

# Software Engineering 2

## Lösungen der Übung

Dr. F.-K. Koschnick, Sybit GmbH

# Aufgabe (Story): Projektseite für RetroWeb bauen

## Teil 1

Als User von RetroWeb möchte ich auf einer Webseite die Details eines Projekts mit den durchgeführten und geplanten Retros sehen, um mich über das Projekt, dessen Retros und über die Termine der geplanten Retros zu informieren.

Teil 1: Auf der Webseite soll nur der Projektname als Überschrift stehen, die Retros lassen wir in dieser Übung außen vor.

Verifiziere, dass

- sich die Projektdetailseite öffnet, wenn der User auf ein Projekt in der Home-Page klickt
- als Überschrift auf der Seite der Projektname steht
- es einen Link zurück auf die Home-Page gibt
- Die Seite den Header und Footer hat, der auf der Login-Seite schon besteht.
- nur ein eingeloggter User die Seite erreichen kann
- das Design der Seite mit bootstrap-Klassen gemacht wird

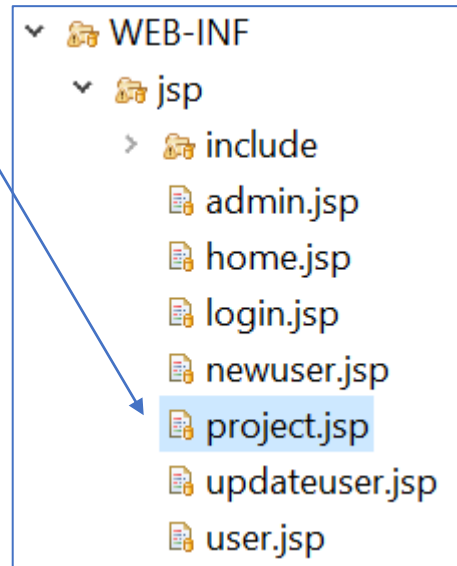
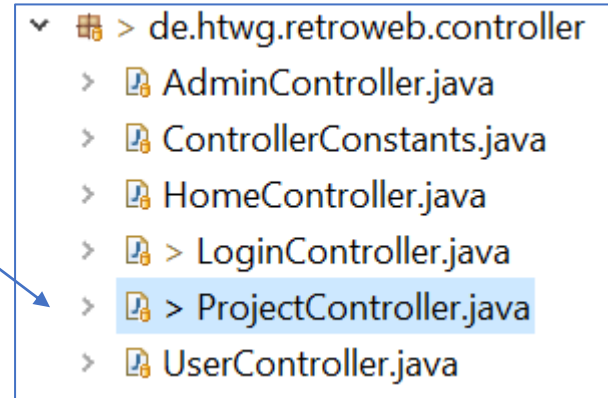
Technische Anmerkungen:

Es ist dabei die bestehende Spring-Boot-Applikation retroweb weiterzuentwickeln. Header und Footer sollen eingebunden werden (von der Aufgabe Design der JSP der Login-Seite von retroweb). Der View der Projektseite soll `project.jsp` heißen, der zugehörige Controller soll `ProjectController.java` heißen. Es soll der bestehende `ProjectService` genutzt werden. Dieser bietet für diesen Zweck die Methode *public Project getById(Long id)* an, wobei der Parameter `id` die `id` des Projekts ist, das der Service zurückliefern soll.

# Reihenfolge des Codings

1. Controller: ProjectController

2. JSP: project.jsp



# 1. ProjectController (Controller)

```
package de.htwg.retroweb.controller;

import java.util.Map;

@Controller
public class ProjectController {

    private static final org.slf4j.Logger LOG = LoggerFactory.getLogger(ProjectController.class);

    @Autowired
    ProjectService projectService;
    @Autowired
    SessionService sessionService;

    @RequestMapping(value = "/project", method = RequestMethod.GET)
    public String showProject(@RequestParam long id, Map<String, Object> model, HttpServletRequest request) {
        LOG.debug("--> project");
        HttpSession session = request.getSession();
        String page = ControllerConstants.PROJECT;

        if (sessionService.isLoggedIn(session)) {
            try {
                Project project = projectService.getById(id);
                model.put("project", project);
            } catch (ResourceNotFoundException e) {
                LOG.error("Project not found", e);
                model.put("errorMsg", "Project not found!");
            }
            model.put("userName", sessionService.getUserName(session));
            LOG.debug("<-- project");
        } else {
            LOG.debug("<-- login");
            page = ControllerConstants.REDIRECT_LOGIN;
        }
        return page;
    }
}
```

**WICHTIG:** den ProjectService innerhalb der if-Abfrage laufen lassen, weil sonst Security-Issue!!!

**model.put(String key, Object value)**  
Model wird mit Daten gefüllt. Es gibt also in der Model-Map jetzt die Keys: project und userName, mit denen im View (JSP) auf die Daten zugegriffen werden kann.

Unbedingt die Exception abfangen, damit kein 500er nach draußen geht.  
**model.put(String key, Object value)**  
Falls das Projekt nicht gefunden werden kann, gibt es eine Fehlermeldung.

## 2. project.jsp (View)

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html>
<html>
  <c:import url="include/head.jsp"/>
  <body>
    <c:import url="include/header.jsp"/>

    <nav class="navbar navbar-expand-lg navbar-light bg-light">
      <ul class="navbar-nav mr-auto">
        <li class="nav-item active">
          <a class="nav-link" href="home">Home</a>
        </li>
      </ul>
    </nav>

    <div class="page">
      <c:choose>
        <c:when test = "${errorMsg != null}">
          <div class="error-msg">${errorMsg}</div>
        </c:when>
        <c:otherwise>
          <h5>Project: ${project.name}</h5>
        </c:otherwise>
      </c:choose>
    </div>
    <c:import url="include/footer.jsp"/>
  </body>
</html>
```

Prüfen, ob Key errorMsg gesetzt ist

Zugriff auf Model mit dem Key *project* und dem Projekt-Attribut name.

# 3. header.jsp (View)

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<header>

<h1 class="header-title">RetroWeb</h1>
<c:if test="${userName!=null}">
<div class="header-username">You are logged in as: ${userName}</div>
</c:if>
</header>
```

Zugriff auf Model mit dem Key *userName*.