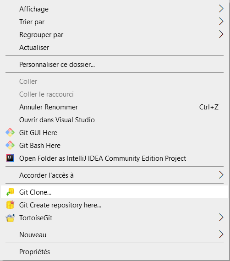
# 1/Installation

The source code is available on the UNIGE gitlab repository: to retrieve it, run a gitclone command on the following repository address:

<https://gitlab.unige.ch/Giovanna.DiMarzo/coordination-energy>



## 1.1/SAPERE configuration database

This object database is used to configure the local node: node name, its address (IP, port) and the addresses of direct neighboring nodes. It is a DBMongo database which is of object type (non-relational).

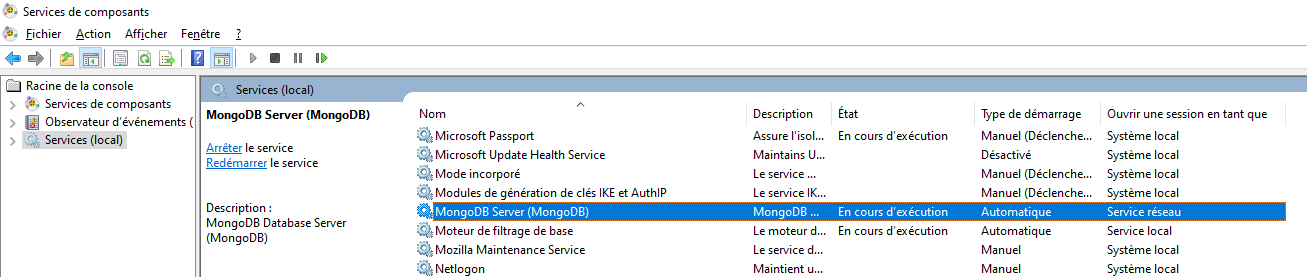
### ~~1.1.1/Installation of MongoDB server~~

~~- Install MongoDB service from:~~

[~~https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows~~](https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows)

~~- Follow these steps to install MongoDB Community Edition using the MongoDB Installer wizard. The installation process installs both the MongoDB binaries as well as the default~~ [~~configuration file~~](https://docs.mongodb.com/manual/reference/configuration-options/) ~~<install directory>\bin\mongod.cfg.~~

~~-During the installation process, Mongodb can be configured as a Windows service started automatically. This avoids having to start the server after each restart of the machine.~~



### ~~1.1.2/Installation of MongoDB client~~

~~A user interface allows to feed and request the MongoDB database~~

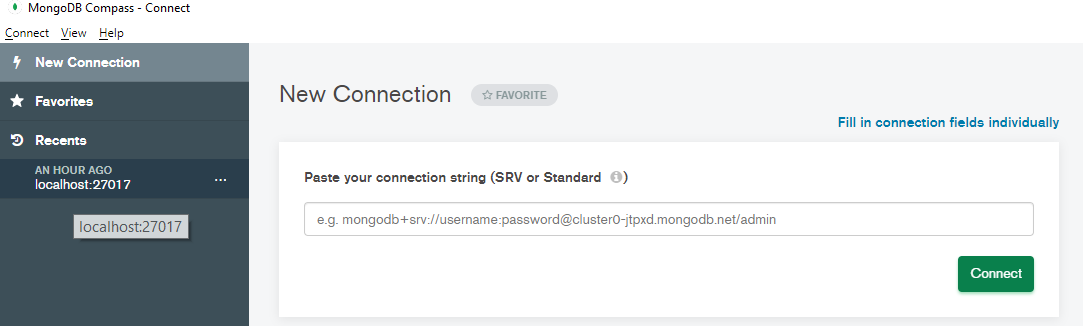
* ~~Install “MongoDB compass”: the installer can be downloaded from~~ [~~https://www.mongodb.com/try/download/compass~~](https://www.mongodb.com/try/download/compass)

### ~~1.1.3/Create a database to configure le local node~~

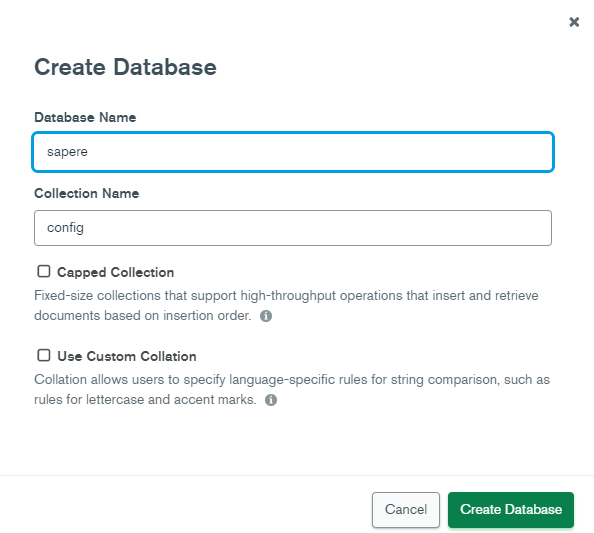
~~This step consists in creating a new base named "sapere" with the two collections "config" and "user" which are necessary for the node configuration.~~

* ~~Launch DBCompass~~

* ~~Connect to the local service (double-click on “localhost:27017”)~~

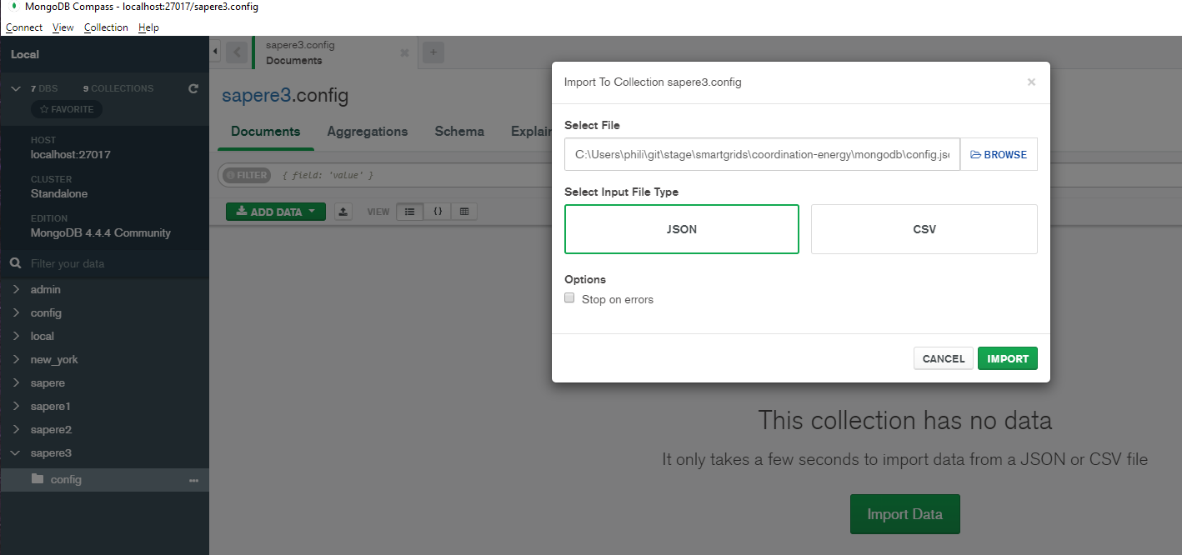


* ~~Create a new data base by using the “+” button~~
* ~~Enter a database name (For example “sapere”)~~
* ~~Enter a collection Name: “config”~~
* ~~Validate~~

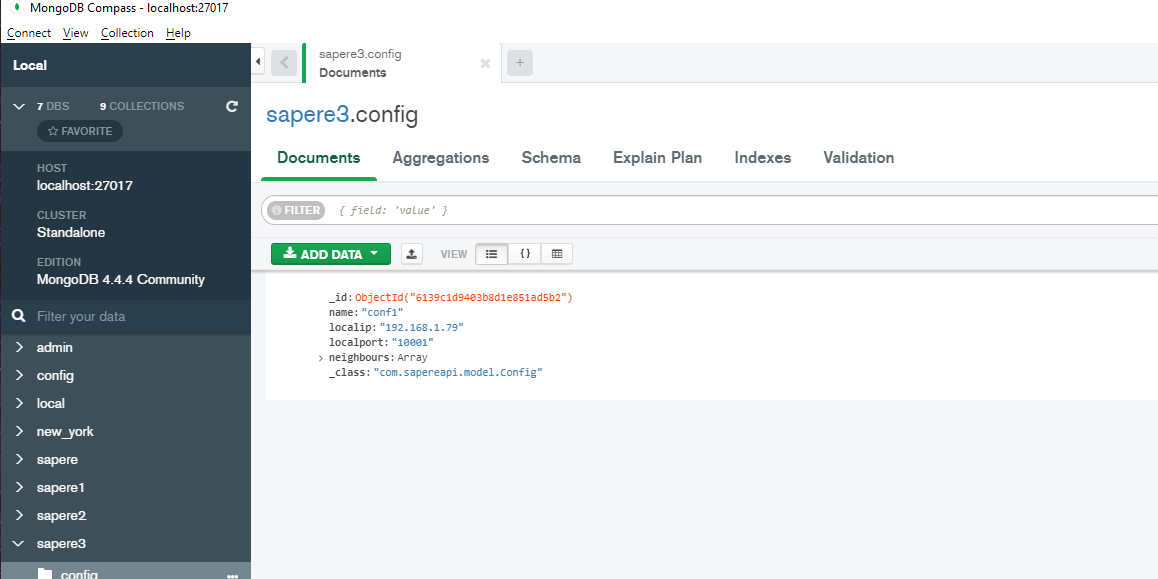


~~A database is created with an empty collection named “config”.~~

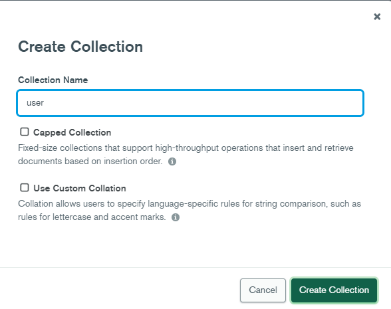
* ~~click on the “config” collection~~
* ~~use “import data” to populate this collection and select a node configuration file contained in mondb directory : for example, config.json~~



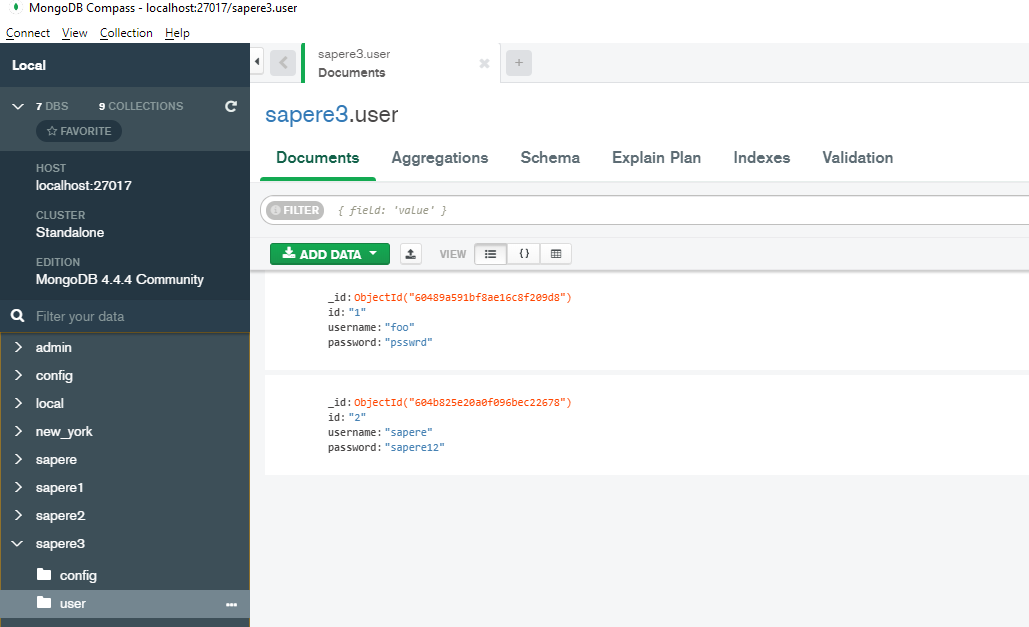
~~config collection after the import:~~



~~- Create a new collection names “user”~~

~~~~

~~populate this collection by using user.json file contained in the same directory~~



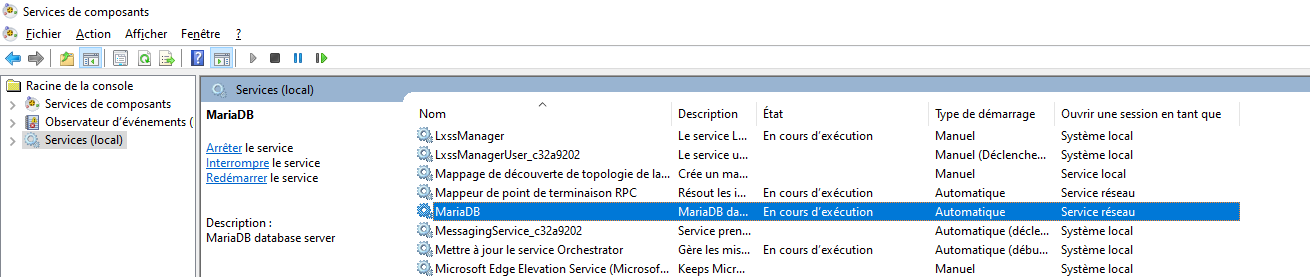
## 1.2/Agents database

Agents use a relational database to store event data, node state, and number of observations that generate Markov chain transition matrices.

### 1.2.1/Server installation

* Download le last version of MariaDB installer from: <https://mariadb.org/download/>
* Launch the installer file (for example for 10.6.4 version: mariadb-10.6.4-winx64.msi)
* During the installation process, MariaDB can be configured as a Windows service started automatically.

(windows service)

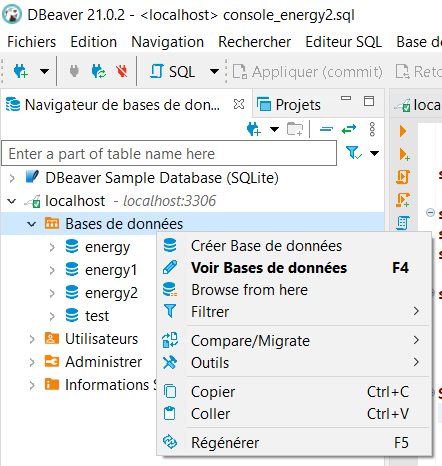


### 1.2.2/Client GUI installation

* Download the last version of DBEaver installer from: <https://dbeaver.io/download/>
* Install DBEaver (for example: dbeaver-ce-21.0.2-x86\_64-setup.exe

### 1.2.3/Database creation

* Launch DBeaver
* On the left frame, select “localhost:3306” server, right click and select the menu to create a database

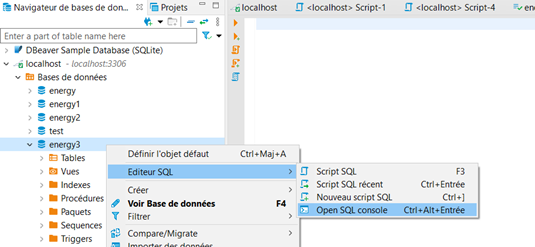


* Enter the database named (for example "energy"). The database name will have to set in the SpringBoot configuration file of SAPERE.
* keep the default charset and collation (utf8 / utf8\_general\_ci)

Graphical user interface, text, application

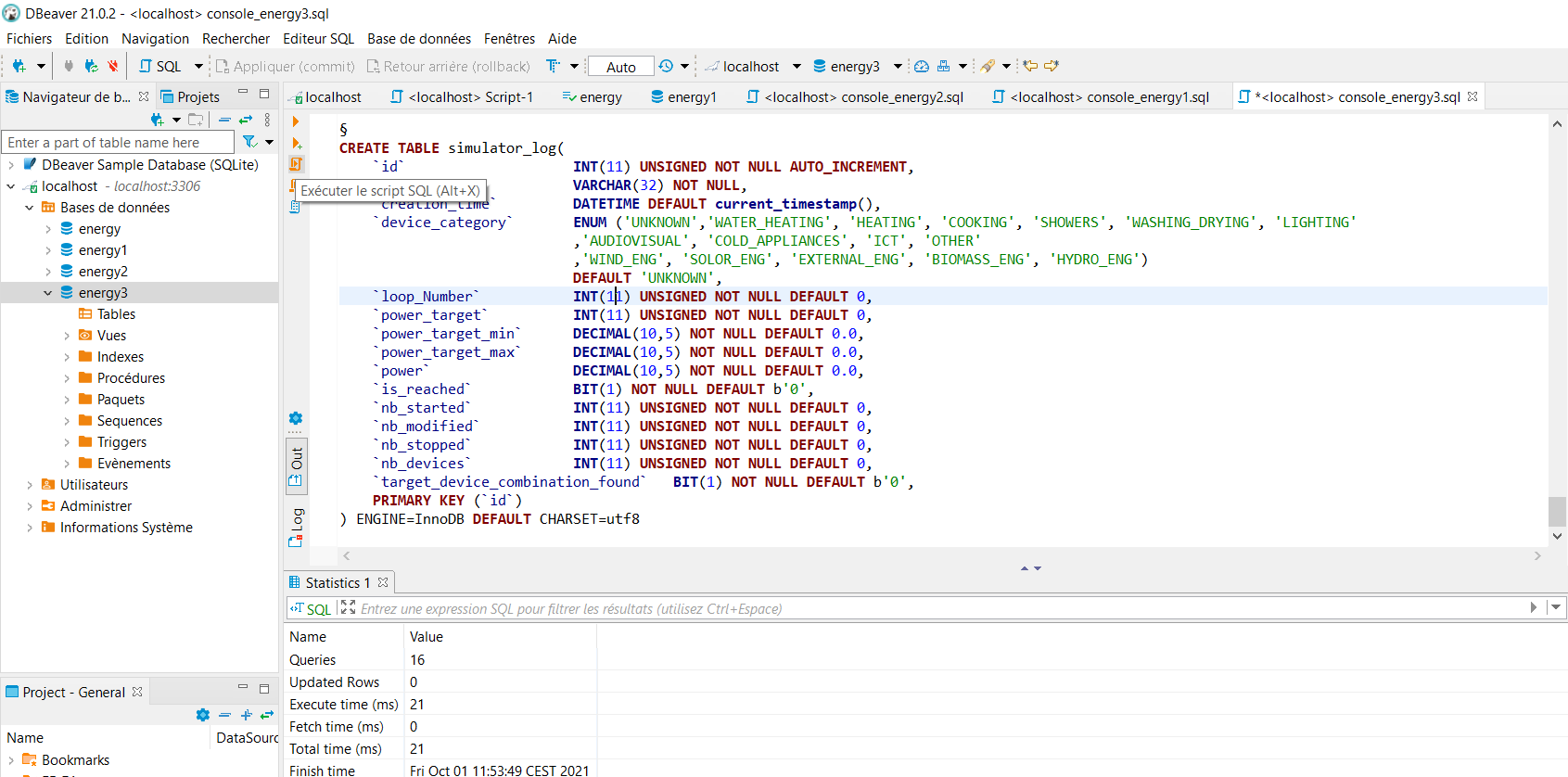
Description automatically generated

* Open a new SQL console: right clock on the new database, select “SQL Editor”-> “Open SQL console”



This console can be saved into a local file. This file is linked to the chosen database and contains the entered SQL requests. Yon can choose to store several console files linked to the same database.

* On this new database, launch the following scripts which are in the Mariadb sub-folder. For each script, copy entire the content to the console and execute it by using “Execute SQL script” button
  + 01\_create\_tables.sql: create the needed tables in the new database
  + 02\_procedures.sql: create the needed SQL function and store procedures
  + 03\_ref\_data.sql: populate the referential data
  + 04\_assign\_rights.sql: aassign rights on this basis to the “learning\_agent” account which is used by the different agents:



## 1.3/Web application:

### 1.3.1/ Installation of Visual Studio code

Visual Studio code is used as development environment for the web application.

* Download Visual Studio code from https://code.visualstudio.com/download
* Install Visual Studio Code
* Open the sapereangular folder from Visual Studio Code
* Launch a terminal from visual studio code (“Terminal” menu)

A screenshot of a computer

Description automatically generated

* On the new command window, go to the sapereangular directory

(use WMing64 console or linux commands)

* Launch the installation command: npm install --save-dev @angular-devkit/build-angular

### 1.3.2/ Install NodeJS

Download NodeJS V16 installer on <https://nodejs.org/en/blog/release/v16.20.0>

Launch the installer

Choose “add to PATH” option on the screen “Custom Setup”.

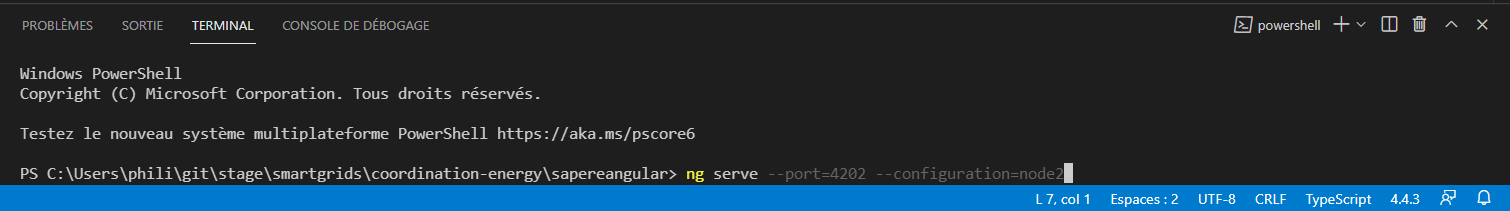
Launch npm install -g @angular/cli

### 1.3.2/ Starting the server

* Launch a terminal session
* move to sapereangular directory
* launch the starting command: ng serve --port=<port\_number> --configuration=<config\_name>

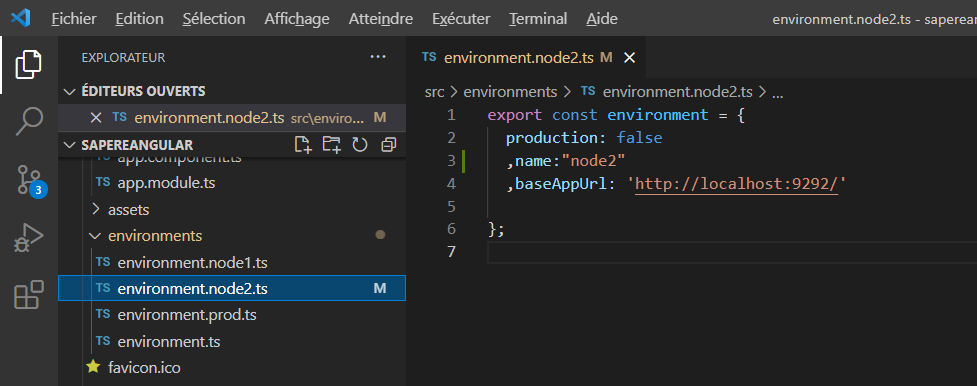
<port\_number> is the port number and <config\_name> is the configuration defined in the file named “environment. <config\_name >.ts” and located in “environments” subdirector.

**Example:**

****

By default,

* port number is 4200
* used configuration is the default configuration which is defined in “environments/environment.ts” file “node2” configuration is chosen (as in the example above), the server uses the configuration defined in “environment. node2.ts” file.



The “baseAppUrl” field defines the address of the SAPERE micro-service located at the same node.

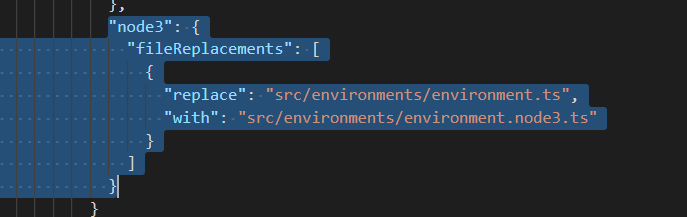
### 1.3.3/Adding a new configuration

If you need to add another node environment, you can add a new environment configuration

* enter the new configuration in a new file named “environments/environment. <config\_name>.ts”.
* update angular.json file :
  + insert the new configuration in “configurations” block:

add a “fileReplacements” item attached to the new configuration

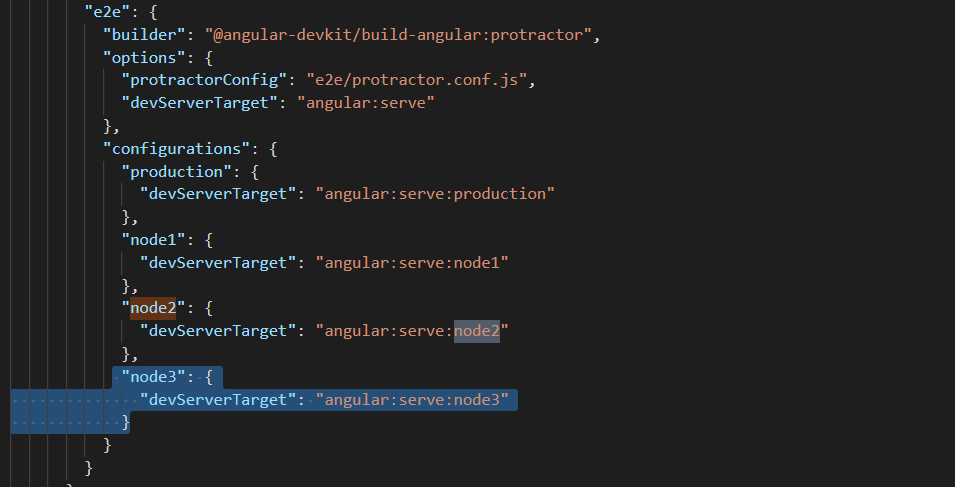
Example, for “node3” configuration:



* + insert the new configuration in “serve” -> “configuration” block



* + insert the new configuration in “e2e” -> “configuration” block:



## 1.4/ SAPERE micro-service

### 1.4.1/ Installation of Eclipse

The installer can be downloaded from:

https://www.eclipse.org/downloads/packages/release/kepler/sr1/eclipse-ide-java-developers

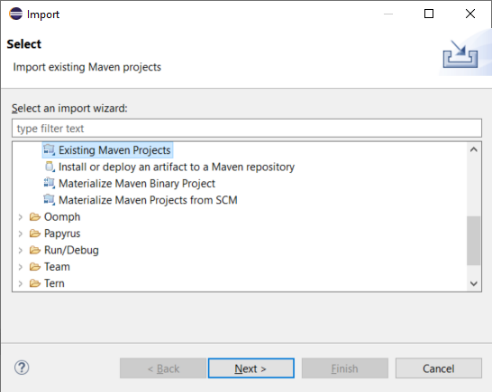
Also, you need to install java if you haven't already.

The last version can be downloaded from:

https://www.oracle.com/java/technologies/downloads/

### 1.4.2/Import of the SAPERE Spring-boot project

* File > Import project > Existing maven project



* In the file browser, select “coordination-energy” directory

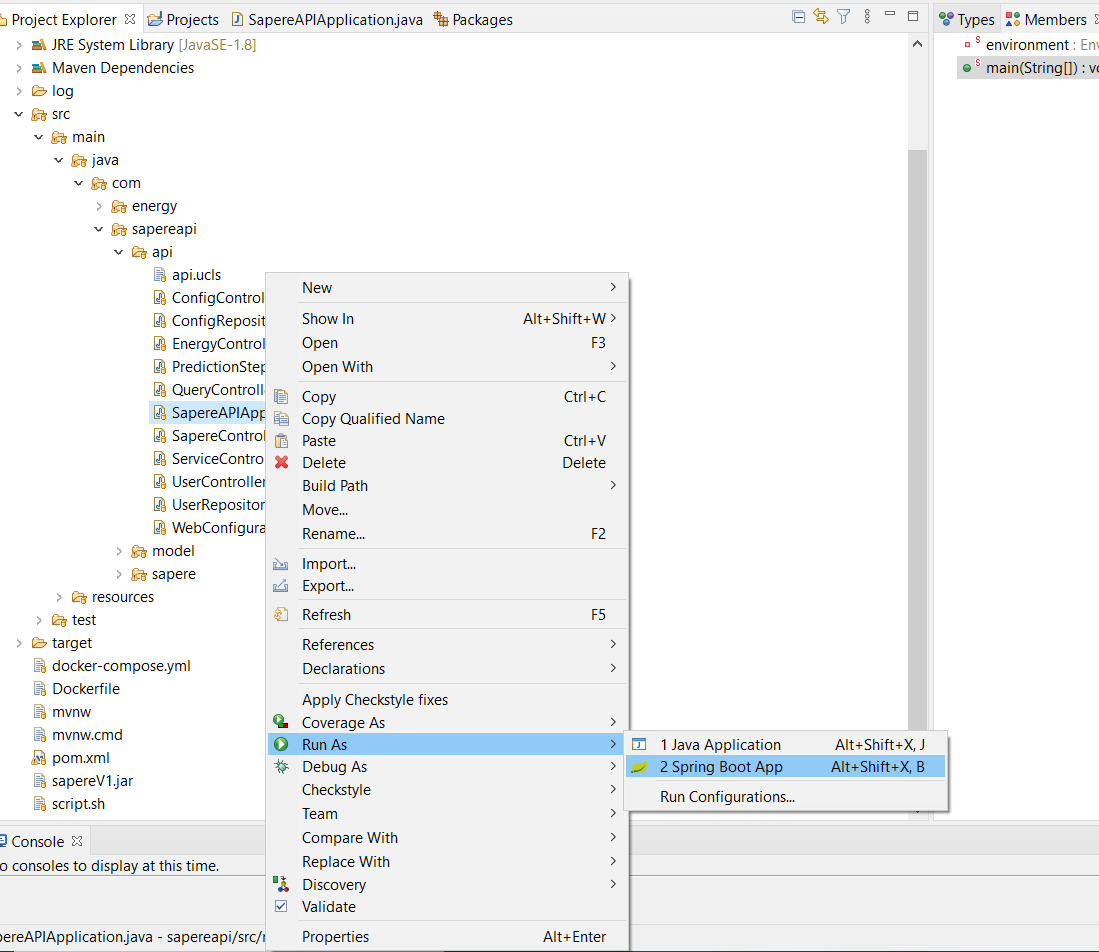
Graphical user interface, text, application, email

Description automatically generated

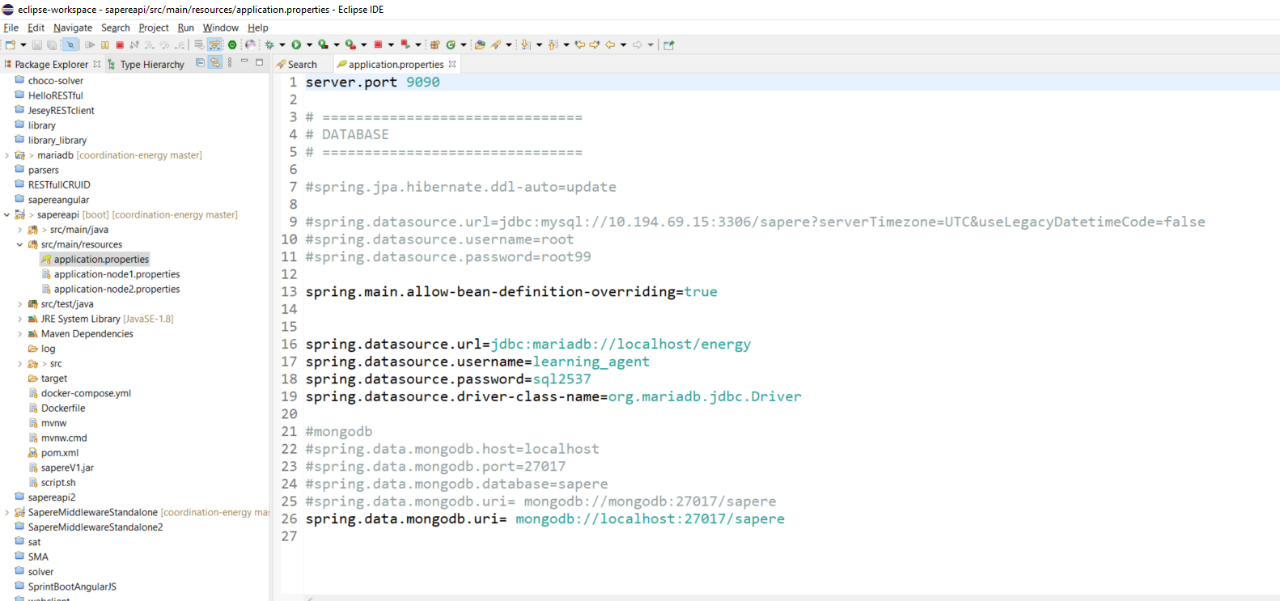
* Select sapereapi/pom.xml and click on “Finish”

### 1.4.2/Starting the SAPERE Spring boot application

* Select the SapereAPIApplication class
* Right click on the class name and select “Run AS” -> “**Spring boot app**”



The default configuration information is in the "application.properties" file . The default server port which is set to 9090



You can change the running configuration:

* Select the menu run -> run configuration
* Update the configuration name and the argument “Arguments”. The argument corresponds to the configuration name (“node3” for “application-node3.properties”)

Graphical user interface, text, application

Description automatically generated

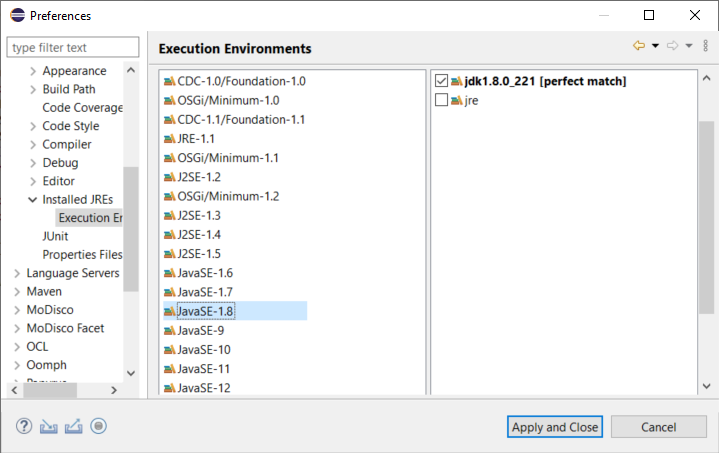
### 1.4.3/Regeneration of SapereV1.jar

SapereV1.jar is the core library of SAPERE

* From the Eclipse project SapereMiddleWareStandalone, export the jar: “right click“ -> ” export “ -> “jar file”
* on the right frame, deselect ".classpath", ".gitignore", ".project" and "SapereMiddlewareStandalone.iml" files because they are not necessary for the generation of this jar.
* on the left frame, deselect the "build" directory because it is not necessary for the generation of this jar.

Une image contenant texte

Description générée automatiquement



* Make sure the jar is properly updated in the sapearpi project
* Go to the command line in the sapereapi project
* Launch :

mvn install:install-file -Dfile=sapereV1.jar -DgroupId=com.sapere -DartifactId=sapere -Dversion=1.0 -Dpackaging=jar

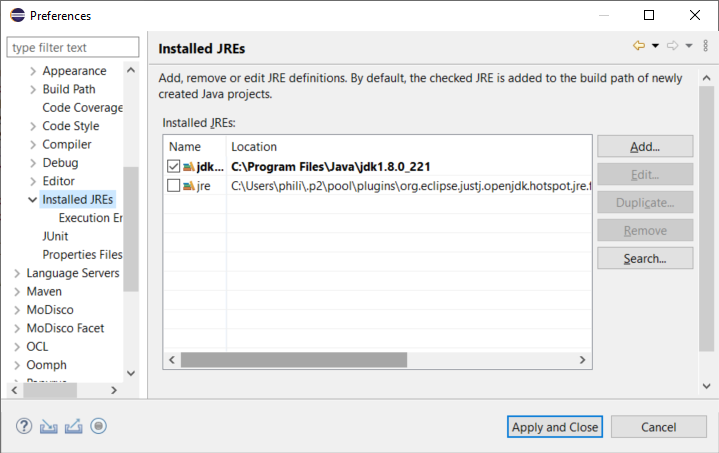
* Launch : mvn clean package
* Launch the maven task "update project" from the eclipse project sapereapi

### 1.4.4/JDK Problem resolution:

“[ERROR] No compiler is provided in this environment. Perhaps you are running on a JRE rather than a JDK” sur la commande maven « mvn clean package »

In Windows->Preference->Installed JREs

* Select the used jdk



* idem on Execution Environment -> JavaSE-1.8

- Make sure that the system environment variables point to the JDK

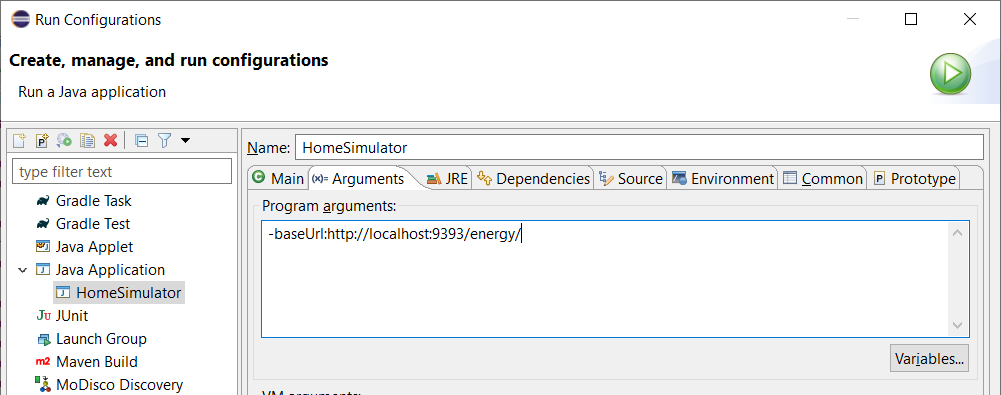
Path : must contains « C:\Program Files\Java\jdk1.8.0\_221\bin »

JAVA\_HOME : C:\Program Files\Java\jdk1.8.0\_221

## 1.5/Running the home simulator

Select run->run configuration->

* Enter the main class com.energy.test.HomeSimulator
* Add the micro-service http address in the parameters -baseUrl:http://localhost:<server\_port>/energy/



* Click on “apply” to save the configuration
* Click on Run

# 2/ Test with 2 node environments

|  |  |  |  |
| --- | --- | --- | --- |
| **Node** | **Start web application** | **Start test simulator** | **Mico-service start-up** |
| Node1 | ng serve –port=4201  –configuration=node1 | com.energy.test.HomeSimulator  -baseUrl:http://localhost:9191/energy/ | com.sapereapi.api.  SapereAPIApplication  node1 |
| Node2 | ng serve –port=4202  --configuration=node2 | com.energy.test.HomeSimulator  -baseUrl:http://localhost:9292/energy/ | com.sapereapi.api.  SapereAPIApplication  node2 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |