

Using the Geoparser Framework

Ludwig Richter (ludwig.richter@posteo.de)

March 24, 2017

In this document, the installation of the Geoparser Framework and its usage is described. In the following it is assumed that you are using Eclipse as IDE (other IDEs may be used, of course, but for simplicity I focus on Eclipse only).

1 Install Java 8 JDK

The geoparser software requires Java 8. Install the latest Java 8 JDK from <http://www.oracle.com/technetwork/pt/java/javase/downloads/index.html> to be able to compile the source code.

2 Install Eclipse and Plugins

2.1 Eclipse

Install the latest Eclipse IDE for Java Developers from <http://www.eclipse.org/downloads/packages/>.

2.2 Configure Java Compiler

Ensure that the Java 8 JDK is used as JRE in Eclipse. Check that it is listed in `Window|Preferences|Java|Installed JREs`. If not, add its installation path via `Add...`

2.3 Optional: SVN Plugin

If you get the source code via SVN, an extra plugin is required in Eclipse. Install `Subversive - SVN Team Provider` via `Help|Eclipse Marketplace...` When you import a project via SVN the first time, you will be prompted to select a `Subversive Connector` implementation. `SVN Kit` works just fine.

2.4 JavaFX Plugin

Install the plugin `e(fx)clipse` via `Help|Eclipse Marketplace...` to enable proper JavaFX support within Eclipse.

3 Other Optional Plugins

Other useful plugins for programming may be `JSON Editor Plugin`, `FindBugs Eclipse Plugin`, and `EclEmma Java Code Coverage`. All of them can be installed via `Help|Eclipse Marketplace...` I used them while developing the framework...

3.1 Optional: SceneBuilder

If you want to implement JavaFX applications (i.e., programs with a GUI), you may want to install SceneBuilder from <http://gluonhq.com/labs/scene-builder/#download>. This tool provides a comfortable JavaFX editor that works well with Eclipse. In Eclipse, go to `Window|Preferences|JavaFX` and set the path to the executable of your SceneBuilder installation. Afterwards, you can open `FXML`-files by right-clicking on them in the Package Explorer view and selecting `Open With Scenebuilder`.

Note: for the viewer apps, the third-party `controlsfx` package is used. In SceneBuilder, you can add the library by clicking on the configuration button next to the Library search field. Click `JAR/FXML Manager`. Add the package by clicking on `Search repositories` and installing the latest version.

4 Download Source Code

If you got the source code directly as an archive or project, you can import the project via `File|Import...|Archive or File|Import|Existing Projects into Workspace`, respectively.

If you get the source code via SVN or Git, you can simply import the project via `File|Import...|Project from SVN` or `File|Import...|Project from Git`.

5 Install GeoNames Database

In order to fill the gazetteer with data, you need to load a GeoNames dump into a PostgreSQL database. In the project's source code, see `/GeoParser/docu/geonames_installation.pdf` for instructions on how to populate the GeoNames database in PostgreSQL.

6 Create Gazetteer Database

You may either use the same PostgreSQL server, where you created the Geonames database, or a separate PostgreSQL server. It does not matter... Create a database called `gazetteer`. Then, install the following required extensions:

```
CREATE EXTENSION IF NOT EXISTS postgis;
CREATE EXTENSION IF NOT EXISTS fuzzystrmatch;
```

Note: the table schema will be automatically generated by the geoparser via JPA, you don't need to do anything manually.

7 Configure Geoparser

The Geoparser Framework can be configured by editing the file: `/GeoParser/src/main/resources/geoparser.config.json`. Alternatively, you can use `/GeoParser/src/main/java/de/unihd/dbs/geoparser/GeoParserConfig.java` to edit the configuration file programmatically. The JavaDocs for `GeoParserConfig.java` also contain hints on how to format the file.

Create/modify the `dbConnectionConfigurations` entries that contain the connection details to GeoNames and the gazetteer, e.g.:

```
"local.geonames": {
  "dbConnectionData": {
    "dbName": "geonames",
    "host": "localhost",
    "port": 5432,
    "userName": "postgres",
    "password": ["S", "O", "M", "E", "_", "P", "A", "S", "S", "W"],
    "authenticationRequired": true
  },
  "sshConnectionData": {
    "sshRequired": false
  }
},
"local.gazetteer": {
  "dbConnectionData": {
    "dbName": "gazetteer",
    "host": "localhost",
    "port": 5432,
    "userName": "postgres",
    "password": ["S", "O", "M", "E", "_", "P", "A", "S", "S", "W"],
    "authenticationRequired": true
  },
  "sshConnectionData": {
```

```
        "sshRequired": false
    }
}
```

Now, add/modify the following `configurationStrings` entry:

```
"geonames.source": "local.geonames"
"gazetteer.persistence_unit.db_source": "local.gazetteer"
```

For experts only: The property `gazetteer.persistence_unit.name` contains the label for the JPA persistence unit to be used. Persistence units can be modified in `/GeoParser/src/main/resources/META-INF/persistence.xml`.

8 Populate Gazetteer Database

Run the program `/GeoParser/src/main/java/de/unihd/dbs/geoparser/gazetteer/importers/GazetteerInstaller.java`. This program first creates the gazetteer table schema.

WARNING: BE AWARE THAT THIS WILL DELETE ANY EXISTING GAZETTEER DATA IN THE DATABASE!!!

To increase bulk-insertion performance, all foreign-key constraints and indexes are deleted. Next, all types defined in `de.unihd.dbs.geoparser.gazetteer.types` are imported. Then, the GeoNames data are imported. Finally, all foreign-key constraints and indexes are restored after all data have been successfully inserted.

Note: This task will take several hours to complete! On a i7-4700MQ @2.4GHz with 16GB RAM it took 3.5 hours. Ensure that you have 4GB available memory and 11-12GB of free data storage for the PostgreSQL database!

9 Use the GUIs to Get Started

You now should be able to view and search the gazetteer data using the `Gazetteer Viewer` provided in `/GeoParser/src/main/java/de/unihd/dbs/geoparser/gazetteer/viewer/GazetteerViewerApp.java`. To get an impression of how geoparsing works, you can use the `Geoparser Viewer` provided in `/GeoParser/src/main/java/de/unihd/dbs/geoparser/viewer/GeoparserViewerApp.java`.

10 Developing new Modules and Architectural Details

This section will come soon (May-Juni). For now, please refer to the JavaDoc documentation provided directly with the source-code. Also, the unit tests and the GUI code should give you a good starting point for understanding how things work.

11 Literature

Getting started with JavaFX: <https://docs.oracle.com/javase/8/javafx/JFXST.pdf>