Project report

for

Libraria

Library Management System

CTU FEE
B6B33EAR
Winter 2018
Seminar 106

Jan Hlaváč, Jan Macek

2018-12-30

1. Application description

Libraria is an Enterprise Application for a managing of a small-town library. Two roles of users operate within the application: Librarian (admin) and Reader (plain user).

2. Project Structure

Libraria is a Maven compilable project written in Java. It stores the application data on predefined database, while the application itself can run on remote server that is accessible e.g. from web browser via REST API.

3. Installation guide

To launch the application, you need following tools:

- Java 8 or newer + Maven
- Apache Tomcat 9.0.12 (version used for developing the app)

3.1 Launching the project

1. Download the project from `https://gitlab.fel.cvut.cz/B181_B6B33EAR/hlavaj28`
2. Open in the unzipped project in IntelliJ Idea
3. Check if the WAR artifacts in Project Structure are present (`File > Project Structure > Artifacts`)
    a. If they are not present, create both exploded and non-exploded WAR artifact.
4. Check for Tomcat Server on the left in `Run > Edit Configurations`.
    a. If it is not present, add `Tomcat Server > Local`
    b. Connect your local Tomcat directory and name the Configuration.
    c. Under Deployment tab, add the non-exploded WAR artifact and add `/libraria` to its context
    d. Under Server tab the URL should contain `http://localhost:8080/libraria`
    e. Apply
5. Run the configuration you just created. The Tomcat Server should launch and the database connection should be established.
6. The application is deployed and running on the local server and can be accessed through REST API via HTTP requests.

Other IDEs would have similar procedure.

4. Team members project evaluation

The biggest problem was during attempts to deploy to Tomcat Server, because the deployment process did not launch AppContext, which resulted in 404 error code at deployment URL. This was solved by creating a WebAppInitializer class that manages the Application Context and allows us to deploy to Tomcat Server.

We chose plain Spring within the deployment process as opposed to SpringBoot.