers of Wikibase and the broader Wikimedia community.
- Complement, (or supplement) where possible, the work from existing Wikidata groups

More details on LD4 Wikibase Working Hour wiki page

About our test instance in wikibase.cloud

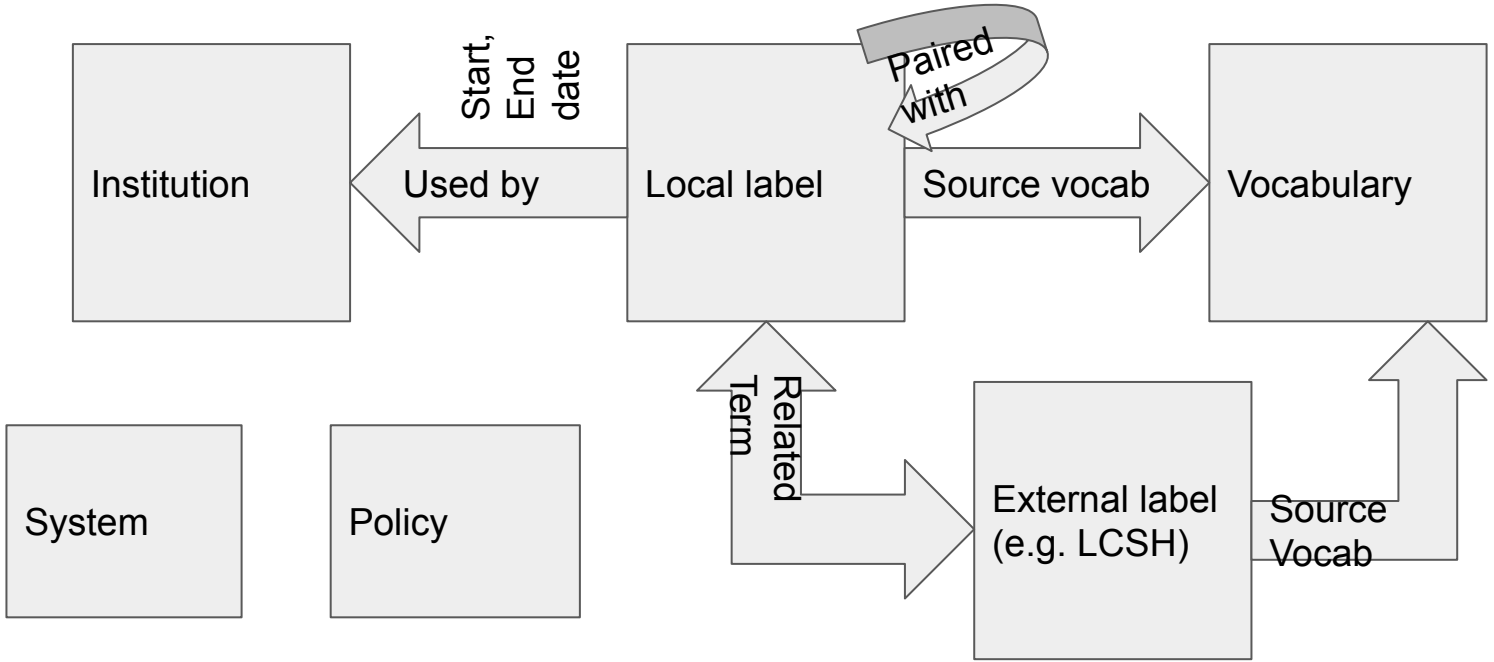
- Originally built in WBStack, the predecessor of wikibase.cloud, in 2021; migrated in June 2022
- Overall goal: create a repository of local labels being used for LCSH headings
- No sustainability plan in place: this is just an exercise or experiment
- Draft data model centered on labels, not concepts. Think of the RDA entity Nomen and its affiliated Appellation properties
- No attempt create new alternative labels, just to document alternative labels currently in use
- We are not trying to replicate or import the entirety of LCSH, just the aspects needed to document the use of alternative labels for LCSH
- Organized around “shapes” or primary entities being described by the model. Some properties repeat across entities, like Label, Alias, Description

Primary entities

- Preferred local label
- Vocabulary
- Institution
- External label
- System
- Policy
- Proposed entities:
  - Source citation
  - Revision proposal
  - Replacement method

Primary entities: Simplified data model

All have default properties: label, alias, description as well as “instance of” property



Outcomes so far

From October-December 2021, the Working Hour hosted three sessions on the WBStack instance

Sessions focused on collaborative data modeling, property creation, and item creation

But we still have a long way to go: most properties have been created, but only a relatively small number of items and statements

In June 2022, our WBStack instance was migrated to wikibase.cloud: https://ld4-wbs-test.wikibase.cloud/

Differences between Wikibase, Wikidata, and Wikibase.cloud

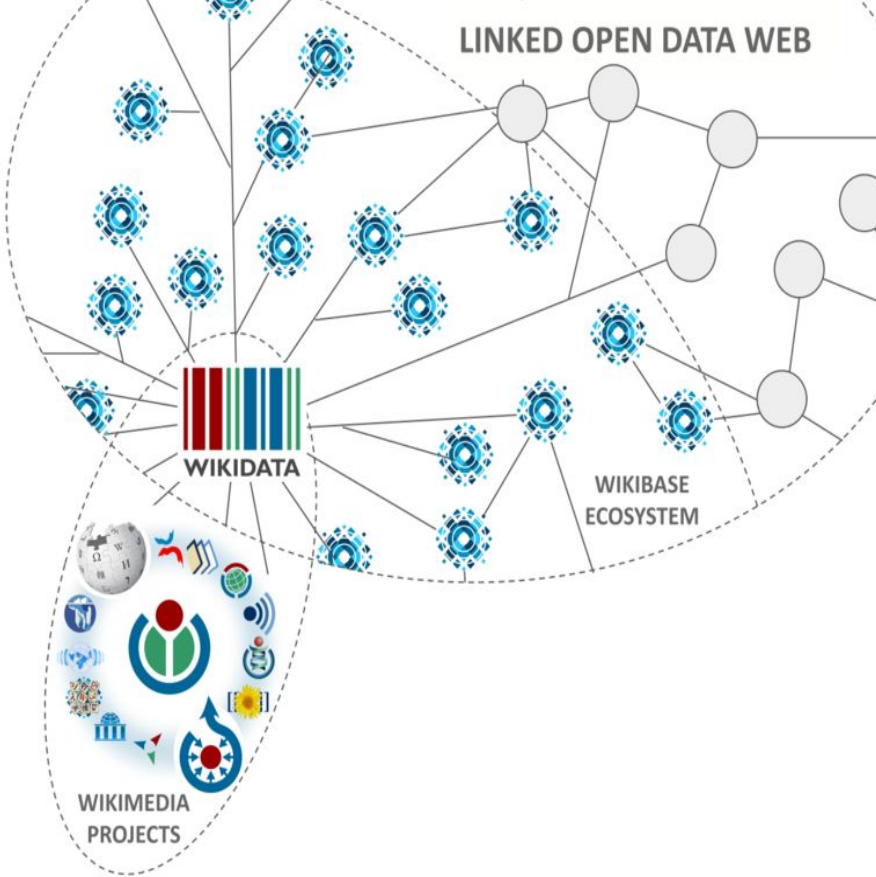
- Wikibase
  - Set of extensions developed by Wikimedia Deutschland for MediaWiki, the open source software designed for use by Wikipedia
    - Core extensions:
      - Wikibase Repository
      - Wikibase Client
    - Data model agnostic tool for creating and storing structured data
  - Software behind Wikidata, and developed for Wikidata
- Wikidata
  - Multilingual collaborative knowledge graph powered by MediaWiki software and the Wikibase extensions, that is, an instance of Wikibase
- Wikibase.cloud
  - Wikibase software as a service supported by Wikimedia Deutschland, that is, installation packages and hosting for instances of Wikbase provided by Wikimedia Deutschland for other individuals and organizations

Wikibase.cloud

- Wikibase instances hosted by Wikimedia Deutschland, using WBStack code developed by Adam Shoreland (Addshore)
  - WBStack funded initially by Addshore, then by Rhizome, and most recently by Wikimedia Deutschland
  - Rhizome continues to be the fiscal sponsor of the Wikibase Stakeholder Group
- Rolled out in 2022 and limited to existing WBStack instances of Wikibase
  - Expanded service expected later in 2022
  - LD4 Wikibase Working Hour test instance migrated in June 2022
- Core component of the Wikimedia Foundation and Wikimedia Deutschland Linked Open Data Strategy (2021) for building out the Wikibase Ecosystem:
  - “Its primary target groups are research groups, private individuals and less well-funded organizations that want to share free and open data with the world but that lack the dedicated resources needed to maintain the software themselves. It is designed for ease of use rather than maximum customization.”--Joint Vision

Wikibase Ecosystem Goals Identified in the Wikimedia [Strategy for the Wikibase Ecosystem (2019)]

- Open up data that is hidden in silos
- Connect data to surface undiscovered connections
- Connect people to enable collaboration
- Strengthen the open knowledge movement as a whole in a new, decentralized, linked data ecosystem
- Enable the creation of new products and services



Wikibase and Wikimedia Linked Open Data Strategy

- Wikimedia Deutschland and the Wikimedia Foundation Linked Open Data Strategy (2021)
  -
- 2021 strategy builds on and updates 4 part Wikidata Strategy of 2019
  - Wikidata/Wikibase Vision
  - Wikidata for Wikimedia projects
  - Wikidata as a platform
  - Wikibase ecosystem
- The WikiLibrary Manifesto
  - “The WikiLibrary Manifesto aims at connecting libraries and Wikimedia projects such as Wikibase in an international network of knowledge. Our goal: The creation and implementation of a single linked open data network for art, culture and science.”

Before creating items or properties In Wikibase

When developing a WB, keep in mind that you're designing a database that should be able to yield useful results from a SPARQL query

- Keep the level of classes (and subclasses if you have them) simple enough for returning useful query results. Avoid creating statements that are too "nested" with qualifiers which generate complex hierarchies of data

- Depending on your needs, decide which properties & items must be created at a minimum (you can always add more later as your database grows)

- Document your usage instructions and preferred practices; you can do that through a separate Wikibase page, or in the discussion tab of an item or property, or in a "usage instructions" property

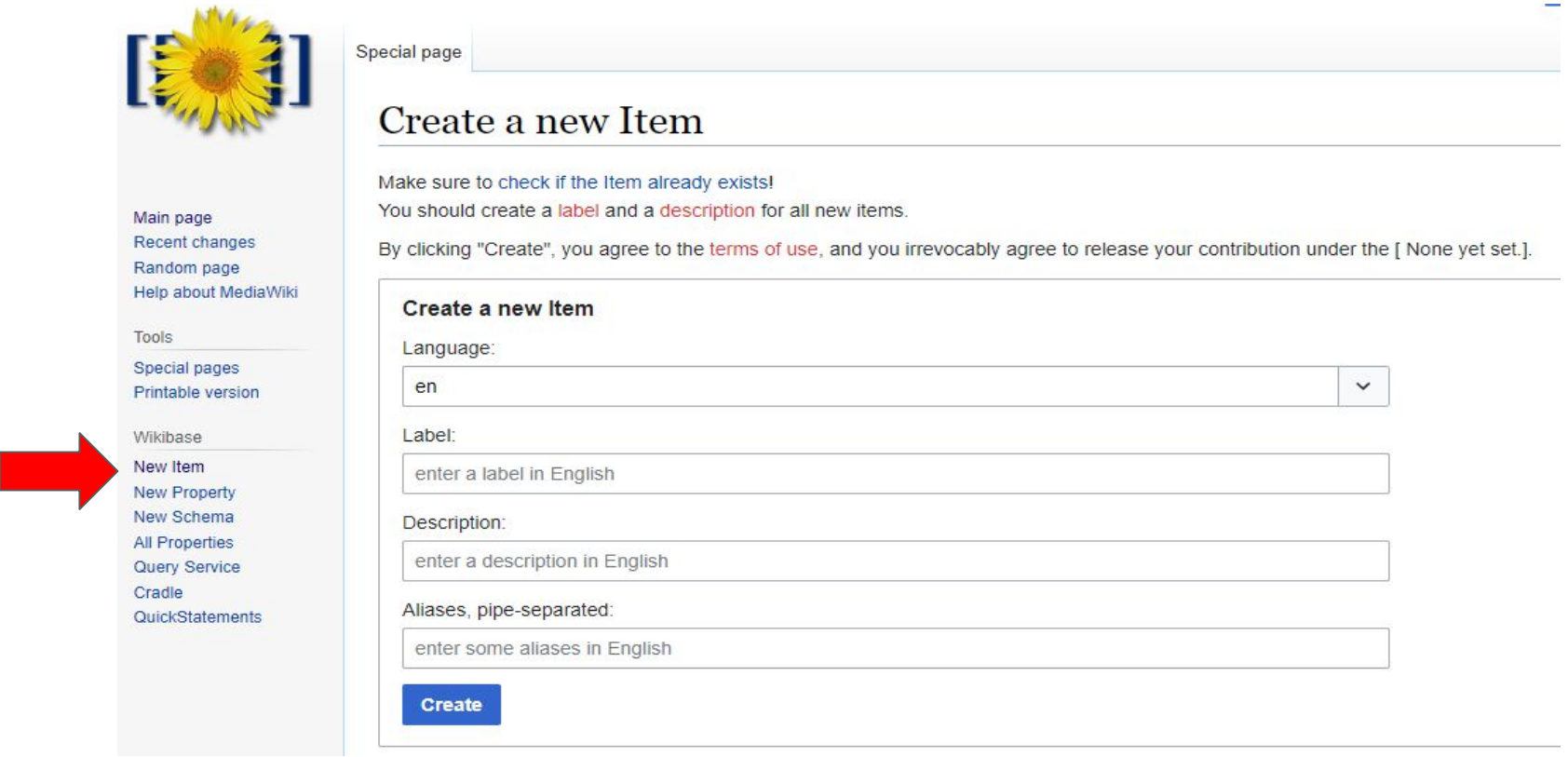
Before creating items or properties in Wikibase

- Make sure the item or property to be created isn't already in the WB. This is especially important when creating properties, because once the data type is set, a duplicate property can only be overwritten with a property that has the same data type

- It can be helpful to maintain a separate, sortable spreadsheet with a list of the items and properties created for your WB, if the search engine isn't always reliable

	A	B	C	D	E	F
1	**LCSH Label**	**LCHS-label_**	**Preferred Local Label**	**Preferred-Local**	**Institution**	**InstitutionID**
2	Officials and employees, Alien		Officials and employees, Noncitizen		Columbia Univer	Q22
3	African American universities and co	Q21	Historically Black colleges and uni	Q23	Cataloging Lab	Q24
4	Albinos and albinism		Albinism OR People with albinism		Cataloging Lab	Q24
5	Aleuts--Evacuation and relocation, 1942-1945		Forced removal and incarceration of Unangax̂, 194		Cataloging Lab	Q24
6	Alien criminals	Q9	Noncitizen criminals	Q17	Harvard Univers	Q29
7	Alien criminals	Q9	Noncitizen criminals	Q17	Columbia Univer	Q22

How to create items and properties in Wikibase



Special page

Create a new Property

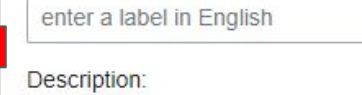
Make sure to check if the Property already exists!
You should create a label and a description for all new Properties, and in addition a valid Property type.

By clicking "Create", you agree to the terms of use, and you irrevocably agree to release your contribution under the [None yet set.].

Main page
Recent changes
Random page
Help about MediaWiki

Tools
Special pages
Printable version

Wikibase
New Item
New Property
New Schema
All Properties
Query Service
Cradle
QuickStatements



Create a new Property

Language:

en

Label:

enter a label in English

Description:

enter a description in English

Aliases, pipe-separated:

enter some aliases in English

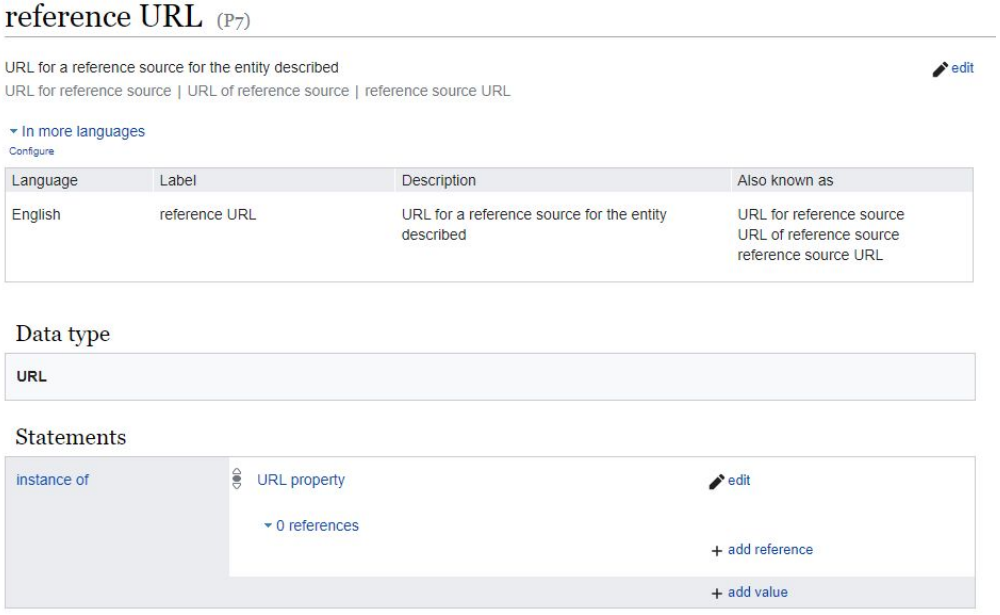
Data type:

(pick a data type)

Create

Creating properties

- For properties, at a minimum there needs to be language of the term, label, description, aliases (if any), and a data type (for example: external identifier, item, monolingual text, point in time, property, quantity, string, URL)



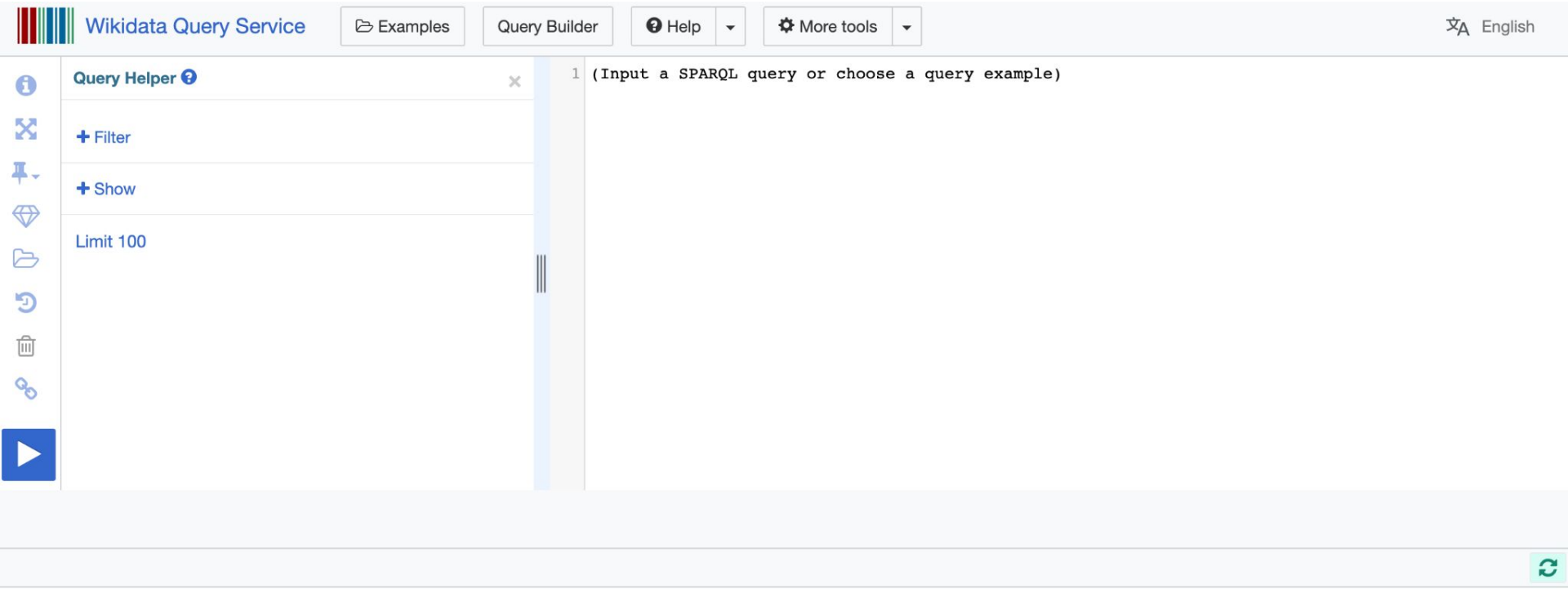
Creating properties

- Properties can also be used as qualifiers
- Properties can be items and items can be properties (for example, “occupation”)
- Reciprocal properties (for example, “has part or parts” and “part of”)
- How to set preference in the case of multiple statements using the same property in an item, e.g. to distinguish between current and obsolete/wrong external identifiers
- Some recommended basic properties to use for items:
    - Every kind of item: instance of; external identifiers like VIAF or Wikidata Q number
    - Humans: birth date; death date; occupation; employer; educated at
    - Organizations: inception date; country; street address; official website

When to use different query services

- Use the **Wikidata Query Service** to query Wikidata proper
- Use the **Docker Wikibase Query Service** for a local installation of Wikibase
- Use the **Wikibase.cloud Query Service** to query a base that is hosted on Wikibase.cloud

Different Services: Wikidata Query Service (User Manual)



Understanding Prefixes

- The subjects and predicates (first and second values of the triple) must always be stored as URI.
- For example, if the subject is Universe (Q1), it will be stored as **<https://www.wikidata.org/wiki/Q1>**.
- Prefixes allow us to write that long URI in a shorter form: **wd:Q1**. Unlike subjects and predicates, the object (triple's third value) can be either a URI or a literal, e.g. a number or a string.
- WDQS understands many shortcut abbreviations, known as prefixes. Some are internal to Wikidata, e.g. **wd, wdt, p, ps, bd**, and many others are commonly used external prefixes, like **rdf, skos, owl, schema**.

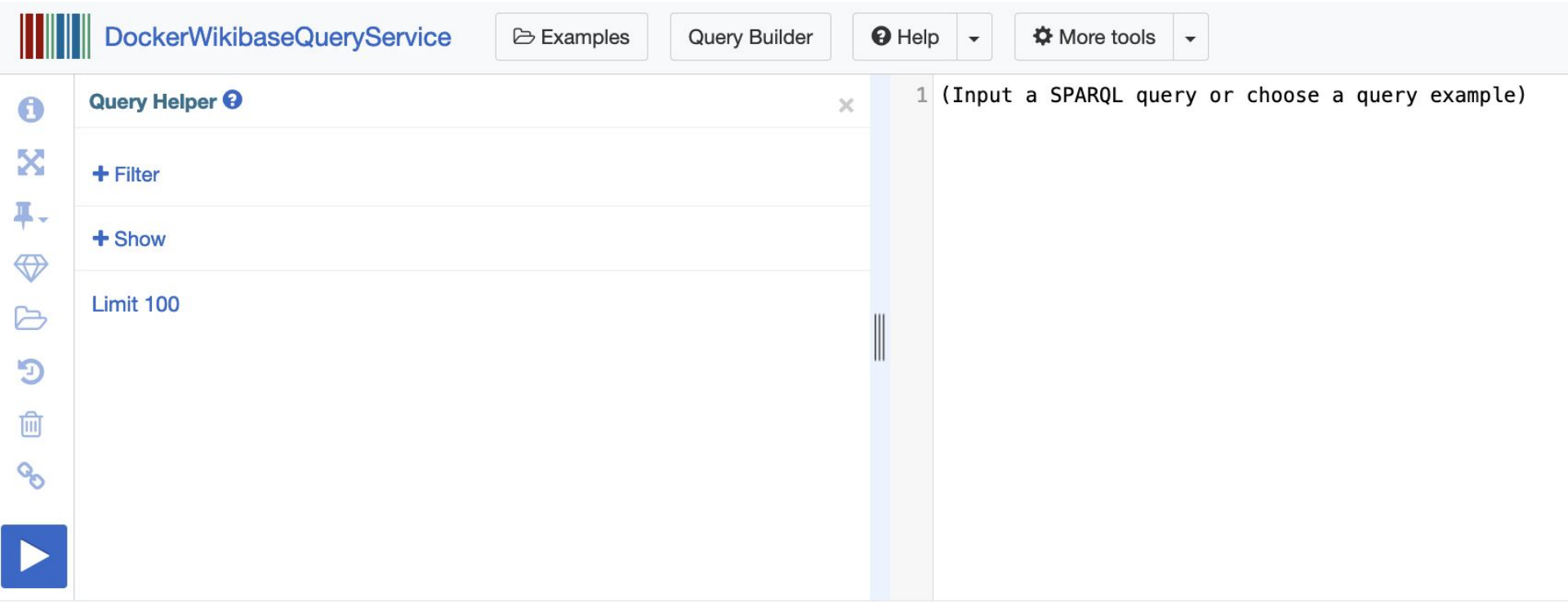
From: https://www.mediawiki.org/wiki/Wikidata_Query_Service/User_Manual/el#Basics_-_Understanding_Prefixes

Prefix examples

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX p: <http://www.wikidata.org/prop/>
PREFIX ps: <http://www.wikidata.org/prop/statement/>
PREFIX pq: <http://www.wikidata.org/prop/qualifier/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX bd: <http://www.bigdata.com/rdf#>

```

Available prefixes & documentation:
https://www.mediawiki.org/wiki/Wikibase/Indexing/RDF_Dump_Format/es#Prefixes_used

Different Services: Docker Wikibase Query Service



Different Services: Docker Wikibase Query Service

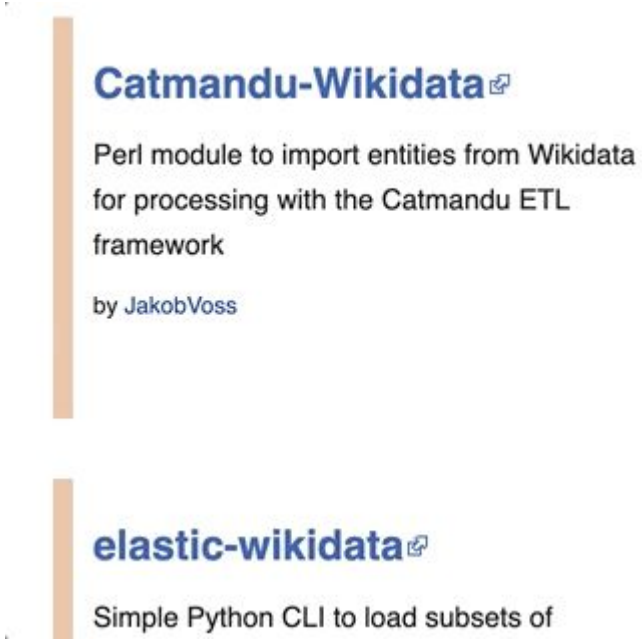
docker-compose -f docker-compose.yml -f docker-compose.extra.yml up

A standalone Wikibase instance can be [configured](#) to support its own prefixes (see Rhizome's [ArtBase Query service](#) for an example)

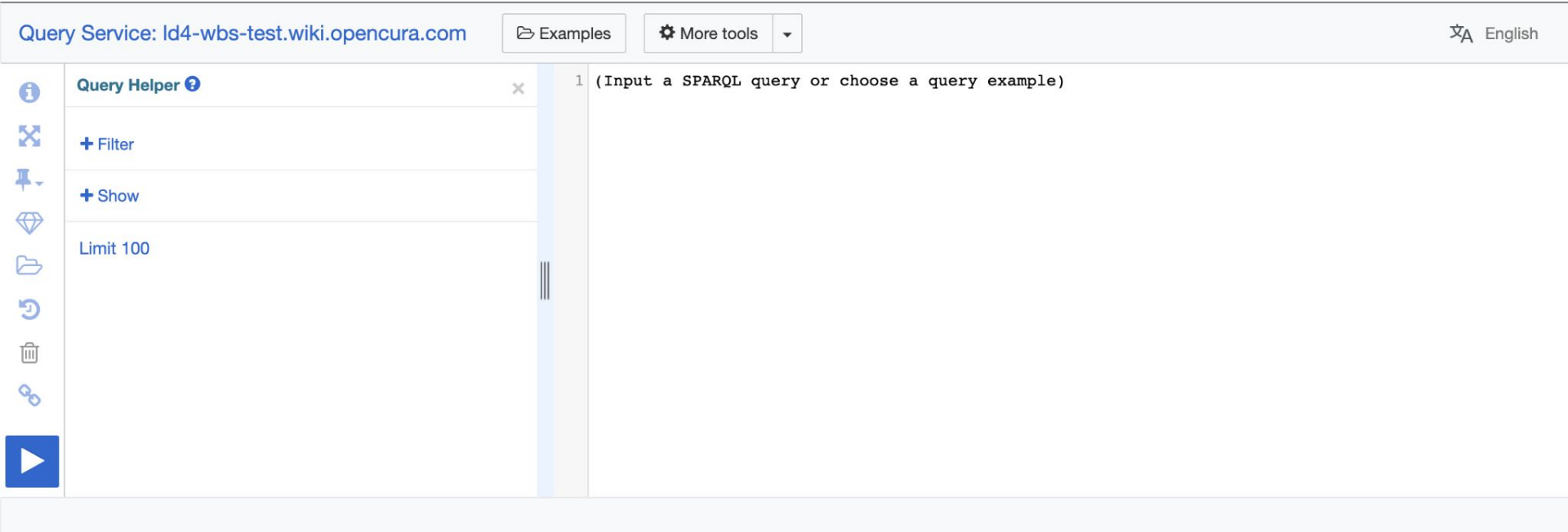
```
version: '3'

services:
  wdqs-frontend:
    image: "${WDQS_FRONTEND_IMAGE_NAME}"
    restart: unless-stopped
    ports:
     - "${WDQS_FRONTEND_PORT}:80"
    depends_on:
     - wdqs-proxy
    networks:
      default:
        aliases:
         - localhost
    environment:
      - WIKIBASE_HOST=${WIKIBASE_HOST}
      - WDQS_HOST=host.docker.internal
  wdqs:
    image: "${WDQS_IMAGE_NAME}"
    restart: unless-stopped
    command: /runBlazegraph.sh
    volumes:
      - query-service-data:/wdqs/data
    networks:
      default:
        aliases:
         - host.docker.internal
    environment:
      - WIKIBASE_HOST=${WIKIBASE_HOST}
      - WDQS_HOST=host.docker.internal
      - WDQS_PORT=9999
    expose:
      - 9999
  wdqs-proxy:
    image: "${WDQS_PROXY_IMAGE_NAME}"
    restart: unless-stopped
    environment:
      - PROXY_PASS_HOST=host.docker.internal:9999
    depends_on:
      - wdqs
    networks:
      default:
        aliases:
         - host.docker.internal
  wdqs-updater:
    image: "${WDQS_IMAGE_NAME}"
    restart: unless-stopped
    command: /runUpdate.sh
    depends_on:
    - wdqs
    - wikibase
    networks:
      default:
        aliases:
         - host.docker.internal
    environment:
     - WIKIBASE_HOST=${WIKIBASE_HOST}
     - WDQS_HOST=host.docker.internal
```

Wikidata:Tools/For programmers

Because Wikidata runs on the Wikibase software, many of the tools that have been developed to work with Wikidata have been decoupled from Wikidata to work with local Wikibase installations as well. Some examples can be found here: https://www.wikidata.org/wiki/Wikidata:Tools/For_programmers



Different Services: Wikibase.cloud Query Service



Invisible differences and missing features

- **Wikibase Cloud query service, which we're using in this workshop, requires prefixes to be specified.**
  - This is not the case in Wikidata, where most prefixes are automatically supported by the query engine

- **The query helper works… almost**
  - "Filter" uses your Wikibase's Q IDs but will insert property IDs from Wikidata that you will need to edit (P31 for instance of)
  - "Show" will correctly construct optional parameters using local property IDs but remove prefixes that you've added

- **Examples are not supplied by the UI**

URL for our demo Wikibase's query service:
https://ld4-wbs-test.wikibase.cloud/query

Some properties in our example Wikibase cloud instance

Wikidata property	Wikibase property	Example Wikibase URL string + property
P31 *instance of*	P1 *instance of*	<https://ld4-wbs-test.wikibase.cloud/prop/direct/P1>
n/a	P2 *source vocabulary*	<https://ld4-wbs-test.wikibase.cloud/prop/direct/P2>
n/a	P3 *used by*	<https://ld4-wbs-test.wikibase.cloud/prop/direct/P3>
n/a	P4 *related term*	<https://ld4-wbs-test.wikibase.cloud/prop/direct/P4>

Wikidata item	Wikibase item	Example Wikibase URL string + item
Q1823134	Q2 *Library of Congress Subject Headings*	<https://ld4-wbs-test.wikibase.cloud/entity/Q2>

Full property list: https://ld4-wbs-test.wikibase.cloud/wiki/Special:ListProperties

Using property and item URLs without prefixes

```
SELECT ?lcsh ?lcshLabel ?institution ?institutionLabel WHERE {

  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }

  # the item has a source vocabulary of ‘Library of Congress’
  ?lcsh <https://ld4-wbs-test.wikibase.cloud/prop/direct/P2>
  <https://ld4-wbs-test.wikibase.cloud/entity/Q2>.

  # the item has an alternative, related term
  ?lcsh <https://ld4-wbs-test.wikibase.cloud/prop/direct/P4> ?relatedterm.

  # the related term is in use by an institution
  ?relatedterm <https://ld4-wbs-test.wikibase.cloud/prop/direct/P3> ?institution.
}
LIMIT 100
```

Query results: https://tinyurl.com/2fwleftx

lcsh	lcshLabel	institution	institutionLabel
<https://ld4-wbs-test.wikibase.cloud/entity/Q10>	Alien detention centers	<https://ld4-wbs-test.wikibase.cloud/entity/Q22>	Columbia University Libraries
<https://ld4-wbs-test.wikibase.cloud/entity/Q10>	Alien detention centers	<https://ld4-wbs-test.wikibase.cloud/entity/Q29>	Harvard University Libraries
<https://ld4-wbs-test.wikibase.cloud/entity/Q15>	American poetry--Indian authors	<https://ld4-wbs-test.wikibase.cloud/entity/Q24>	Cataloging Lab
<https://ld4-wbs-test.wikibase.cloud/entity/Q21>	African American universities and colleges	<https://ld4-wbs-test.wikibase.cloud/entity/Q24>	Cataloging Lab
<https://ld4-wbs-test.wikibase.cloud/entity/Q33>	Child pornography	<https://ld4-wbs-test.wikibase.cloud/entity/Q24>	Cataloging Lab

Prefixes structure

Formula

```
PREFIX arbitrary prefix for properties: <your wikibase URL/prop/direct/>
PREFIX arbitrary prefix for entities: <your wikibase URL/entity/>
```

Example Wikibase.cloud instance

```
PREFIX wdt: <http://ld4-wbs-test.wiki.opencura.com/prop/direct/>
PREFIX wd: <http://ld4-wbs-test.wiki.opencura.com/entity/>
```

How to use Wikibase.cloud query service with prefixes

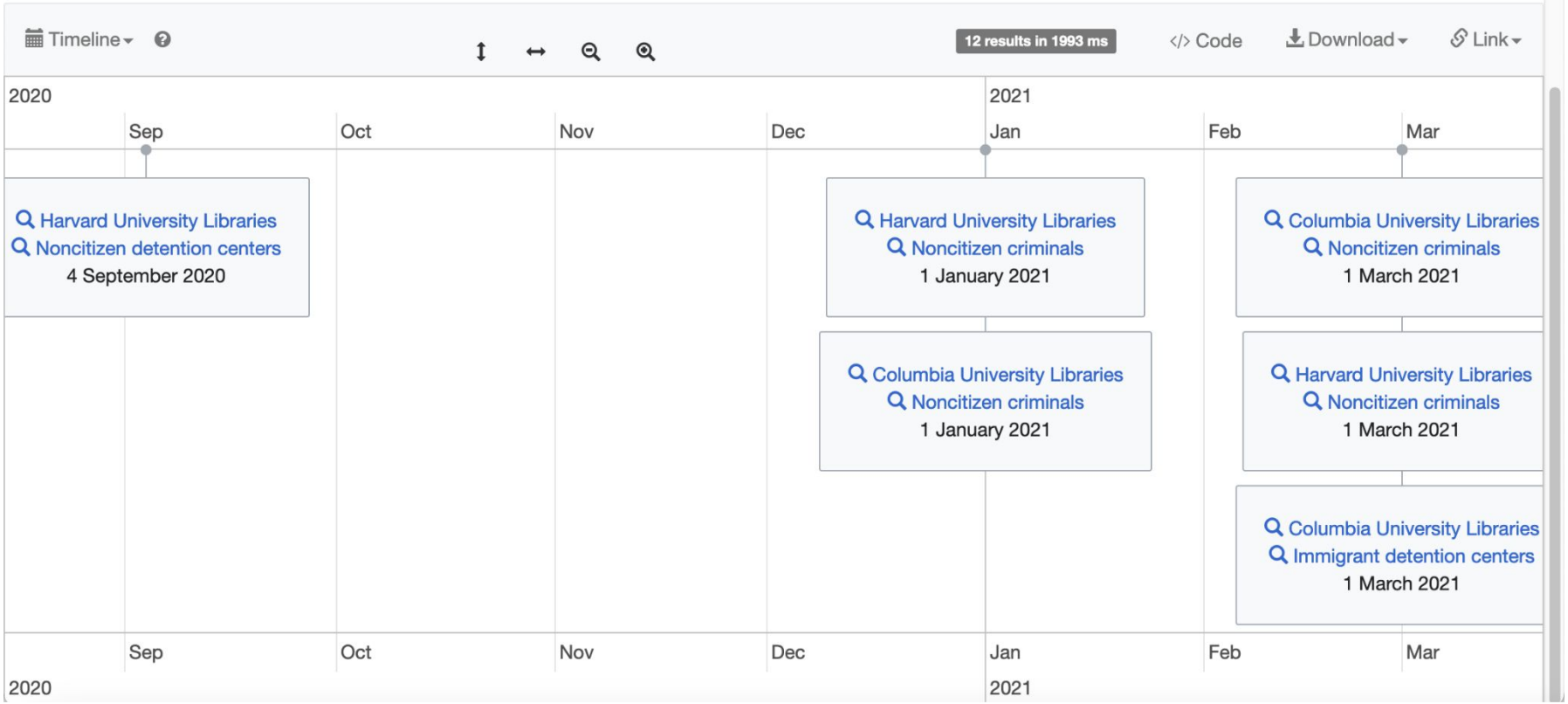
```
PREFIX wdt: <https://ld4-wbs-test.wikibase.cloud/prop/direct/>
PREFIX wd: <https://ld4-wbs-test.wikibase.cloud/entity/>

SELECT ?lcsh ?lcshLabel ?institution ?institutionLabel WHERE {

  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }

  ?lcsh wdt:P2 wd:Q2.                 # the item's source vocabulary is LCSH
  ?lcsh wdt:P4 ?relatedterm.          # the item has an alternative, related term
  ?relatedterm wdt:P3 ?institution.   # the related term is in use by an institution

}

LIMIT 100
```

(We're using the prefixes that we're familiar with from Wikidata's query service, because they're easy to read. But they don't point to Wikidata's items or properties here!)

Adding a qualifier to a Wikibase.cloud query and displaying a visualization

```
#defaultView:Timeline
PREFIX wdt: <https://ld4-wbs-test.wikibase.cloud/prop/direct/>
PREFIX wd: <https://ld4-wbs-test.wikibase.cloud/entity/>
PREFIX p: <https://ld4-wbs-test.wikibase.cloud/prop/>
PREFIX pq: <https://ld4-wbs-test.wikibase.cloud/prop/qualifier/>

SELECT ?term ?termLabel ?date ?institution ?institutionLabel WHERE {
     SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }
     ?term wdt:P3 ?institution.
     ?term p:P3 ?statement.          # p: object of this property is a statement
     ?statement pq:P5 ?date.         # pq: retrieves value of qualifier on that statement

}

LIMIT 100
```

Wikibase Cloud query timeline



Hands-on activity (if there's time): modifying a query, pt. 1

Open this query as a starting point: https://tinyurl.com/2dhkjftd
The link will open to a list of results; hover your mouse over the menu on the right-hand side of the page and click on 

Add a "used by" column, starting by insert the following line into the query:

```
OPTIONAL { ?preferredlocallabel wdt:P3 ?used_by. }
```

Then insert the following into the `SELECT` statement: `?used_by`

After confirming that this returns results, add a label for the "used by" items by inserting `?used_byLabel` into the `SELECT` statement.

Query result: https://tinyurl.com/2m5qpmn3

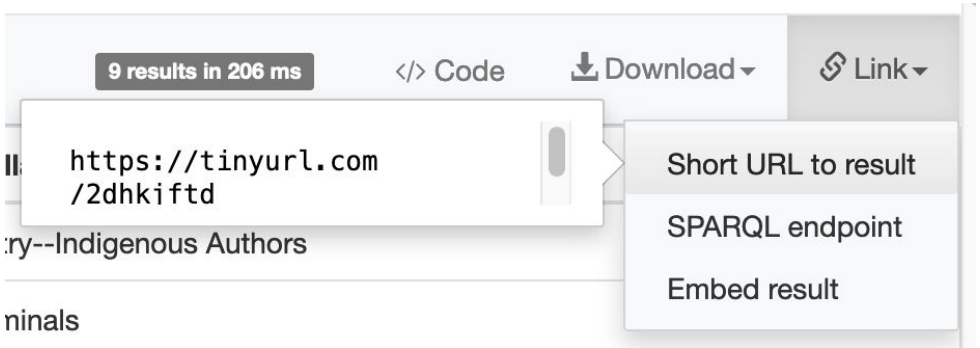
lcsh	lcshLabel	institution	institutionLabel	used_by	used_byLabel
<https://ld4-wbs-test.wikibase.cloud/entity/Q10>	Alien detention centers	<https://ld4-wbs-test.wikibase.cloud/entity/Q22>	Columbia University Libraries	<https://ld4-wbs-test.wikibase.cloud/entity/Q22>	Columbia University Libraries
<https://ld4-wbs-test.wikibase.cloud/entity/Q10>	Alien detention centers	<https://ld4-wbs-test.wikibase.cloud/entity/Q29>	Harvard University Libraries	<https://ld4-wbs-test.wikibase.cloud/entity/Q22>	Columbia University Libraries
<https://ld4-wbs-test.wikibase.cloud/entity/Q15>	American poetry--Indian authors	<https://ld4-wbs-test.wikibase.cloud/entity/Q24>	Cataloging Lab	<https://ld4-wbs-test.wikibase.cloud/entity/Q22>	Columbia University Libraries
<https://ld4-wbs-test.wikibase.cloud/entity/Q21>	African American universities and colleges	<https://ld4-wbs-test.wikibase.cloud/entity/Q24>	Cataloging Lab	<https://ld4-wbs-test.wikibase.cloud/entity/Q22>	Columbia University Libraries
<https://ld4-wbs-test.wikibase.cloud/entity/Q33>	Child pornography	<https://ld4-wbs-test.wikibase.cloud/entity/Q24>	Cataloging Lab	<https://ld4-wbs-test.wikibase.cloud/entity/Q22>	Columbia University Libraries
<https://ld4-wbs-test.wikibase.cloud/entity/Q10>	Alien detention centers	<https://ld4-wbs-test.wikibase.cloud/entity/Q22>	Columbia University Libraries	<https://ld4-wbs-test.wikibase.cloud/entity/Q22>	Columbia University Libraries
<https://ld4-wbs-test.wikibase.cloud/entity/Q10>	Alien detention centers	<https://ld4-wbs-test.wikibase.cloud/entity/Q29>	Harvard University Libraries	<https://ld4-wbs-test.wikibase.cloud/entity/Q22>	Columbia University Libraries
<https://ld4-wbs-test.wikibase.cloud/entity/Q15>	American poetry--Indian authors	<https://ld4-wbs-test.wikibase.cloud/entity/Q24>	Cataloging Lab	<https://ld4-wbs-test.wikibase.cloud/entity/Q22>	Columbia University Libraries
<https://ld4-wbs-test.wikibase.cloud/entity/Q21>	African American universities and colleges	<https://ld4-wbs-test.wikibase.cloud/entity/Q24>	Cataloging Lab	<https://ld4-wbs-test.wikibase.cloud/entity/Q22>	Columbia University Libraries
<https://ld4-wbs-test.wikibase.cloud/entity/Q33>	Child pornography	<https://ld4-wbs-test.wikibase.cloud/entity/Q24>	Cataloging Lab	<https://ld4-wbs-test.wikibase.cloud/entity/Q22>	Columbia University Libraries

Hands-on activity (if there's time): modifying a query, pt. 2

Now, choose another property and add it to your query.

And/or choose a visualization to for your query results.

Share what you've made by copying the short URL from the "Link" menu at the top right of the results and pasting it into the chat!



How to use QuickStatements with a Wikibase instance

What is QuickStatements?

“The tool can add and remove statements, labels, descriptions and aliases; as well as add statements with optional qualifiers and sources” in Wikidata

QuickStatements (QS) can also be used with Wikibase instances, and any wikibase.cloud instance has built in support for QS

When using QS, it is common to generate the dataset via a spreadsheet editor or OpenRefine, but there are also other ways to use QuickStatements

QuickStatements: Syntax (CSV)

First row needs to contain headers, as described below. The values in the column then serve as predicates for the statements

First column must be labeled “qid” – enter the Q number of the item being edited, or if adding a new item, leave this blank

L + 2-letter language code, e.g. Len, sets a label for the item

D + 2-letter language code, e.g. Den, sets a description for the item

A + 2-letter language code, e.g. Aen, sets an alias / alternate label for the item

P columns: make statements by labeling a column with a P-number

qid	Len	P1	P2
Q36	Officials and em	Q14	Q2
Q37	Albinos and albi	Q14	Q2
Q38	Aleuts--Evacuati	Q14	Q2
	Alien property (G	Q14	Q2
	Aliens (Greek la	Q14	Q2

QuickStatements: Syntax (CSV)

Column headers, continued:

S + property number (minus P): adds a source to the statement

q + property number (minus P): adds a qualifier to the statement

Removing a statement: preface the P-number in the column header with a negative sign, e.g. -P1 to remove a statement from your Wikibase instance

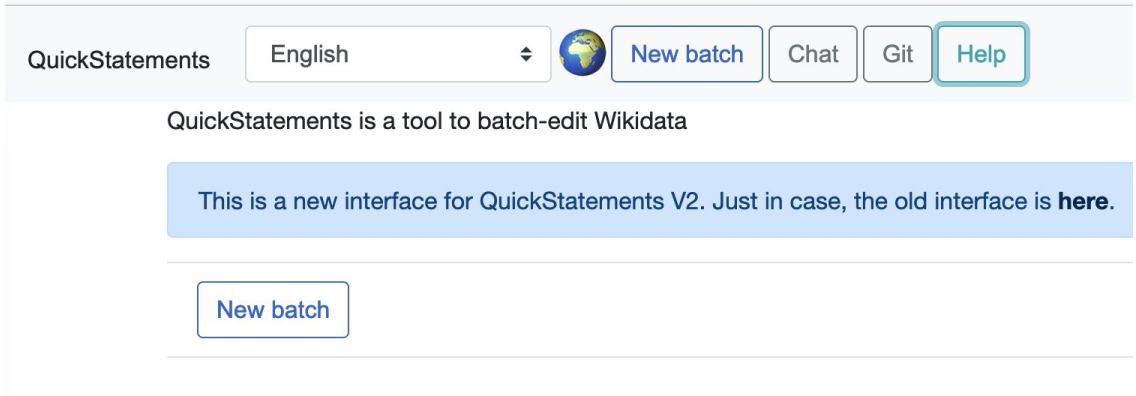
Example and demo: Adding LCSH labels to our instance

Dataset being used: https://bit.ly/3amJfxh

QuickStatements is accessible via the left sidebar in a wikibase.cloud instance

You may need to log in once you're at the QuickStatements page

Click "New Batch"

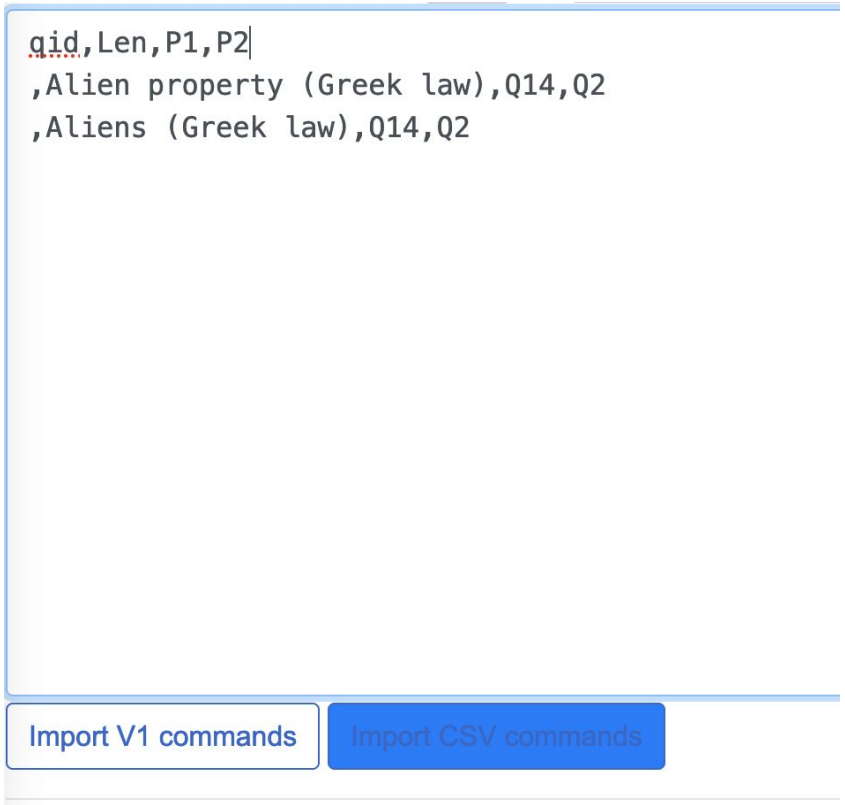


Example and demo: Adding LCSH labels to our instance

Name your batch (optional – assists in retrieval, rollbacks, etc.)

Paste in your QS commands or properly formatted .csv

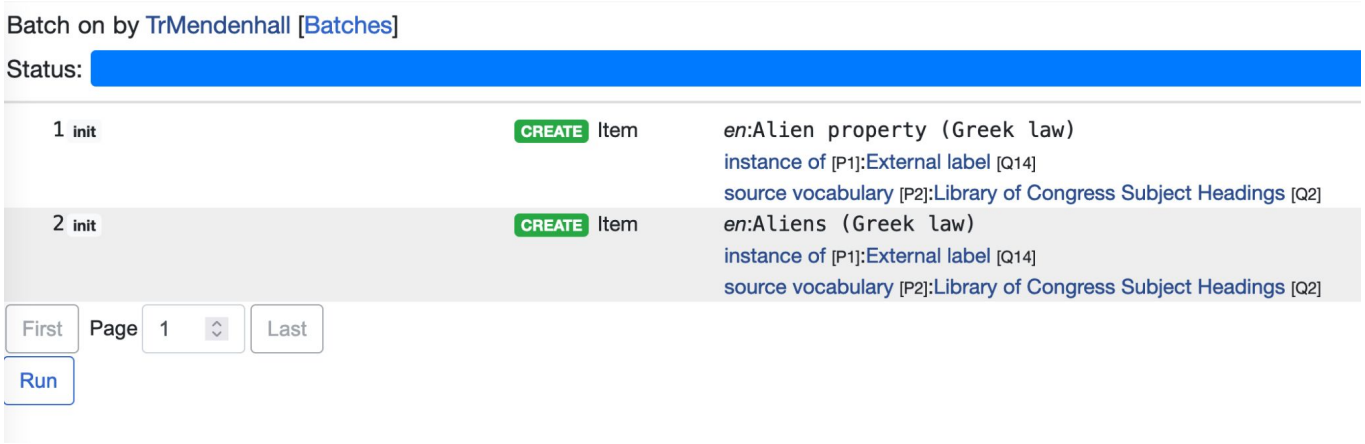
Click “Import commands” (.csv in our case)



Example and demo: Adding LCSH labels to our instance

On the new screen, check the preview screen for accuracy

If you see no issues with quality, click “run batch”



Other ways of using QuickStatements

This demo has shown how to use QuickStatements via a .csv file

For more complex updates, you may need to explore using the “Import commands” option rather than the .csv option

How to integrate OpenRefine into Wikibase workflows

Resources

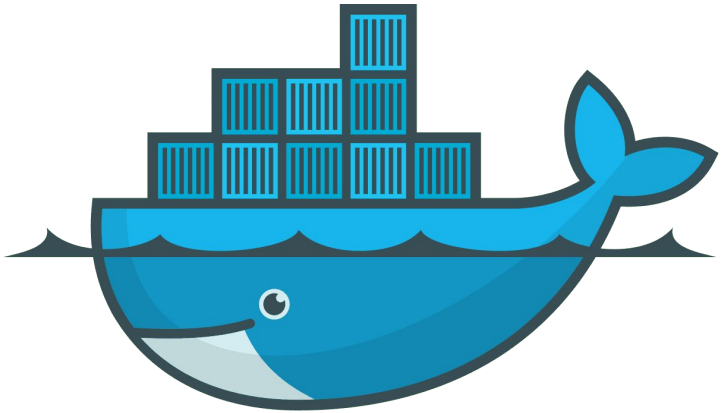
- OpenRefine User Manual: section on reconciling with Wikibase
- Wikibase reconciliation interface for OpenRefine
- Docker Desktop
- Jim Hahn's presentation on this topic
- OpenRefine to Wikibase: Data Upload Pipeline
- TIB presentation on this topic
- Sample Wikibase manifests for use with the OpenRefine [Wiki]data extension

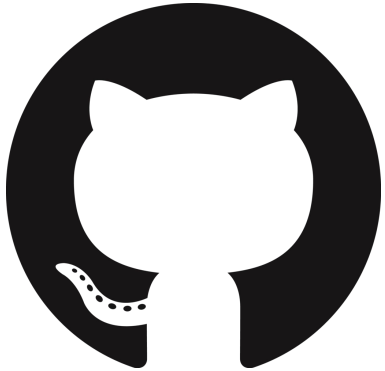
OpenRefine

Steps

Download and install Docker Desktop (if you haven't already)

Download the Wikibase reconciliation interface for OpenRefine, either manually or via a git clone command (using the Terminal / Command Prompt). Download the interface to a directory that you can easily access





Steps

Copy the config-docker.py file and rename it config.py

Update the config.py file to reflect the parameters of your Wikibase / WBStack / Wikibase cloud instance. See this example for details. Note that you don't need to update every parameter to get the reconciliation service to function correctly.

```
# Endpoint of the MediaWiki API of the Wikibase instance
mediawiki_api_endpoint = 'https://ld4-wbs-test.wikibase.cloud/w/api.php'

# SPARQL endpoint
wikibase_sparql_endpoint = 'https://ld4-wbs-test.wikibase.cloud/query/sparql'

# Name of the Wikibase instance
wikibase_name = 'LD4 WBStack Test'

# URL of the main page of the Wikibase instance
wikibase_main_page = 'https://ld4-wbs-test.wikibase.cloud/wiki/Main_Page'

# Wikibase namespace ID, used to search for items
# For Wikidata this is 0, but most by default Wikibase uses 120, which is the
```

Example of a customized config.py file

Steps

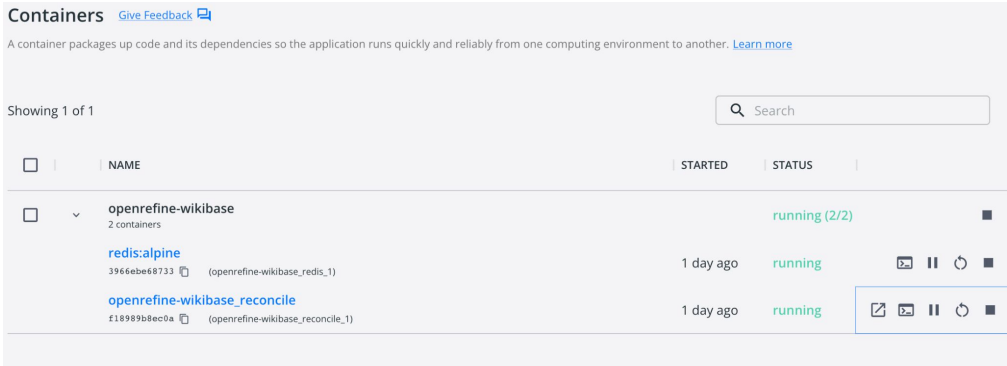
Make sure that Docker is running

Open your terminal (Mac) or command prompt (Windows) and navigate to the directory to which you saved/cloned the Wikibase reconciliation interface: cd ~/Your/directory/path/here/openrefine-wikibase

In the terminal, run docker-compose

- docker-compose up
- sudo docker-compose up
- The latter variant may be necessary on a Mac, depending on your setup

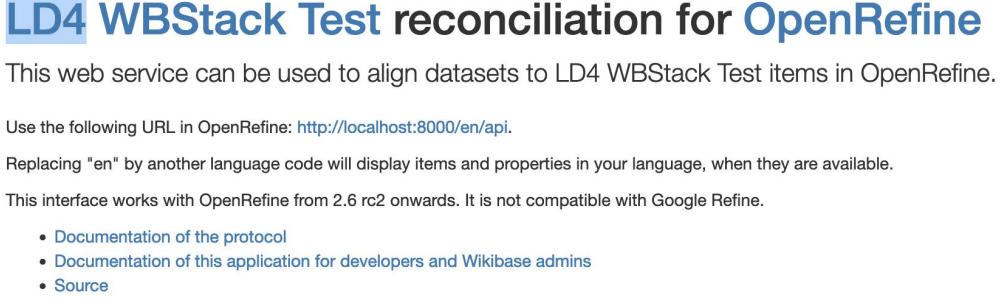
```
timothyryanmendenhall@Timothys-MacBook-Pro openrefine-wikibase % sudo docker-compose
Password:
Creating network "openrefine-wikibase_default" with the default driver
Creating openrefine-wikibase_redis_1     ... done
Creating openrefine-wikibase_reconcile_1 ... done
Attaching to openrefine-wikibase_reconcile_1, openrefine-wikibase_redis_1
redis_1       | 1:C 05 Jul 2022 18:03:44.590 # oO0OoO0OoO0Oo Redis is starting oO0OoO0
redis_1       | 1:C 05 Jul 2022 18:03:44.591 # Redis version=6.2.6, bits=64, commit=00
redis_1       | 1:C 05 Jul 2022 18:03:44.591 # Warning: no config file specified, usin
a config file use redis-server /path/to/redis.conf
redis_1       | 1:M 05 Jul 2022 18:03:44.592 * monotonic clock: POSIX clock_gettime
redis_1       | 1:M 05 Jul 2022 18:03:44.599 * Running mode=standalone, port=6379.
redis_1       | 1:M 05 Jul 2022 18:03:44.599 # Server initialized
redis_1       | 1:M 05 Jul 2022 18:03:44.600 * Ready to accept connections
reconcile_1   |  * Serving Quart app 'app'
reconcile_1   |  * Environment: production
reconcile_1   |  * Please use an ASGI server (e.g. Hypercorn) directly in production
reconcile_1   |  * Debug mode: True
reconcile_1   |  * Running on http://0.0.0.0:8000 (CTRL + C to quit)
```



Steps

The reconciliation service should be running. In your Terminal window, or in Docker desktop, you should see a confirmation that the service is running and the URL of the service, probably http://localhost:8000 [this URL can be adjusted in the config.py file]. You can also paste that address into a web browser to verify the service is running

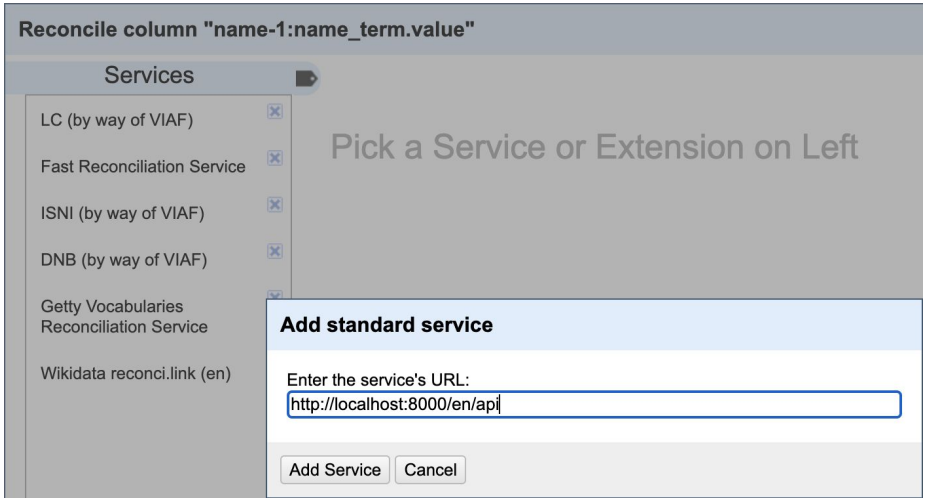
After completing this setup, you can just start the reconciliation service via Docker Desktop–no need to open the Terminal!



Web browser view of the reconciliation service

In OpenRefine

- Start OpenRefine
- Load in this dataset
- Click on the column “Institution” and select Reconcile→Start reconciling
- Click “add standard service”
- Enter in the following reconciliation service URL: http://localhost:8000/en/api



In OpenRefine

Proceed with reconciliation as usual

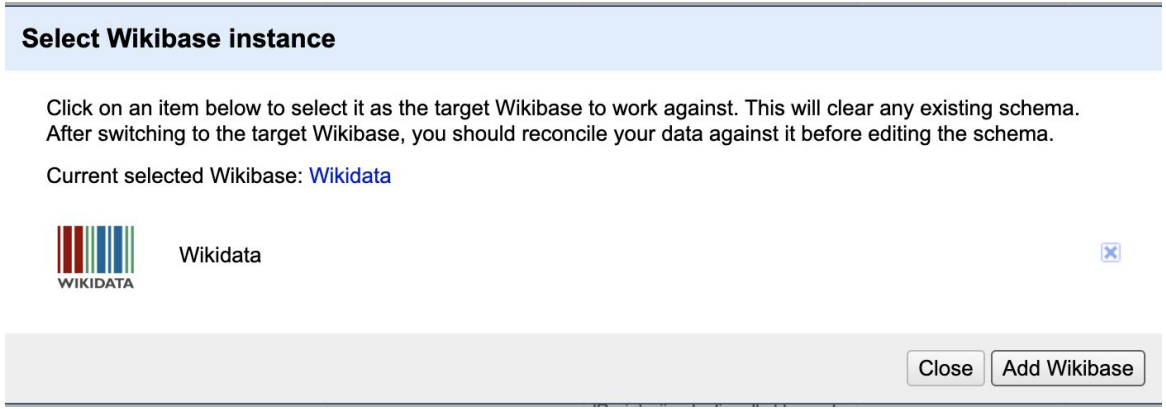
cell.recon.match.id: extracts Q-number of the matched Wikibase item

cell.recon.match.name: extracts the label of the matched Wikibase item

Beyond reconciliation: bulk edits and additions

Unless your reconciliation service is hosted on a web server (rather than your computer), this process does not entirely work at the moment. I will lay out the steps involved, though.

- In the upper right of an OpenRefine project, click Wikidata→Select Wikibase instance
- Click Add Wikibase

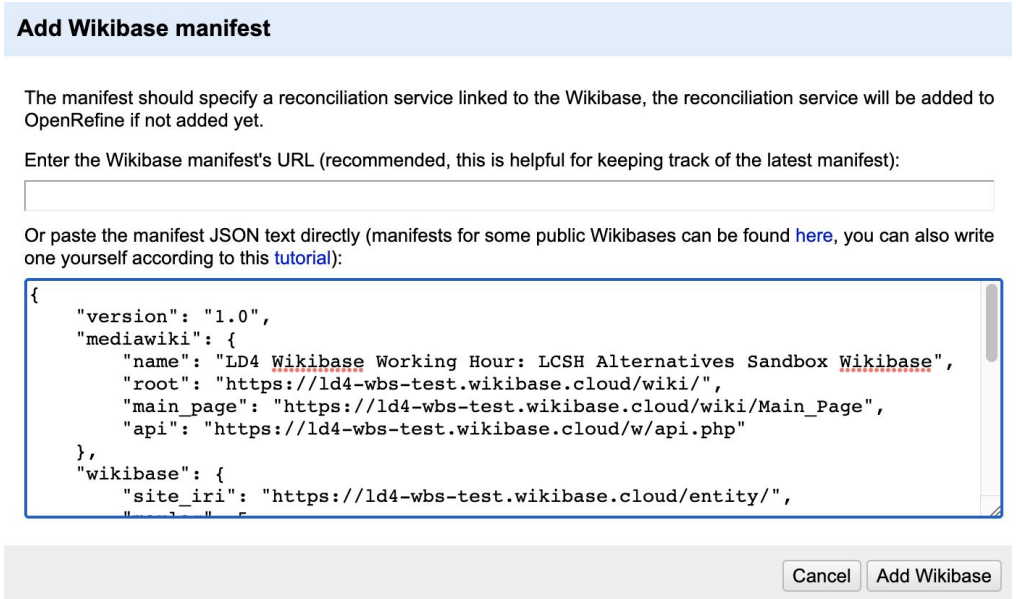


Beyond reconciliation: bulk edits and additions

In the dialogue that appears, paste in a manifest. Samples are available on GitHub that can be customized for your instance

See also this sample

Note that you must follow the version 1 format of the manifests, unless you are using a development version of OpenRefine (currently v. 3.6.x)



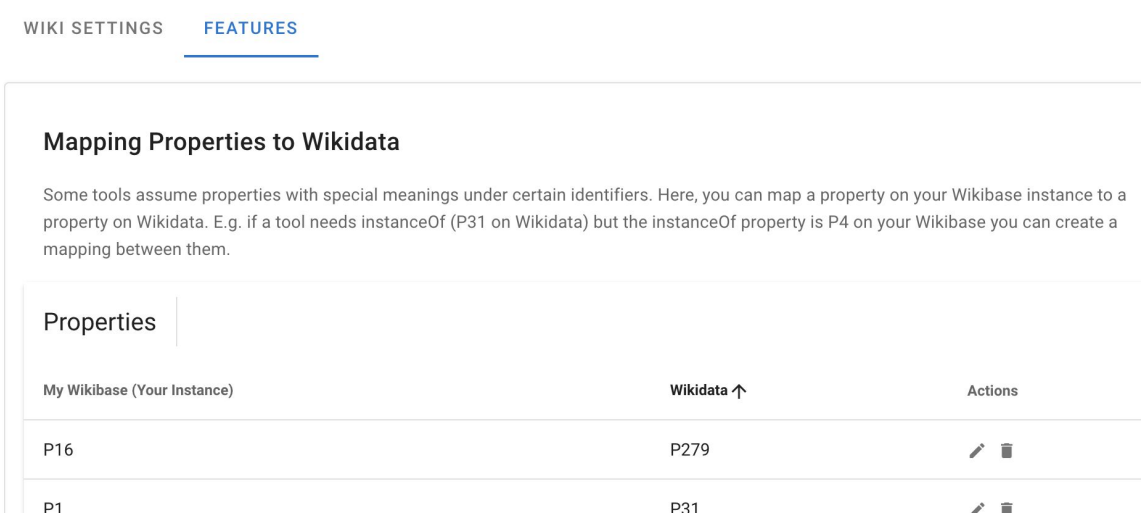
Beyond reconciliation: bulk edits and additions

Note again that our service is not hosted on a web server, so the functionality won’t entirely work. If it did, it would more or less parallel similar Wikidata features in OpenRefine: creating and validating a data scheming, then loading new items and batch edits directly to Wikidata

How local Wikibases can be used with Wikidata

Mappings: map items and properties in your instance to Wikidata

Note: Federated properties is not currently supported in wikibase.cloud



How local Wikibases can be used with Wikidata

https://tinyurl.com/2fqxsldn

Federated queries

lcshItem	lcshItemLabel	localItem	localItemLabel	lcshID	wditem	LCauth
<https://ld4-wbs-test.wikibase.cloud/entity/Q39>	Illegal aliens	<https://ld4-wbs-test.wikibase.cloud/entity/Q40>	Undocumented immigrants	sh2016000739	wd:Q104846145	Illegal immigration

How local Wikibases can be used with Wikidata

And of course… OpenRefine

Join the planning group

Interested in helping to plan the LD4 Wikibase Working Hour?

Email: ld4-wikidata-coord@googlegroups.com

Questions?