

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

Hackathon Session - Office Hour

**WIKIMANIA
SINGAPORE**

Houcemeddine Turki (User:Csisc)
Yamen Bousrih (User:Yamen)
Lane Raspberry (User:Blueraspberry)



Data Engineering and Semantics
هندسة البيانات و دلالاتها



WIKIMEDIA
TN User Group



**WIKI WORLD
HERITAGE**
by WIKIMEDIA



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.

**WIKIMANIA
SINGAPORE**



**WIKIMEDIA
FOUNDATION**

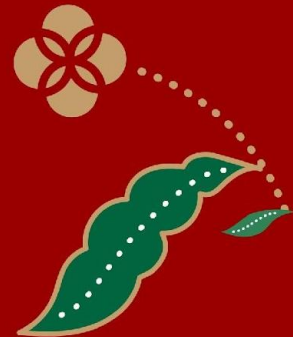
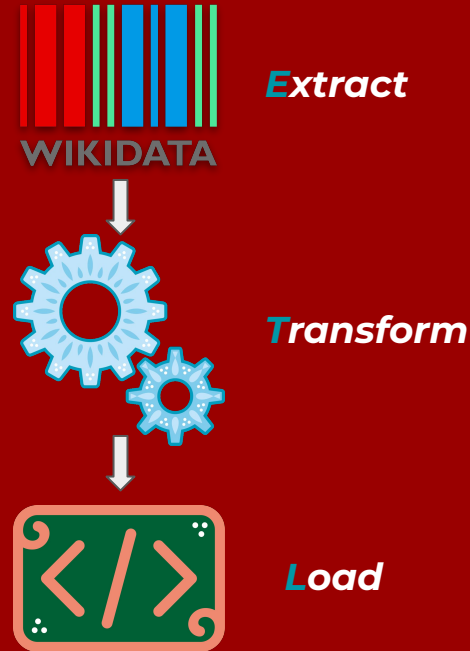


This is an overview

It shows easy tools for developing web solutions based on Wikidata

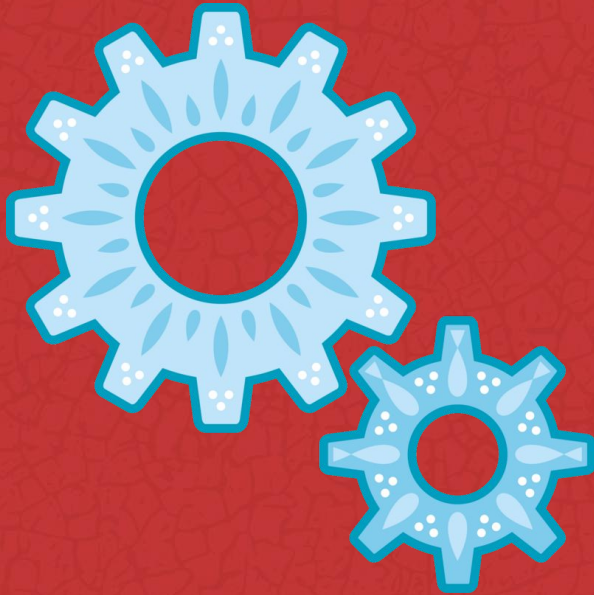
Here, we like to familiarize Wikimedia users with user-friendly programmatic tools for building Wikidata web applications.

Then, we showcase two projects we have been developing based on these tools.



Tools

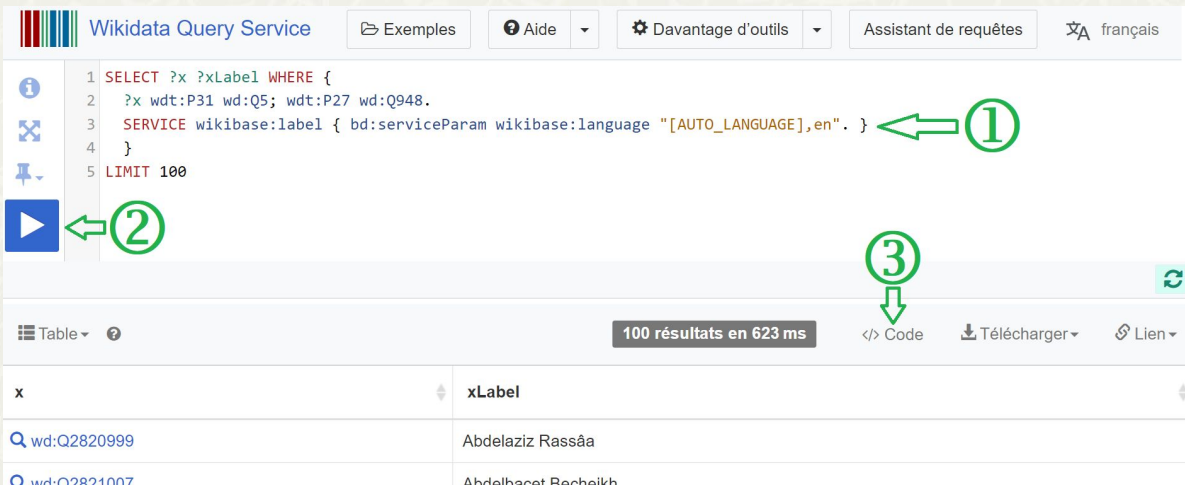
User-Friendly Programming Tools



- *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>.



Wikidata Query Service interface showing a SPARQL query and its results. The interface includes a header with navigation links (Exemples, Aide, Davantage d'outils, Assistant de requêtes, français) and a sidebar with icons for help, refresh, and search. The query is: `SELECT ?x ?xLabel WHERE {
?x wdt:P31 wd:Q5; wdt:P27 wd:Q948.
SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }
}
LIMIT 100`. The results table shows 100 results in 623 ms, with columns x and xLabel. The first two results are: `wd:Q2820999` (Abdelaziz Rassâa) and `wd:Q2821007` (Abdelhacét Bercheikh). A table icon is visible on the left of the results area.

x	xLabel
wd:Q2820999	Abdelaziz Rassâa
wd:Q2821007	Abdelhacét Bercheikh

**WIKIMANIA
SINGAPORE**




Interface showing the embedding code for the query results. The code is: `<iframe style="width: 80vw; height: 50vh; border: none;"
src="https://query.wikidata.org/embed.html#SELECT%20%3Fx%20%3FxLabel%20WHERE%20%7B%0A%20%20%3Fx%20wd%3AP31%20wd%3A%5%3B%20wd%3AP27%20wd%3AQ948.%0A%20%20SERVICE%20wikibase%3A%20%7B%0A%20%3AserviceParam%20wikibase%3Alanguage%20%22%5BAUTO_LANGUAGE%5D%2C%22.%20%20%7D%0A%20%20%7D%0A%20%20%7D%0ALIMIT%20100" refererPolicy="origin"
sandbox="allow-scripts allow-same-origin allow-poppers"></iframe>`

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>.

```
from wikibaseintegrator import wbi_login, WikibaseIntegrator
from wikibaseintegrator.datatypes import ExternalID
from wikibaseintegrator.wbi_enums 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>')

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='Q141806')

# 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.KEEP)

# 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()
```



**WIKIMANIA
SINGAPORE**



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>.

```
← → ↻ 🔒 hub.toolforge.org/viaf:24597135?property=P7704&format=json
```

```
{"origin":{"id":"viaf:24597135","value":"24597135","properties":["P214"],"qid":"Q34981"},"destination":{"properties":["P7704"],"url":"https://data.europeana.eu/agent/base/60043"}}
```

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

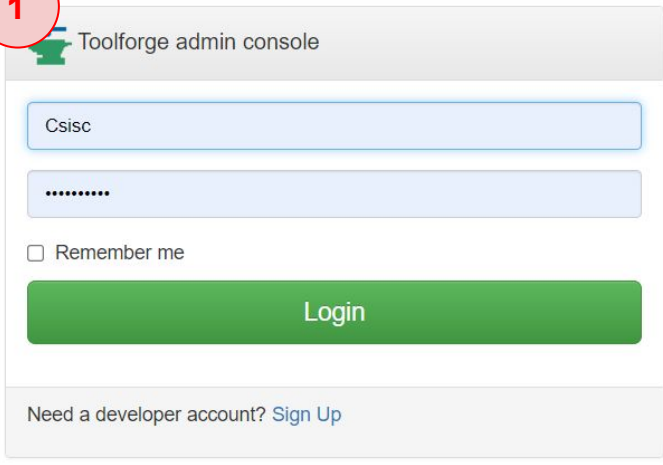


**WIKIMANIA
SINGAPORE**

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/>.

1



Toolforge admin console

Csisc


.....

Remember me

Login

Need a developer account? [Sign Up](#)

2



Git repositories

sawtpedia

+ create repository

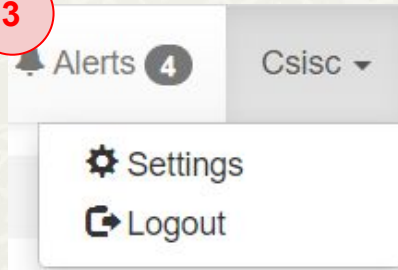
WIKIMANIA
SINGAPORE



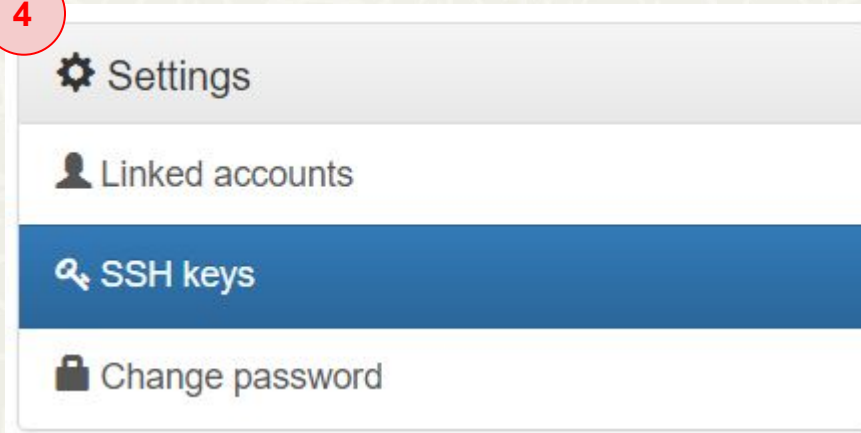
Wikimedia Toolforge

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/>.

3



4



5



<https://wikitech.wikimedia.org/wiki/Help:Toolforge/Quickstart>

**WIKIMANIA
SINGAPORE**



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/>.

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

Defining **static** web pages

```
@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 **dynamic** web pages

WIKIMANIA
SINGAPORE

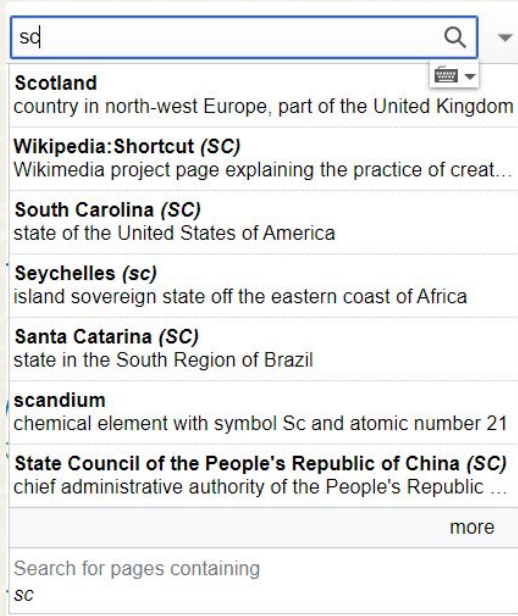
```
if __name__ == "__main__":
    app.run(debug=True, use_reloader=False)
```

Running the web tool



MediaWiki Search API

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



```
$(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  
          }  
        }  
      });  
    }  
  });  
});
```

JS

WIKIMANIA
SINGAPORE



MediaWiki Search API

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

Scotland
country in north-west Europe, part of the United Kingdom
Wikipedia:Shortcut (SC)
Wikimedia project page explaining the practice of creat...
South Carolina (SC)
state of the United States of America
Seychelles (sc)
island sovereign state off the eastern coast of Africa
Santa Catarina (SC)
state in the South Region of Brazil
scandium
chemical element with symbol Sc and atomic number 21
State Council of the People's Republic of China (SC)
chief administrative authority of the People's Republic ...
[more](#)
Search for pages containing
sc

```
3 import spacy
4 from wikibaseintegrator import wbi_functions
5 import numpy as np

91     for item in nphr:
92         wikidata_id = wbi_functions.search_entities(item, language=lang)
93         if (wikidata_id != []):
```



**WIKIMANIA
SINGAPORE**



Requests

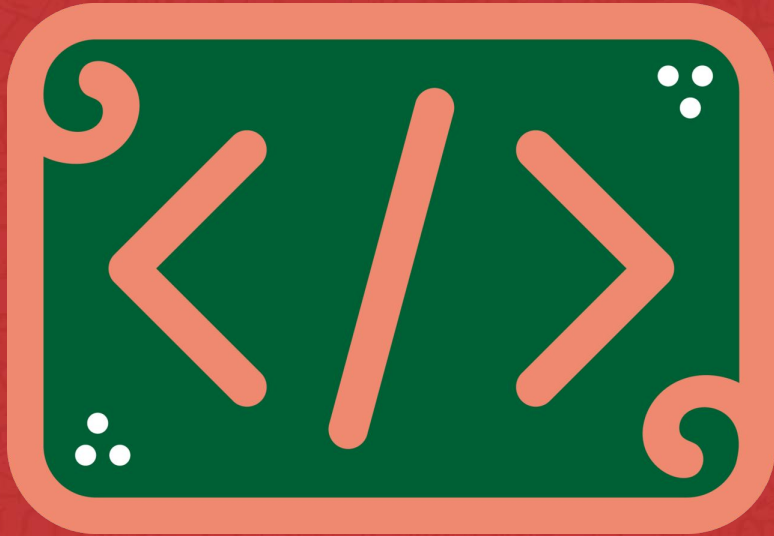
Requests is an HTTP client library for the Python programming language.
Available at <https://pypi.org/project/requests/>.

```
>>> 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'
>>> r.text
'{"authenticated": true, ...}'
>>> r.json()
{'authenticated': True, ...}
```



Web Applications

Wikidata-based web services for the
Social Good



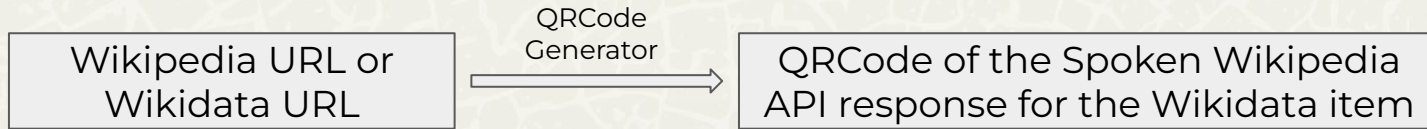
WIKIMANIA
SINGAPORE

Sawtpedia

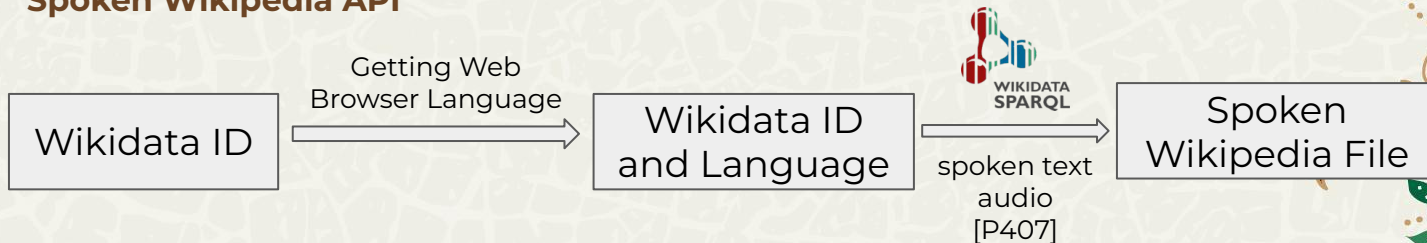
A generated QR code for a monument that once scanned by mobile allows the user to listen to the Wikipedia page related to that monument in the mobile phone's language.



QRCode Generation GUI



Spoken Wikipedia API



**WIKIMANIA
SINGAPORE**

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



Sawtpedia



Remaining to do

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

**WIKIMANIA
SINGAPORE**



Sawtpedia



spoken text audio



 edit

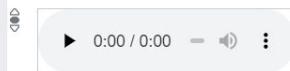
Ribat de Lamta رباط لعمطة.ogg

1 min 43 s; 1,008 KB

language of work or name Arabic

▼ 0 references

+ add reference



 edit

Ribat de Lamta audio text fr.ogg

1 min 6 s; 638 KB

language of work or name French

▼ 0 references

+ add reference

+ add value

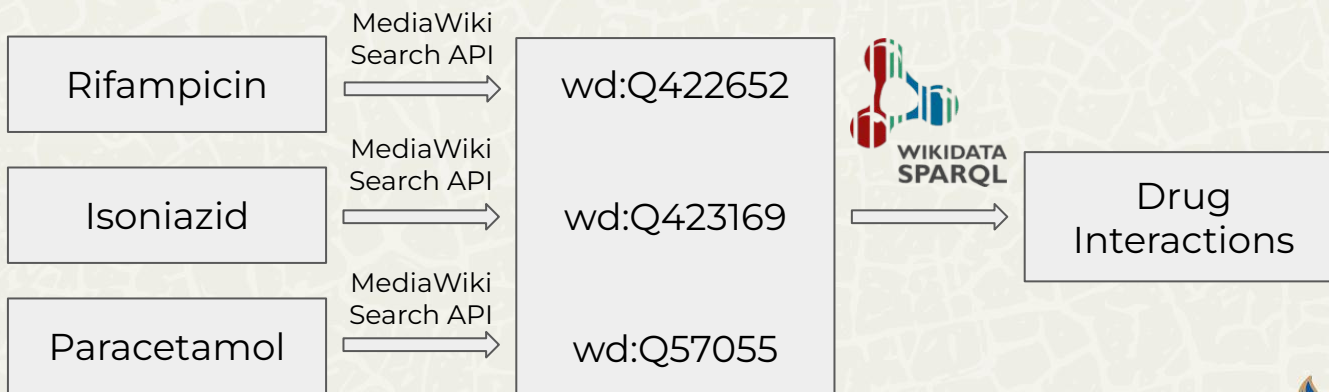


**WIKIMANIA
SINGAPORE**



MedCYN

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



**WIKIMANIA
SINGAPORE**

Search

pubchem.ncbi.nlm.nih.gov/compound/with-antipyrin-but-pain-relieving-properties [95,550]



Seek medical support

with-antipyrin-but-pain-relieving-properties [95,550]



11

Going to the [Wikidata:Commons service](#) (Code: [Commons](#)) (parts of the service code were removed from the default quantity of default) (found on [Wikidata:Commons](#) for content: [CC-0](#) for [text](#), [MIME](#) for [audio](#)) (found on [Wikidata:Commons](#)) (found on [Wikidata:Commons](#))

MedCYN

Remaining to do

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



**WIKIMANIA
SINGAPORE**



Open Call for Contributors

Wikidata Books: Creating a tool that generates an 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 article 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:

**WIKIMANIA
SINGAPORE**



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.

**WIKIMANIA
SINGAPORE**



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

**WIKIMANIA
SINGAPORE**





**WIKIMANIA
SINGAPORE**