Documentation

Users Management

Dimitar Avramov

September 17, 2020

I - Introduction

The main idea of this project is building a small web application that manages user accounts in database. For this purpose there are several functionalities implemented. Separated in five main pages, each with some specific logic.

- Register

  Part where you can sign up for the web site. There is a form in which the required inputs are:
  - First name
  - Last name
  - Username
  - Email address
  - Birthdate
  - Password

There is some validation from font-end and back-end for example: minimum length, maximum length, confirm password, uniqueness of username etc. The password for every user is encrypted and not available for all kind of users in the system (Database administrators, admins on the site etc.). When everything is correct the account is registered successfully and saved in the data base.

- Login

  Login form is based on authentication using your username and password.

- All Users (available from role: USER)

Section in which are displayed all the accounts available in the web site. Data is shown in sortable table for every column (first name, last name, username, email, birthdate). There is an option for searching specific string which shows you filtered results gathered from every column. On this page also you can perform some other functionalities like:

- Create – add new account (available from role: USER)
- Edit – change account data (available from role: MODERATOR)
- Delete – Delete account from the database (available from role: ADMIN)

*Note that there are mentioned some roles which we will explain in the next chapter.

- Logs (available from role: MODERATOR)

  In this part you accounts with minimum role moderator can see the history of changes from any client of the website. With this feature you can make statistics on things like who is the most active user, at what time the application is most used, which section is the most visited etc. On the other side there is a possibility to track which of the moderators or administrators did wrong things like delete, update, add and protect from unexpected or unwanted activities

- Organization(available from role: ADMIN)

Section in which with simply user-friendly table are listed again the users in a short way. But,

here comes some other idea to change the authorization process easily. Changing the roles is realized with a simple way using two buttons: promote and demote. Note that, here you cannot change the role of the owner of the web application(ROOT-ADMIN) and also your personal role.

Storing the information data in database is secured with the implementation that the database works on port which is available only on localhost.

II – Authorization

Authorization process is realized with a role hierarchy logic. There are four types of users each of them extending the role below. Using that way we can provide the functionalities assigned for the particular role. On the top of authorization lays the owner of the application who possess all the rights and cannot be removed or edited for the purpose of not getting unwanted activities.
The four main roles are:

- ROOT-ADMIN
- ADMIN
- MODERATOR
- ROLE_USER

III – Deploying and building

Open the folder back-end with an application which supports Spring Framework, for example IntelliJ IDEA. Then, using Maven you can install all of the dependencies defined in the file <u>pom.xml</u>. Here is defined information about the artifacts, their version, scope, name etc.

For storing data, the project is using MySQL database. In <u>application.properties</u> file is described the configuration of the database. With the next line is shown the port on which the database is running.

```
spring.datasource.url =
jdbc:mysql://localhost:3306/westernacher_db?useUnicode=true&characterEncoding=utf8&useSSL=true&useLegacyDatetimeCode=false&serverTimezone=Europe/Sofia&createDatabaseIfNotExist=true
```

Also, here we can see the database table name which in this case is **westernacher_db** and some other configuration that the table is created automatically if does not exists**.**

On the next line you should specify your database username and password

```
spring.datasource.username =
spring.datasource.password =
```

Note that in the project is used my own username password, you should change that information.

```
#Server Properties
server.port=8000
```

The upper line shows the port on which the server will be running. You can change it and point to some another port.

When you are done with the set up of the dependencies and the database, you can build the application. For that purpose you should have the following system prerequisites installed:

- Java 11
- Database: MySQL
- Application server: Apache Tomcat 9

After building successfully you can run the application. Application server is running on localhost:8000. The next step is to open the front-end folder from some application, for example WebStorm. From there you should open index.html file via some browser which is the starting point of the REST application. Then you have started the project and you are guest in the site. By now, to access some resources of the web application you should have an account. In the future, project can implement parts which are visible from 'world'. The implementation to assign roles to the users is done with the idea that the first registered user is the owner of the application and has the authority of ROOT-ADMIN. All of the next registered accounts are assigned to the basic role of the user. So, when you register your first account you are the owner and you can manage all of the accounts with full privileges.