

vf-advanced-java-development

Project: Make an E-commerce Website for Sporty Shoes

Project objective:

As a Full Stack Developer, complete the features of the application by planning the development and pushing the source code to the GitHub repository.

Background of the problem statement:

Sporty Shoes is a company that manufactures and sells sports shoes. They have a walk-in store, and now, they wish to launch their e-commerce portal sportyshoes.com. You're asked to develop a prototype of the application. It will be then presented to the relevant stakeholders for budget approval. Your manager has set up a meeting where you're asked to do the following:

- Presenting the specification document which has the product's capabilities, appearance, and user interactions
- Setting up Git and GitHub account to store and track your enhancements of the prototype
- Explaining the Java concepts used in the project
- Discussing the generic features of the product:
- There will be an admin to manage the website. An administrator login will be required to access the admin page.

The admin should be able to change his password if he wants, he should be able to:

- Manage the products in the store including categorizing them
- Browse the list of users who have signed up and be able to search users
- See purchase reports filtered by date and category

Prerequisites

Start IntelliJ IDE project - spring initializr

To create a Springboot project in the IDE, go to the page <https://start.spring.io/> to assemble the necessary components.

Choose:

- Project = Maven
- Language = Java
- Spring Boot = 3.4.2
- Project Metadata = Name ...
- Packaging = Jar
- Java = 17

The following dependencies are needed:

1. Spring Web
2. Spring Boot Dev Tools
3. Spring Data JPA
4. MySQL Driver

The image shows the Spring Initializr web interface. On the left, there's a sidebar with a hamburger menu and a clock icon. The main area is divided into two columns. The left column contains 'Project' settings (Language: Java, Maven; Spring Boot version: 3.4.2), 'Project Metadata' (Group: com.example, Artifact: EcommerceProject, Name: EcommerceProject, Description: course end project for Spring Boot, Package name: com.example.EcommerceProject, Packaging: Jar, Java version: 17), and 'Dependencies' (Spring Web, Spring Boot DevTools, Spring Data JPA, MySQL Driver). The right column contains a 'GENERATE' button (CTRL + G) and an 'EXPLORE' button (CTRL + SPACE).

click **Generate** and download.

Start IntelliJ IDE project - create project in IntelliJ IDE

To create a Springboot project in the IDE the following steps has to proceed:

1. Unzip the .zip download and copy the package into the project directory ~/.
2. open the IDE and click in the Menu/Open/project directory/'project' --> new project is going to be created
3. choose in "Project Settings" the needed Software Development Kit (SDK): **corretto-17** (if not available; download)

For test purposes with JUnit 5 we need to add JUnit as a dependency

1. Open pom.xml in the root directory of your project.
2. In pom.xml, press AltInsert and select Dependency.

3. In the dialog that opens, type org.junit.jupiter:junit-jupiter in the search field. Locate the necessary dependency in the search results and click Add.
4. When the dependency is added to pom.xml, click Reimport All Maven Projects in the Maven tool window to import the changes.

Start MySQL - database as a docker container

To start the database as a docker container, run the following command:

```
docker run \  
  --name mysql-db \  
  --env MYSQL_ROOT_PASSWORD=mypasswd \  
  --env MYSQL_DATABASE=JavaProject \  
  --env MYSQL_USER=Karsten \  
  --env MYSQL_PASSWORD=12345 \  
  --detach \  
  --publish 3307:3306 \  
mysql
```

Define the application properties - ServerPort, Database, JPA

open application.properties, copy the following properties:
spring.application.name=Webdb

#if you need to change the port of the tomcat the do the below step
server.port=8089

```
#MySQL database  
spring.datasource.driverClassName=com.mysql.jdbc.Driver  
spring.datasource.url=jdbc:mysql://localhost:3306/JavaProject  
spring.datasource.username=root  
spring.datasource.password=123456
```

```
#jpa-hibernate
spring.jpa.database-platform=org.hibernate.dialect.MySQLDialect
#converts the pojo into the DDL formatted table
#create- if the table is exiting or not existing this command deletes the table and
creates a new table
# update - if table exists it doesnt delete the table / if table doesnt exists it
creates it
spring.jpa.hibernate.ddl-auto=update
```

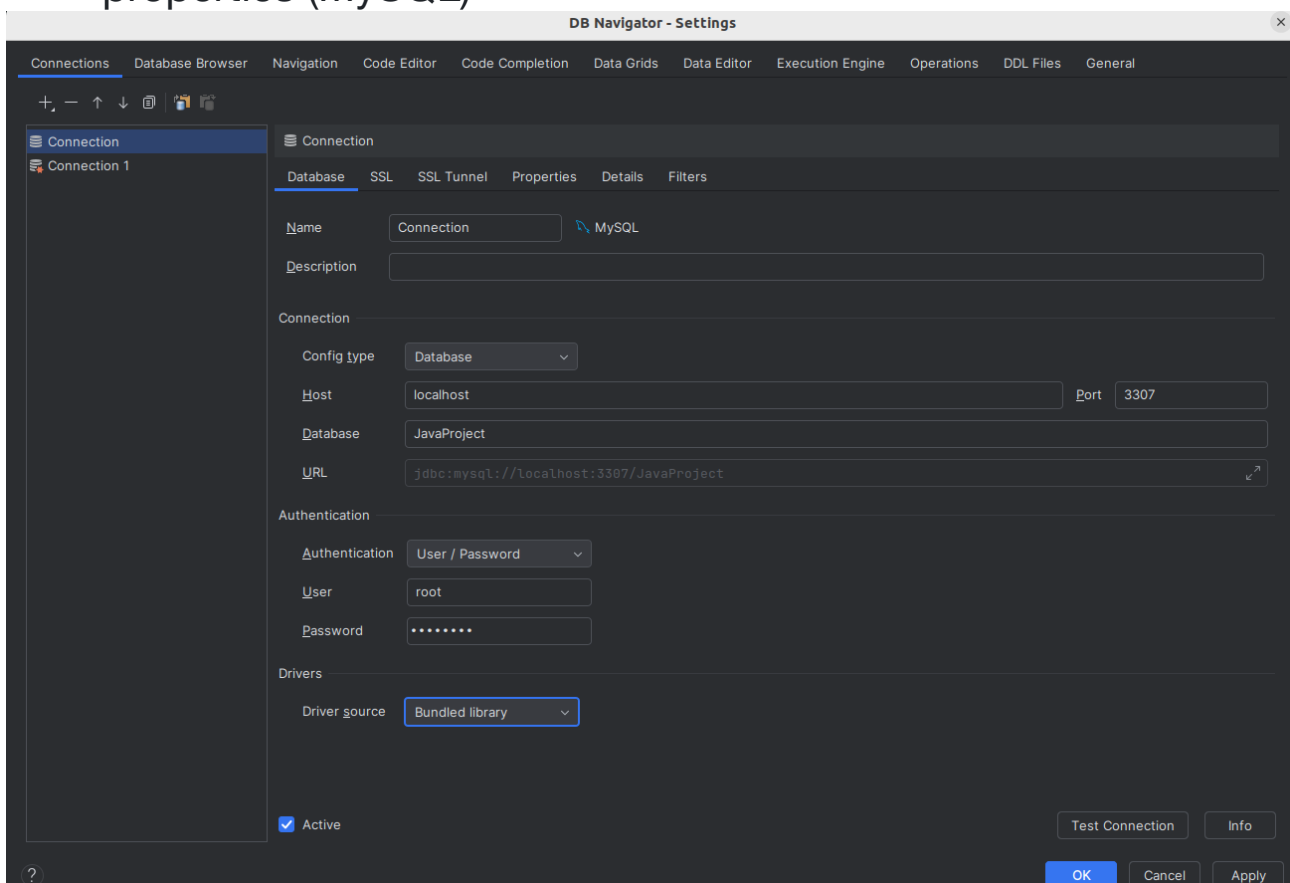
```
image.path=/tmp/images/
```

```
#sql on my server
spring.jpa.show-sql=true
```

Connect MySQL database with project

precondition is an installed database plugin (e.g. "DataBaseManager")

1. Open the DB Browser
2. Click "New Collection" and choose "MySQL"
3. Fill in the necessary information analog the lines of application properties (MySQL)



4. Test connection and if working click "Apply"

Create mandatory user roles:

The following users roles are mandatory and requires:

1. Administrator
2. User

After the backend server started, execute the following commands:

```
curl -X POST http://localhost:8089/roles --header "Content-Type: application/json" --data '{"role": "Administrator"}'  
curl -X POST http://localhost:8089/roles --header "Content-Type: application/json" --data '{"role": "User"}'
```

One initial Admin User needs to be created:

```
curl -X POST http://localhost:8089/users/add --header "Content-Type: application/json" --data '{"name": "Admin", "password": "admin", "role": {"role": 1}}'
```

React + Vite

This template provides a minimal setup to get React working in Vite with HMR and some ESLint rules.

Currently, two official plugins are available:

- [@vitejs/plugin-react](#) uses [Babel](#) for Fast Refresh
- [@vitejs/plugin-react-swc](#) uses [SWC](#) for Fast Refresh
- Installation: `npm install`
- Start: `npm run dev`