HSE Setup and Usage Guide

Robert Jans

December 19, 2020

Contents

1 (Temporary) Source Download and Running Application

The source codes and related documentation files are available on *GitHub* under

https://github.com/robix82/bsc_project/.

The repository can be cloned by issuing the following command

`git clone https://github.com/robix82/bsc_project.git`

For testing the application and its interactions with Qualtrics, I temporarily deployed it on

http://www.robix-projects.org/hse.

You can log in using the following credentials;

- username: admin

- password: admin

2 Technologies used

The project is constructed as a web application using the *SpringBoot* framework. It consists in a back-end written in java, HTMl pages (served via the *Thymeleaf* templating engine), some (JavaScript) to be run on client side, and *css* files for the user interface styling. Data is stored in part in a *MySql* database and in part as text files on the server's file system. The *JQuery* and *Bootstrap* frameworks are used for keeping the front-end code as simple as possible.

Dependency management and build configurations are handled by the *Maven* project management tool. In order to allow a simple deployment procedure, the application is packaged into a *Docker* image at build time. The application can be started and stopped using *Docker Compose* which automatically downloads, initializes, and links the required *MySql* database. For a detailed deployment description, see section 3. For inspecting and/or modifying the source code, I suggest using the *Eclipse* IDE.

3 Configuration, Build and Deployment

3.1 Dependencies

The system on which you build the application must have the following software installed:

- Java JDK 11

- Apache Maven 3.6.3

- Docker version 20.10.1

If you need to run the application locally without using docker, also *MySql* is required.
The server on which the application is to be deployed, only needs *Docker* and *Docker Compose*.

3.2 Configuration Files

3.2.1 Maven configuration: pom.xml

The build settings used by *Maven* are defined in `/hse/pom.xm`. This file contains some general information about the project, such as name and version, as well as a list of java packages and *Maven* plugins which are downloaded and set up at build time. The File also declares two profiles ("dev" and "prod"). The "dev" profile is intended for creating a local build to be used during development, while the "prod" profile is to be used for building the deployment version. The profiles are linked to specific configuration files

which contain various settings such as server ports and global constants: when the profile "dev" is selected, the application uses the file `/hse/src/main/resources/application-dev.properties`; when "prod" is selected, `/hse/src/main/resources/application-prod.properties`.

By default the "dev" profile is selected; for using the "prod" profile, the flag `-Pprod` is to be included in the *Maven* command (e.g. `mvn -Pprod clean install`).

3.2.2 Specific configurations in `.properties` files

The directory `/hse/src/main/resources/` contains three files with extension `.properties`: `application.properties`, `application-dev.properties` and `application-prod.properties`. The first one contains settings that are applied independently of the selected profile, while the other two contain profile-specific settings. The crucial settings to be considered at build time are the `spring.datasource.xxx` properties, which indicates the database which the application is going to connect to, and the `baseUrl` parameter, which indicates the prefix used at server level. For instance in the `application-prod.properties` as I have set it up, the data source is pointing to a *MySql* Docker container, and the base url is set to `/hse/`, since I deploy it on `http://www.robix-projects.org/hse/`. In the `application-dev.properties` file the data source points to a local *MySql* instance and the baseUrl is `/`.

The other properties indicate directory paths and should not need to be modified.

3.2.3 docker-compose.yml

The simplest way to run the application on a server is by using *Docker Compose*. The way in which the containers are created from the images and the internal ports used are defined in `/hse/docker-compose.yml`.

3.3 Creating a local build

3.3.1 Preparing the database

In order to run the application locally , a *MySql* database service running on port 3306 is required. The service must contain a database named `hse_db` and should be accessible via username `root` and password `root`. The tables are created automatically at application startup. If you need to use other login credentials, or the service is running on another port, these parameters can be set in `application-dev.properties`.

3.3.2 Issuing the build command

The command

```
mvn clean install
```

initiates the build process. The process involves executing several test suites, which should work without failure. In case the tests fail (e.g. due to path incompatibilities or missing files) the tests can be skipped using the `-DskipTests` flag:

```
mvn -DskipTests clean install
```

3.3.3 Running the application

Once the application is built, it can be run in several ways. During development it is convenient to run it from the IDE (in *Eclipse* package explorer, right-click on project → Run As → Spring Boot App). Alternatives are to run it from command line using *Maven*:

```
cd hse/
mvn spring-boot:run
```

or using *Java*:

```
cd hse/target/
java -jar hse-0.1.jar
```