Seminar #11 – Security

Petr Křemen

December 13, 2017

1 Introduction

Download the source code of the reporting tool for this seminar from [3] and explore

Spring security offers the following annotations – `@PreFilter`, `@PostFilter`, `@PreAuthorize`, `@PostAuthorize`. Become familiar with them (EAR lectures, Spring web) before starting with the following tasks. While solving the tasks, you might find useful [1] and [2].

2 Tasks

2.1 Authorization

Ex. 1 — (0.5pt) Change `OccurrenceReportService` so that each user running this service only gets reports authored by him/her plus all reports with severity ACCIDENT. Use data-driven Spring security – you only need to use the above mentioned Spring annotations to achieve this. Test your solution on example data.

Ex. 2 — (0.5pt) Ensure that each user is only allowed to update reports created by himself/herself. You will need to perform the following modifications to the code:

- modify `OccurrenceReportService`. You will need to extend the `update` method from the `AbstractRepositoryService` so that you can annotate it. Make sure the method is annotated as `Transactional`.
- annotate the `update` method to ensure that each report can only be modified by its creator.
- extend the `RestExceptionHandler` class to handle the `AccessDeniedException` thrown by Spring whenever the user is not allowed to perform the update. Make sure the method returns the HTTP status 403.
- Test your solution on example data.

References

[1] Spring Expression Language. Spring. https://docs.spring.io/spring/docs/4.3.12.RELEASE/spring-framework-reference/html/expressions.html

[2] Expression-Based Access Control. Spring. https://docs.spring.io/spring-security/site/docs/3.0.x/reference/el-access.html

[3] EAR Seminars. https://cw.fel.cvut.cz/wiki/courses/ear/seminars