

Seminar #3

Miroslav Blaško

1 Goal

You will get experience with Spring and its testing.

2 Getting Ready

- Ensure you have available the software stack installed during the first lab (PostgreSQL server, PgAdmin, Netbeans 8.2).¹
- Clone the GIT repo <https://gitlab.fel.cvut.cz/ear/reporting-tool-seminars>, switch to the branch `WS2017-seminar-3-problem`².
- Open the project in Netbeans 8.2 and resolve any warnings/errors in configuration reported by Netbeans.³

At the end of the current week, the branch `WS2017-seminar-3-solution` becomes available where you can check your solutions.

3 Familiarization with the Project

The project is a small and JPA-ized excerpt of a bigger one developed for safety management in the czech aviation industry. The system allows to create safety reports and classify them. The system is based on JPA, Spring, and ReactJS. First, clean and build the project.

Before proceeding, go through the structure of `pom.xml`, Java sources at `src/main/java` and resources at `src/main/resource`. Become familiar with java classes `SecurityUtils` and `AbstractRepositoryService`.

¹Due to compatibility issues JDK 9 cannot be used to run reporting tool project.

²Run `git checkout WS2017-seminar-3-problem`

³In case you are using IntelliJ Idea, you can install Lombok Plugin from <https://plugins.jetbrains.com/plugin/6317-lombok-plugin> to resolve errors due to missing getters and setters within the model package. Note, that getters and setters should be generated automatically due to Lombok annotations `@Getter` and `@Setter`.

4 (0.5pt) Task 1: Fix Implementation of Tag Service

The goal of this task is to fix issues related to tagging within the reporting tool. Simple tagging service supports adding tags and to listing all available tags. Addition of tags should track its author (creator) and the time when it was created. Run⁴ and fix all failing tests within the project. Only additive changes (mostly addition of Spring annotations) are needed to fix issues and modifications of tests are not allowed.

5 (0.5pt) Task 2: Implement Support for Tagging of Occurrence Reports

The goal of this task is to extend the implementation with an addition of tags and searching for tags related to occurrence reports. Uncomment existing source code from `OccurrenceReportServiceTest.java`. Implement missing functionality so that source code is compilable and all tests are passing. In addition, change the semantics of method `OccurrenceReportService.findAll(Tag tag)` to return a collection of reports ordered by occurrence start date (descending order). Create new JUnit test to test correct ordering of the collection returned by `OccurrenceReportService.findAll(Tag tag)` method. The test should fail for the previous implementation, while it should pass for the new one.

⁴Hint: Your IDE most-likely has option to run all JUnit tests within the project. Use it instead of execution of maven command manually (e.g. by 'mvn test') as it usually provides easier way to find failing tests and execute them again.