Procedure for updating the name resolver for observation facilities.

Contents

1. Observation Facilities Lists

List	Facility Type	number of Records
NSSDC	space	1571
NASA/NAIF	space	307
NASA/PDS	space	510
SPASE	space + ground	1970
SANA	space	1513
AAS	ground	563
Harvard/ADS	ground	2056
IRAF	ground	28
IAU/MPC	ground	2335
Xephem	ground	461
WMO/Oscar	space	683
WISERep (telescopes)	ground	108
Astroweb	space + ground	375

2. Extract data on Wikidata base

Wikidata is a collaborative database that aims to collect and structure data in a multilingual and open way. Anyone can contribute to the project by adding, editing or improving information about an item.

If you need additional information on the structure of the database and our data extractions methods, please refer to the documentation related to Wikidata.

The Python script can be found at:
.../GitHub/FacilityList/data/WIKIDATA/scripts/query_wikidata.py

We select information of interest, such as Naif ID, IAU ID, label, aliases, etc. The output file is located at: ***.../GitHub/FacilityList/data/WIKIDATA/script/extract_wikidata.json***

3. Extraction of data observation facilities lists

We employ a web scraping method to extract data lists. The website parsing is done using the Beautiful Soup Python module, allowing us to retrieve pertinent information such as ID, mission names, and observatory names. The script for each list can be found at :
.../GitHub/FacilityList/data/names_of_list/scripts/extract_data_name_of_list.py.

To update all the lists, execute the bash script at

```
.../GitHub/FacilityList/data/run_all_extractions.sh
```

The output files are allowd in json format at
.../GitHub/FacilityList/data/name_of_list/data/name_of_list.json

4. Comparison of lists

We will compare all the elements that we find in the lists (NSSDC, NAIF, IAU, etc.) with the elements that we have in the Wikidata list. The elements that we find in each list do not have the same identifiers and require us to build a script that is adaptable to the data (NAIF_ID, NSSDC_ID, pds_id).

The Python scripts for comparing the lists are located in each folder of the respective lists at:
.../GitHub/FacilityList/data/name_of_list/scripts/compare_name_of_list-wiki.py"

To initiate all comparisons, you can use the bash script located at:

```
.../GitHub/FacilityList/data/run_all_extractions.sh
```

NB : To compare the lists, we'll be using the fuzzywuzzy library, which searches for string matches. A score is assigned to determine the level of satisfaction with the match, with a score of 100 indicating a perfect match

The output files `tres_certain_name_of_list` and `non_trouves_name_of_list` are allowed in each folder of the list :
.../GitHub/FacilityList/data/name_of_list/script/...

4.1 Results of comparison

The results are sorted according to the score. If the score is above 400, the comparison is a perfect match, and all results will appear in the `tres_certain` file. If the score is below 400, then the results will appear in the `non_trouve_name_of_list` file.

4.2 Results analysis

4.2.1 Verification of the found elements
Firstly, we verify if the results in the file `tres_certain` are coherent.

4.2.2 Verification of the unfound elements

We perform a quick search to determine the reasons why the elements are not found in the wikidata list.
Several cases may occur:

-The entry does not exist, so we need to create a new item in wikidata.
-The element exists on wikidata but does not appear in the query result. The ID or name is missing (add it to wikidata).

Before creating a new entry on wikidata, it is imperative to check if the element does not already exist or if it has another name.

We will start by querying the Wikidata knowledge base. For example, we will write a query that lists all the elements that have an NSSDCA identifier.

Here are some examples:
https://nssdc.gsfc.nasa.gov/planetary/chronology.html
(https://nssdc.gsfc.nasa.gov/planetary/chronology.html)
https://ofrohn.github.io/seh-doc/list-missions.html (https://ofrohn.github.io/seh-doc/list-missions.html)
https://en.wikipedia.org/wiki/List_of_observatory_codes
(https://en.wikipedia.org/wiki/List_of_observatory_codes)

4.3 Adding on Wikidata base

For adding information to Wikidata, there are several methods available. If you have a few elements, manual addition is an option. However, if you have a large number of elements, it’s preferable to use a mass-addition tool.

Please refer to the documentation for instructions on how to use Wikidata.

5. Importing database in Elasticsearch

Reloading the ‘obsfacility_index’ index

1. Place the JSON file “extract_wikidata” in `/root/obsfacility_data/`

from another machine connected to the ObsParis network:

```
scp extract_wikidata.json root@voparis-elasticsearch:/root/obsfacility_data/extra
```

2. Connect to voparis-elasticsearch:

```
ssh root@voparis-elasticsearch
```

5. (Re)create the index with the script “create_db.py”:

```
python3 /share/elasticsearch/scripts/db_create.py --index obsfacility_index --map
```

3. Fill the index:

```
python3 scripts/obsfacility_insert.py obsfacility_index obsfacility_data/extract_
```