ISEL \ DEETC

MEIM | MEIC

SIGM | CSI – Guia aula prática

Paulo Trigo Silva

1. Search for the tile-map coordinates of ISEL

Consider the "`https://wiki.openstreetmap.org/wiki/Tile_servers`" page with a list of online raster tile-map servers based on OpenStreetMap data.

a) Consider the "`https://wiki.openstreetmap.org/wiki/Tile_servers`" page with a table that lists the online raster tile-map servers based on OpenStreetMap data. Notice the table's column named "tiles url" that shows the format to be used in order to get info from each server.

   Use the server identified as the "OpenStreetMap's Standard tile layer" (first row in that table) and point your Browser to the zoom-level 0 (zero) as provided by that tile-map server. You should get a small square like this:

b) Now, get the 4 tiles of zoom-level 1 and choose the one that contains Portugal.

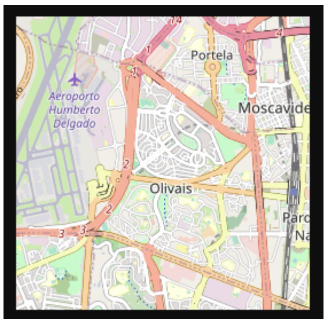
c) Choose, at the zoom-level 2, the tile that contains Portugal.

d) At zoom-level 3 Portugal is contained in 2 tiles. So, at zoom-level 3, what are the X and Y tile coordinates of each (of the 2 tiles) that contains Portugal?

e) Now, focus on Lisbon city and go to zoom-level 4, 5 and 6. So, at zoom-level 3, what are the X and Y coordinates of Lisbon city?

f) Now, focus on "Aeroporto Humberto Delgado" (in Lisbon) and zoom until level 9. So, at zoom-level 9, what are the X and Y coordinates of "Aeroporto Humberto Delgado"?

g) Go with zoom until level 13 where you get the image below. So, at zoom-level 13, what are the X and Y coordinates of "Aeroporto Humberto Delgado"?.



SIGM | CSI – Guia aula prática

Paulo Trigo Silva

2. Build a Web client of tile-map server(s)

The “MapLibre GL JS” is the open source framework that we will be using to build tile-map (both raster and vector) Web clients; cf., “`https://maplibre.org/projects/`”.

Consider the information in the folder “`_MapLibre_GL_JS`”.

a) Complete the “`a01_map_tile_example_maplibregl_raster.html`” client code to ask for tile-maps form the “`tile.openstreetmap.org`” server using the “`xyz`” scheme and the “`png`” format; also define the value of 22 as the maximum zoom level.

b) Complete the “`a02_map_tile_example_maplibregl_raster-multi-source.html`” client code to ask for tile-maps from 2 sources simultaneously: a) the same source as in the previous exercise, and b) form the “`basemaps.cartocdn.com/rastertiles/dark_all`” server using the “`xyz`” scheme and the “`png`” format; also define the value of 22 as the maximum zoom level; notice that there are additional places where to fill some missing code.

3. Explore the micro-service (“container”) approach

You may use the PostgreSQL (and PostGIS) already installed in your machine and skip this item. Or you may follow this goal of exploring micro-services (using Docker support) and execute this item.

Consider the information in the folder “`__info_install-and-use`”.

a) Follow instructions in “`_install_01_PostGIS-pgAdmin_with-docker.txt`”.

4. Install and use a “tile-server”

Consider the information in the folder “`__info_install-and-use`”.

a) Follow instructions in “`_install_02_pg_tileserv_with-docker.txt`”.

5. Load vector-data and serve it with a “tile-server”

Consider the information in the folder “`__info_install-and-use`”.

Consider the information in the folder “`_MapLibre_GL_JS`”.

a) Follow instructions in “`_install_03_load-vector_data_with-docker.txt`”.

b) Complete the “`a03_map_tile_example_maplibregl_vector_pg_tileserv.html`” client code to ask for tile-maps from your local server (named “tileserv”).

c) Adapt the code so that <u>only</u> the <u>country name</u> is shown when user clicks (right button) over it.

ISEL \ DEETC

MEIM | MEIC

SIGM | CSI – Guia aula prática

Paulo Trigo Silva

6. Rebuild all the micro-service approach using "docker-compose"

Consider the information in the folder "`__info_install-and-use`".

a) The "docker-compose" enables to specify and launch a whole micro-service setup from a single file. Recall your setup while following "`_install_04_config-using-docker-compose.txt`".

b) Redo the previous questions to check if everything is again working properly.

7. Install a python virtual environment (venv)

Consider the information in the folder "`_MapServer_simple`".

a) Follow instructions in "`_00_README.txt`".

8. Implement a (simple) tile-server (`MapServer_simple`)

Consider the information in the folder "`_MapServer_simple`".

Consider the python (partial) implementation in "`tileMapServer_simple.py`".

a) In that python code, adapt the dictionary named "`DATABASE`" to your own configuration.

b) Complete the "`isValid`" method. Hint: <`your-code-here`> tag and theoretical lesson.

c) Complete the "`TileEnvelope`" constructor.

d) Complete the "`SELECTmapVectorTile`" method.

e) Complete the "`SELECTenvelope`" method.

9. Build a Web client of the `MapServer_simple`

Consider the information in the folder "`_MapLibre_GL_JS`".

a) Analyze "`b01_map_tile_example_maplibregl_vector_map_server_simple.html`" code and execute (just drag into a Browser).

b) Get "`b02_map_tile_example_maplibregl_vector_raster_map_server_simple.html`" and complete the code so that it combines both the vector tiles (as in the previous exercise) and the raster tiles (already coded in this file).

10. Explore some aspects of the Web client and `MapServer_simple`

a) In "`b02_map_tile_example_maplibregl_vector_raster_map_server_simple.html`" comment the currently used "`vector_url`" and uncomment the currently commented one.

b) Change the "`tileMapServer_simple.py`" so that <u>only</u> the <u>country name</u> is shown when user clicks (right button) over it.

c) Repeat the previous exercise but now configure the change in the Web client.