How to use Wikidata to build web tools for the social good

Hackathon Session - Office Hour

Houcemeddine Turki (User:Csisc)
Yamen Bousrih (User:Yamen)
Lane Rasberry (User:Bluerasberry)
Disclosure

This work is a part of the “Adapting Wikidata to support clinical practice using Data Science, Semantic Web and Machine Learning” Project, funded by the Wikimedia Research Fund of Wikimedia Foundation.
Here, we like to familiarize Wikimedia users with user-friendly programmatic tools for building Wikidata web applications. Then, we showcase two projects we have been developing based on these tools.
Tools

User-Friendly Programming Tools

- Wikidata Query Service
- Wikibase Integrator
- Wikidata Hub
- Wikimedia Toolforge
- Flask
- ElasticSearch API
- Requests
Wikidata Query Service

Wikidata Query Service is the Wikimedia implementation of SPARQL server, based on the Blazegraph engine, to service queries for Wikidata. Available at https://query.wikidata.org.
**Wikibase Integrator**

A Python module to manipulate data on a Wikibase instance (like Wikidata) through the MediaWiki Wikibase API and the Wikibase SPARQL endpoint. Available at https://github.com/LeMyst/WikibaseIntegrator.

```python
from wikibaseintegrator import wbi_login, WikibaseIntegrator
from wikibaseintegrator.datatypes import ExternalID
from wikibaseintegrator.wbiEnums import ActionIfExists
from wikibaseintegrator.wbi_config import config as wbi_config

wbi_config['USER_AGENT'] = 'MyWikibaseBot/1.0 (https://www.wikidata.org/wiki/User:MyUsername)

# login object
login_instance = wbi_login.OAuth2(consumer_token='<consumer_token>', consumer_secret='<consumer_secret>')</login_instance>

wbi = WikibaseIntegrator(login=login_instance)

# data type object, e.g. for a NCBI gene entrez ID
entrez_gene_id = ExternalID(value='<some_entrez_id>', prop_nr='P351')

# data goes into a list, because many data objects can be provided to
data = [entrez_gene_id]

# Search and then edit an Item
item = wbi.item.get(entity_id='Q141808')

# Set an english label but don't modify it if there is already an entry
item.labels.set(language='en', value='An updated item', action_if_exists=ActionIfExists.REPLACE)

# Set a French description and replace the existing one
item.descriptions.set(language='fr', value='Une description un peu longue', action_if_exists=ActionIfExists.REPLACE)

item.claims.add(data)
item.write()
```
Wikidata Hub

This is a Web hub: it let's you craft URLs to go from an origin to a destination on the web, at the condition that you provide enough information on those points to be identified within Wikidata. Available at https://hub.toolforge.org.

```
import requests
idurl = "https://hub.toolforge.org/P932:"+ws+"?format=json"
idget = requests.get(idurl)
idjson = idget.json()
```
Wikimedia Toolforge

Toolforge is a hosting environment, also known as Platform as a Service. Toolforge makes it easy for you to perform analytics, administer bots, run webservices, and create tools. Available at https://toolforge.org/.
Wikimedia Toolforge

Toolforge is a hosting environment, also known as Platform as a Service. Toolforge makes it easy for you to perform analytics, administer bots, run webservices, and create tools. Available at https://toolforge.org/.

https://wikitech.wikimedia.org/wiki/Help:Toolforge/Quickstart
Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. Available at https://flask.palletsprojects.com/en/2.3.x/.

```python
@app.route('/
def home():
    return render_template('home.html')

@app.route('/link', methods=['GET', 'POST'])
def home02():
    #Getting the Wikidata item of the concerned topic
    wd = request.args.get('id')
    #Recognizing the browser language
    lg = request.args.get('lang')
    if (lg == None):
        language = str(request.accept_languages)
        languages = language.split(',,
        preflg = languages[0]
```

Defining **static** web pages

Defining **dynamic** web pages

Running the web tool

```python
if __name__ == '__main__':
    app.run(debug=True, use_reloader=False)
```
MediaWiki Search API

GET request to perform an advanced search for wiki pages by title or content text match.

```javascript
$(document).ready(function () {
  $('#searchTerm').autocomplete(
    {
      minLength: 2,
      resolver: 'custom',
      events: {
        search: debounce((searchTerm, callback) => {
          var url = "https://www.wikidata.org/w/api.php?callback=?";
          var settings = {
            dataType: 'jsonp',
            data: {
              search: searchTerm,
              action: "wbsearchentities",
              language: "fr",
              uselang: "fr",
              format: "json",
              strictlanguage: true
            }
          };
          $.ajax(url, settings, function (data) {
            callback(data);  
          });
        })
      }
    }
  );
});
```
MediaWiki Search API

GET request to perform an advanced search for wiki pages by title or content text match.

```python
import spacy
from wikibaseintegrator import wbi_functions
import numpy as np

for item in nphr:
    wikitext_id = wbi_functions.search_entities(item, language=lang)
    if (wikitext_id != []):
```
Requests

Requests is an HTTP client library for the Python programming language. Available at https://pypi.org/project/requests/.

```python
>>> import requests
>>> r = requests.get('https://httpbin.org/basic-auth/user/pass', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
```
Web Applications

Wikidata-based web services for the Social Good
Sawtpedia

A generated QR code for a monument that once scanned by mobile allows the user to listen to the Wikipedia page related to that monument in the mobile phone's language.

**QRCode Generation GUI**

- **Wikipedia URL or Wikidata URL**
- **QRCode Generator**
- **QRCode of the Spoken Wikipedia API response for the Wikidata item**

**Spoken Wikipedia API**

- **Wikidata ID**
- **Getting Web Browser Language**
- **Wikidata ID and Language**
- **spoken text audio**
- **spoken text audio [P407]**
- **Spoken Wikipedia File**

**WIKIMANIA SINGAPORE**

If file not found: Generate TTS for the lead of the Wikipedia Page in the web browser language.
Remaining to do

- Resolve Iphones/Ipads issue reading ogg files.
- Adjust the API in order to record number and language of scan
MedCYN

An interactive web tool for clinical decision support based on Wikidata Query Service.

- Rifampicin (wd:Q422652)
- Isoniazid (wd:Q423169)
- Paracetamol (wd:Q57055)

Drug Interactions
Seek medical support

Search

- punarnava -

Hydrocodone with acetaminophen, or hydrochlorothiazide and

WIKIDATA
Remaining to do

- Add support of other natural languages
- Add new features for clinical decision support
Open Call for Contributors

**Wikidata Books:** Creating a tool that generates an e-books based on Wikidata queries.

**Step 1:** The tool should allow the user to select a cover photo from Commons

**Step 2:** The tool should allow the user to define the structure (sections) of the e-book

**Step 3:** The tool should allow the user for each section to define a Wikidata SPARQL query that returns a list of Wikipedia articles that will be extracted and used as content for the section.

**Step 4:** includes DataViz: e.g. maps and timelines in chosen sections of the book

**Use Case: Generation of an e-book for the Medina of Tunis:**

- The user will choose a beautiful photo of the Median from Commons
- The user will define the following sections:
  - **Section 1:** City gates ⇒ Using Wikidata, we can retrieve all the Wikipedia articles related to the city gates
  - **Section 2:** City palaces ⇒ Using Wikidata, we can retrieve all the Wikipedia articles related to the city palaces
  - **Section 3:** ...
Open Call for Contributors

WikiFestival Calendar: Creating an online calendar for UNESCO Cultural Heritage festivals all over the world based on Wikidata

The idea is to use Wikidata in order to generate an online Calendar that lists the festivals all over the world related to UNESCO Cultural Heritage. User should be able to filter by Country.
Conclusion

There are plenty of tools that be used to build Wikidata-based web services.

These tools are easy-to-use and are available free-of-charge.

The most important is to have an idea about a web tool to develop using Wikidata.
Any question

Houcemeddine Turki
User: Csisc

E-mail: turkiabdelwaheb@hotmail.fr
Phone: +21629499418
Twitter: @Csisc1994
LinkedIn: Houcemeddine Turki