How to create a sumo simulation using an osm file.

1. Open https://download.geofabrik.de/ and download file in .osm.pbf format. For example I will use Stuttgart region. After that change the name of file if it consist '-' In my example Stuttgart-regbez-latest.osm.pbf → Stuttgart.osm.pbf
2. After that osmium tool extract is needed. Here is how I have installed it
   - download mambaforge here https://github.com/conda-forge/miniforge/releases/latest/download/Mambaforge-Windows-x86_64.exe
   - while installing it, make sure to select to the "Add Mambaforge to my PATH environment variable"
   - make sure all git bash/powershell/cmd instances are closed
   - open up cmd as admin
   - inside of cmd, type "mamba install osmium-tool", press 'Enter'
   - Osmium should be installed and available to use from anywhere in the command line! Just type "osmium --version" to verify the installation.
3. Exporting a smaller area from an .osm.pbf file and also changing the format to .osm for that 4 coordinate is needed
   - Open https://www.openstreetmap.org/export#map=12/48.7553/9.3214
   - Then use option to manually choose the area

   - After getting the coordinates Open cmd from directory, where .osm.pdf File is. Arrows are showing the order for writing coordinates in the cmd
   - Ussing command 'osmium extract -b 9.19,48.7,9.43,48.8 stuttgart.osm.pbf -o esslingen.osm' where stuttgart.osm.pbf is our file from the beginning and Esslingen.osm is an output file.
4. After that Sumo Saga, that is already installed in Sumo,is needed. The location is: C:\Program Files (x86)\Eclipse\Sumo\tools\contributed\saga if

the Sumo isn't installed then application wouldn't run properly. Installation guide SUMO: https://sumo.dlr.de/docs/Downloads.php

5. Copy all files from saga to directory with Esslingen.osm file
   - Installing a module to make it work faster pip install rtree
   - After that using command python scenarioFromOSM.py --osm esslingen.osm --out test where out is a folder, which will be created after everything.
6. In the end the folder test is created. Take all the files from that into our project repository and change already existing files in Retrieve and Without_Transport.
7. After that open the OSM.sumocfg file using text redactor or Pycharm and into the <addition files> add file data.rou.xml

```
<input>
    <net-file value="osm.net.xml"/>
    <route-files value="osm_pt.rou.xml"/>
    <additional-files value="osm_polygons.add.xml,basic.vType.xml,osm_stops.add.xml,osm_complete_parking_areas.add.xml,
    osm_parking_rerouters.add.xml,osm_taxi_stands.add.xml,osm_taxi_rerouters.add.xml, data.rou.xml"/>
</input>
```

8. One of the last steps is to adjust the time of available Public Transport
   - Open the osm_pt.rou.xml file
   - Scroll to the end of the file where Flows are
   - Use find tool to locate flows

```
<flow id="pt_bus_103:2" type="pt_bus" route="pt_bus_103:2" begin="14682.0" end="86682.0" period="600" line="103:2"
    <param key="name" value="Bus 103: Zell Johannesstraße =&gt; Zell Albblick =&gt; Hedelfingen"/>
    <param key="completeness" value="0.52"/>
</flow>
```

   - End parameter is needed. In Pycharm everything is simple and all parameters can be changed at once.
   - The time in the end parameter is in seconds, so keep that in mind
   - Parameter flow after changing end value

```
<flow id="pt_bus_103:0" type="pt_bus" route="pt_bus_103:0" begin="14622.0" end="36000000.0" period="600" line="103:0"
    <param key="name" value="Bus 103: Zell Alleenstraße =&gt; Hedelfingen"/>
    <param key="completeness" value="0.52"/>
</flow>
```

9. Finally, the code is ready to work with new Map 😊