

Project AWT

Robin Schmid

14.01.2016

1 Initial Position

2 Technologies

To create this project, I used several technologies. I will shortly present them in the following sections.

2.1 Server

The application runs on a Glassfish 4.0 Server. The reason I chose Glassfish over Tomcat or others is easy to explain. First I started my Project, but as the structure grew, I had to look out for some more advanced technologies. I then read of CDI (Contexts and Dependency Injection). This is the new standard since Java EE6 and to support Java EE, I had to look out for something else then Tomcat.

The database is an MySQL database on a simple server provided by XAMPP (locally).

2.2 Backend

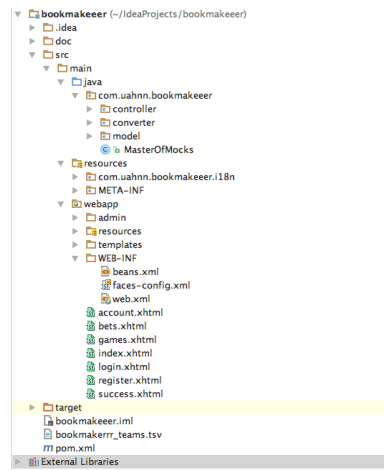
For the core of the project I used JSF 2.2. All the necessary dependencies are managed with Maven. Maven is a powerful tool that provides an easy way to find the correct JARS without downloading them manually. One of this dependencies is the JPA (Java Persitence API). I have never work on a Java project with a database behind. So I had to get some information about it. I heard that Hibernate was a good ORM Mapper. I tried it out, but as my project became more and more a EE project, I put down all my effort on Hibernate and chose the reference implementation, EclipseLink.

2.3 Frontend

For the frontend I completely resigned the use of JSP. I only used Facelets. As I normally work as a frontend developer in my company, I could easily think of some nice extra tools for the frontend. For the most important part, the structure of the pages, I decided to use the Bootstrap framework. To style the pages in my favor, I used the one and only CSS. In stead of writing CSS code directly, I wrote my stylings in SASS. Then with Gulp I builded a minified CSS. With the help of Gulp I could also directly inject dependent

style files, for example the already named Bootstrap or Font Awesome for some nice icons. For the dependency management in this part, I used Bower.

3 Project Structure



4 Components

5 Conclusion