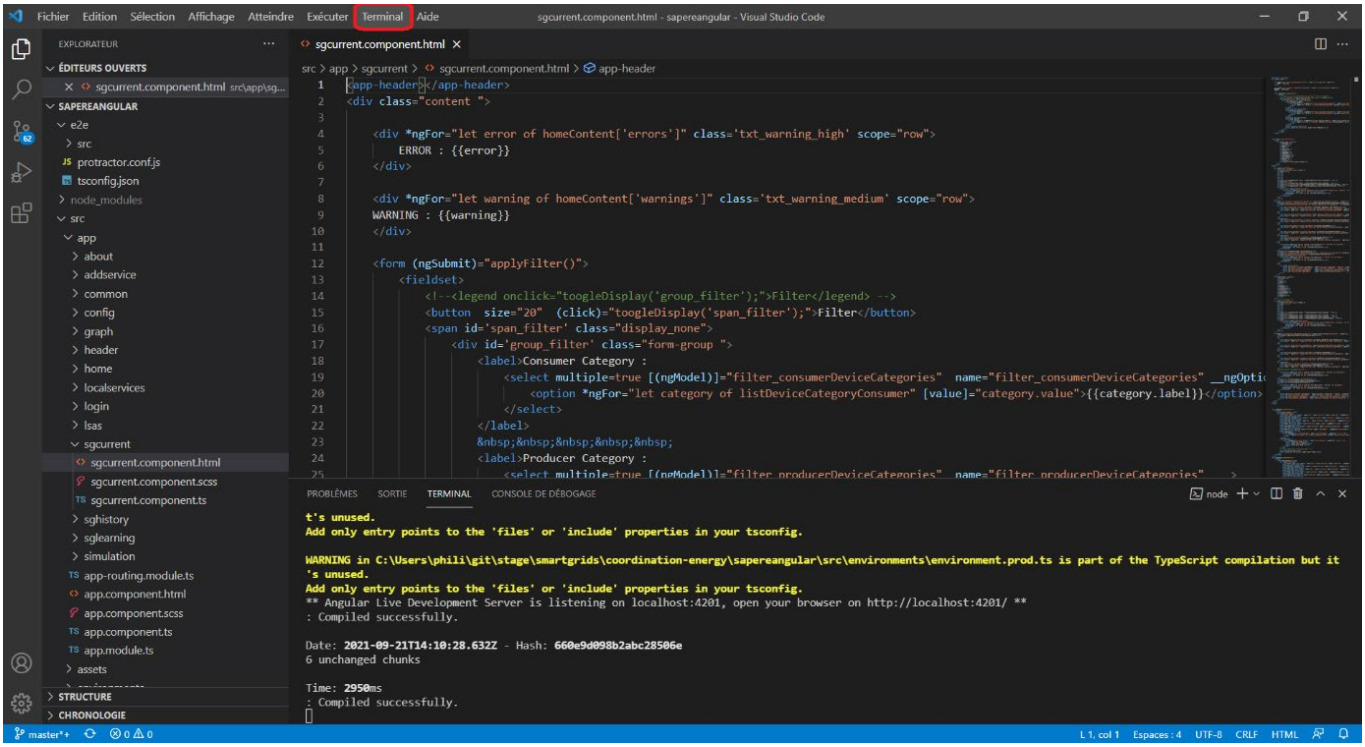
The source code is available on the Github repository: to retrieve it, run a gitclone command on the following repository address:
https://github.com/philippe-glass/energy/tree/master/sapereangular

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)



- On the new command window, go to the sapereangular directory:
- Launch the installation command: npm install --save-dev @angular-devkit/build-angular.

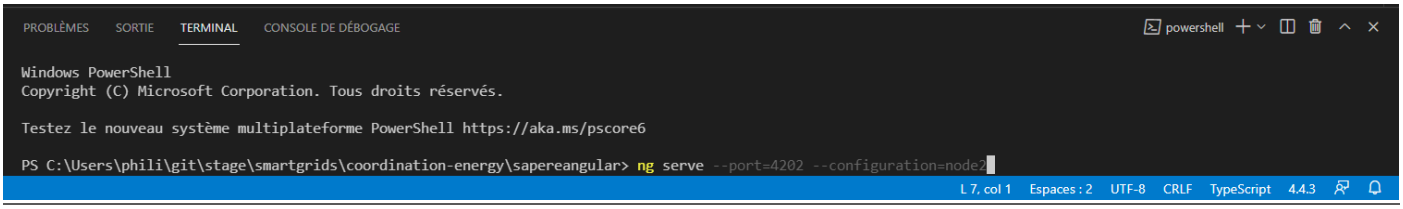
2/ Starting the server

- Launch a new terminal session (menu “terminal” > “new terminal”)
- Ensure that the session is located in ”sapereangular” root directory (it should be the default directory)
- From the session, launch the 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.

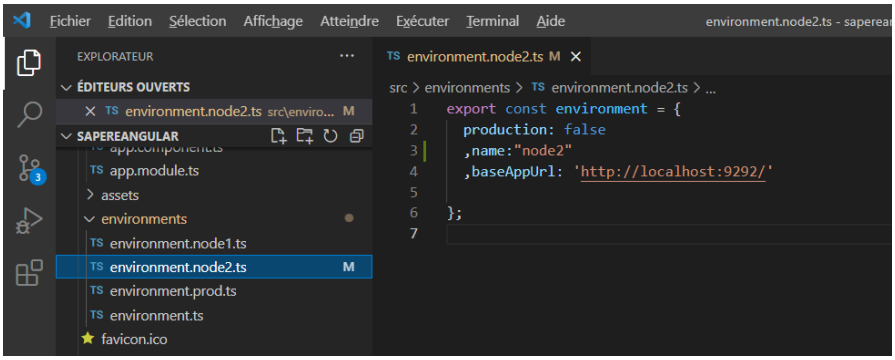
.

Example of launch command:



By default,

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



Note that the configuration file contains the 3 following fields:

- “production”: a Boolean which confirms (or not) if the environment is production.
- “name”: the environment name
- “baseAppUrl”: the SAPERE web-service route queried by the web application.

3/Adding a new configuration

If you need to add another node environment, simply complete the following procedure:

- enter the new configuration in a new file named “environments/environment. <config_name>.ts”. This file must follow the format described above (cf. “environment. node2.ts” screen capture)
- 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:

```
"serve": {
  "builder": "@angular-devkit/build-angular:dev-server",
  "options": {
    "browserTarget": "angular:build"
  },
  "configurations": {
    "production": {
      "browserTarget": "angular:build:production"
    },
    "node1": {
      "browserTarget": "angular:build:node1"
    },
    "node2": {
      "browserTarget": "angular:build:node2"
    },
    "node3": {
      "browserTarget": "angular:build:node3"
    }
```

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

```
"e2e": {
  "builder": "@angular-devkit/build-angular:protractor",
  "options": {
    "protractorConfig": "e2e/protractor.conf.js",
    "devServerTarget": "angular:serve"
  },
  "configurations": {
    "production": {
      "devServerTarget": "angular:serve:production"
    },
    "node1": {
      "devServerTarget": "angular:serve:node1"
    },
    "node2": {
      "devServerTarget": "angular:serve:node2"
    },
    "node3": {
      "devServerTarget": "angular:serve:node3"
    }
  }
}
```

- Test the new configuration:

  ng serve --port=<port_number> --configuration=<config_name>