

NUTS Documentation

This text is an excerpt from a longer document “D04.01: Persistent URIs for NUTS”, which was created by PwC EU Services under the ISA² Programme.

1.1 Technical documentation

This section provides technical details such as the programming language used, type of database used, where to download or access the tool, etc.

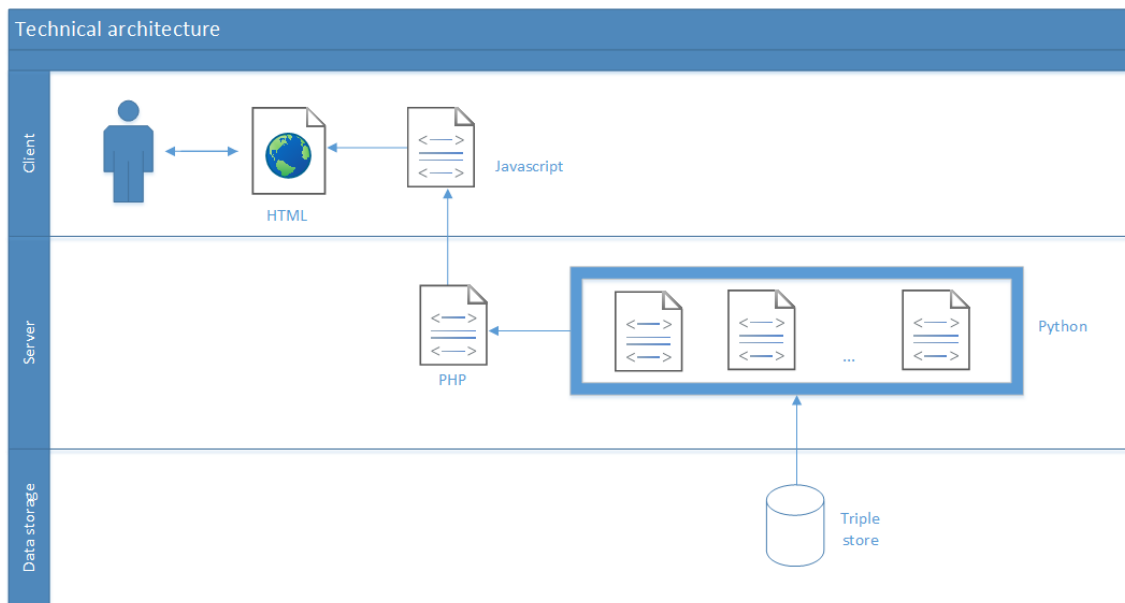


Figure 1 - Technical architecture

Figure 1 depicts a high-level overview of the pilot and the technologies used:

- HTML, to implement the user interface of the pilot;
- Javascript, to update the visualisation;
- PHP, to execute the python scripts;
- Python, to query the triple store in order to get information using SPARQL; and
- Triple store where the NUTS information is stored according to the data model implemented (see section **Error! Reference source not found.**).

Client

The user accesses the map visualisation through an **HTML** website, which is deployed in a commercial cloud server. The HTML uses **javascript** to execute the actions, which are mainly drawing the map and updating the regions according to the year and NUTS level selected.

Server

The javascript needs to retrieve stored information from the database. To do so, the pilot executes **python** scripts which contain the necessary queries. Since the javascript is on the client side and the database on the server, there is a **PHP** file in between the javascript and python files. Depending on the input arguments, the PHP file executes the corresponding python script to get the data.

Data storage

The python scripts query the triple store using the SPARQL query language. The database that the pilot uses is a Virtuoso triple store and the graph "http://52.50.205.146/NUTS-codes". The update of the map and evolution of codes is checked in the database via the available SPARQL endpoint:

<http://52.50.205.146/sparql>

Input files

The pilot uses the NUTS geometries that are published in the official Eurostat website¹, i.e. it downloads the geometries published by Eurostat and stores them locally to draw the map. The geometries are available in different formats, of which the pilot uses GeoJSON (60M). The geometries are stored in a javascript file per NUTS level and year.

In addition, the pilot takes as input the list of NUTS codes and changes from 2010-2013 and 2013-2016 that are published in the Eurostat NUTS history website². The spreadsheets are downloaded from the Eurostat website and transformed into NUTS RDF (see section **Error! Reference source not found.**). The transformation is performed by using Google Refine and python scripts created to automatically generate the triples. This procedure is explained in section 1.3 in more detail.

It is important to take into account that the geometries for 2016 NUTS codes are not published yet. However, the spreadsheet with the NUTS changes from 2013 to 2016 is available. Therefore, the pilot 2016 map is drawn using the 2013 geometries, but it reflects the changes from 2013 to 2016.

Structure of the source code

The source code can be found on GitHub:

https://github.com/SEMICEu/NUTS_pilot

The structure of the source code consists of:

- datasets – Folder with the files storing the NUTS geometries, which are transformed into JSON variables. The folder contains a file with a JSON variable per NUTS year and level (e.g. "nuts0_2010_data" contains the NUTS geometries of the NUTS level 0 in 2010);
- scripts – Folder with the scripts to be called and executed from the main page and css configuration. It consists of:
 - map.css – CSS file with the website layout;
 - draw_map.js – javascript file with the main functions, i.e. the map drawing and update of colours;
 - queries.js – javascript file with the functions to get the information from the triple store;
 - sliders.js – javascript file with the functions needed to visualise and update the map information when the year slider is changed;
 - languages.js – javascript file to support multilingualism. For the moment, it only contains information for the display of information in English;
 - callScript.php – PHP file that executes the python scripts;
 - code_changed.py – python script to query the database in order to get the list of NUTS URIs that changed from one year to other (split, merged

¹ <http://ec.europa.eu/eurostat/statistical-atlas/gis/viewer/distribution/v1/>

² <http://ec.europa.eu/eurostat/web/nuts/history>

- and replaced relations). It stores the changes in a JSON file called 'data_year.txt' (one per year);
- discontinued.py – python script to get the codes that are discontinued from 2013 to 2016. This script is only needed because we do not have the geometries corresponding to 2016 and thus, the 2013 geometries have to be linked with the 2016 codes;
- list_changes.py – python script to check the changes of the NUTS code from a year, linked to the type of change and value, i.e. URI of the previous NUTS code. It creates a file 'list_changes_year.txt' with the result from the query in JSON;
- list_props.py – python script to store all the properties and its values for all the NUTS code of a year in the file 'props_year.txt'. This script is created to optimise the user experience. It should be executed if there are changes in the triple store or if new geometries are included;
- merged.py – python script that stores the merges from a year to the next year (e.g. from 2013 to 2016) in the file 'changes_merged_year.txt'. This script is created to optimise the user experience;
- replaced.py – python script similar to the merges, but for replacements from a year to the next;
- split.py – python script similar to the merges, but for splits from a year to the next;
- scripts_spreadsheet – Folder with the scripts needed to generate the triples of new codes from the changes spreadsheet. It consists of:
 - NUTS.py – script to generate the changes triples, looking for keywords like "replaces", "split", etc.;
 - RDF Skeleton – skeleton to get the triples from the spreadsheet with the basic information using Google Refine;
- map.html – Main page of the pilot.

1.2 Deployment

To deploy the pilot locally or in other server, the following list of prerequisites need to be fulfilled:

- For the client side, the pilot HTML needs to be available via a public service (we used xampp in the local implementation). Currently, the pilot is deployed in an Amazon server;
- As technologies, the server needs to have installed python and php;
- For the data storage, the pilot needs to have installed a triple store. Currently, it uses Virtuoso.

To deploy it locally (i.e. having a webserver running on localhost) or in another server, the following steps need to be taken:

1. Download the code from GitHub³;
2. Extract the code under the public directory on the server; and
3. Execute the scripts to generate the supportive txt files with the JSON variables: code_changed.py, discontinued_2016.py, list_changes.py, list_props.py, merged.py, replaced.py and split.py. All the scripts, except the "discontinued" one, have to be executed as many times as there are years that are to be displayed in the pilot (e.g. having as argument 2010, 2013 and 2016 respectively).

³ https://github.com/SEMICEu/NUTS_pilot

1.3 Adaptability

The pilot can be adapted or extended by adding more geometries, like the ones from 2016, or storing more changes from a year to another received in a spreadsheet.

To add new geometries in GeoJSON, the following steps are needed:

1. Download the file in GeoJSON⁴;
2. Create a new file under *datasets* with the declaration of the variable and having as value the GeoJSON data;
3. Include the file in the HTML head;
4. Modify the function "get_selected_NUTS" under "draw_map.js", adding a case for the new year; and
5. Add a new value in the slider. To do so, add the year in the variable "quartersList" in the file "sliders.js".

Note that it is possible that the NUTS geometries contain countries that are not part of Europe, such as Turkey, Norway, Switzerland or Montenegro. In case that you do not want to show them in the visualisation, you need to remove the codes and related coordinates from the countries.

If new geometries are added, the triple store should be updated with the NUTS codes and the corresponding changes. These changes are published in a spreadsheet⁵ by Eurostat. To include them in the triple store, follow the readme file in the *scripts_spreadsheet* folder.

⁴ <http://ec.europa.eu/eurostat/statistical-atlas/gis/viewer/distribution/v1/>

⁵ <http://ec.europa.eu/eurostat/web/nuts/history>