# **Annex - 4**

# 4. Developer Manual

This section serves as an installation guide describing which steps to take to set up the application.

## 4.1 Geographic Database

### 4.1.1 Set up PostgreSQL and PostGis

The geographic database is used by the chatbot application to retrieve and manage geographical data. In this project, the database runs on a Virtual Machine using Ubuntu 16.0.4 LTS. Therefore, Ubuntu’s terminal is used to install most of the software. In order to make sure Ubuntu has access to the current package index, it is advised to execute an update command before installing the software:

sudo apt-get update

The first step is to install the data management system, PostgreSQL. To install the version used in this project, the following command is used:

sudo apt-get install -y postgresql=9.5+173 postgresql-contrib=9.5+173

Then, the database *“touristdb”* is created as well as the managing user, which is called *“touristuser”*. The createuser command will prompt for a password which can be chosen by the developers.

sudo -u postgres createuser -P touristuser

sudo -u postgres createdb –owner **touristuser** touristDB

**sudo -u postgres psql -c "grant all on database touristDB to touristuser" touristDB**

Now we have set up the database, the PostGIS extension is installed and added to make the database able to handle geospatial data.

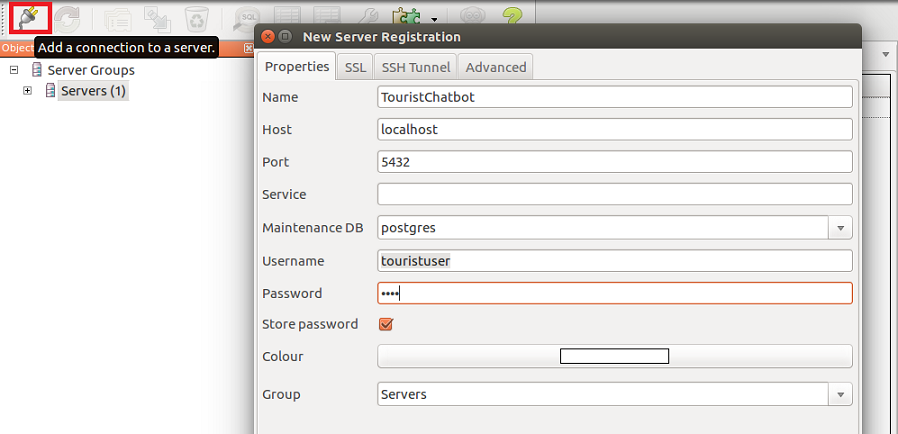
sudo apt-get install -y postgis postgresql-9.5-postgis-2.2

sudo -u postgres psql -c "CREATE EXTENSION postgis; CREATE EXTENSION postgis\_topology;" touristDB

The next step is optional, but seems convenient if the developers want to manage their database with the help of a user interface. The managing tool pgadmin facilates running and editing SQL queries and viewing the stored data.

 sudo apt-get install pgadmin3

To access the database in pgadmin3, a connection to the server must be added, which can be realized by clicking the plug button in the upper toobar and then entering the following values.



In the object browser, the database schemas can be viewed accessing TouristChatbot -> databases -> touristDB.

### 4.1.2 Import Data into Database

Now that we have set up the database, it needs to be filled with geospatial test data. In this project, test data of the city Hamburg, Germany is used which can be downloaded as a .pbf file from the website <http://download.geofabrik.de/europe/germany/hamburg.html>, or [here](http://download.geofabrik.de/europe/germany/hamburg-latest.osm.pbf) as a direct download link. Then, the tool Osmosis is used to import the OSM data which can be installed by the following command:

sudo apt-get install osmosis

The next commands prepares the database for the osmosis import. It sets the hstore extension and the pgsnapshot database schema which causes that all relevant tag data are stored in a hstore column.

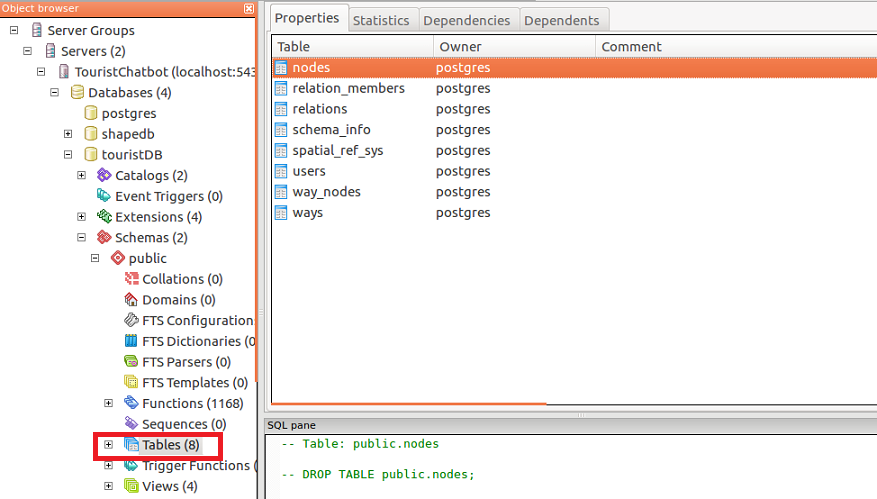
sudo -u postgres psql -c "CREATE EXTENSION hstore;" touristDB

psql -U touristuser -d touristDB -f /usr/share/doc/osmosis/examples/pgsnapshot\_schema\_0.6.sql

After that, the import itself is realized. Remember to execute this command in the folder where the downloaded .pbf file is situated and to add the corresponding password (which is set by the developer in the previous step of this manual).

osmosis --read-pbf file="hamburg\_germany.osm.pbf" --write-pgsql host="localhost" database="touristDB" user="touristuser" password=*’password’*

In order to see if the import was successful, pgadmin3 can be used to take a look at the now imported data. Again, this step is optional. In the object browser, the database tables can be viewed accessing TouristChatbot -> databases -> touristDB-> Schemas -> public -> Tables.



If you open the context menu on one of the tables (e.g. nodes) by clicking right, the option *view data* is available which shows the previously imported data.

# Bibliography

<http://www.paulshapley.com/2016/04/how-to-install-postgresql-95-and.html>