Seminar #11 – Security

Petr Křemen

December 13, 2017

1 Introduction

Download the source code of the reporting tool for this seminar from [3] and go through its security settings.

Spring security offers the following annotations:

`@PreFilter` for filtering input list based on security constraints expressed in SpEL.

`@PostFilter` for filtering output list based on security constraints expressed in SpEL.

`@PreAuthorize` for authorizing method execution based on security constraints expressed in SpEL.

`@PostAuthorize` for authorizing return from the method execution based on security constraints expressed in SpEL.

Become familiar with these annotations (EAR lectures, Spring web) before starting the following tasks. Refer to the Spring web pages [1] and [2] for details.

2 Tasks

2.1 Authorization

Ex. 1 — (0.5pt) Ensure that each user only sees reports authored by himself/herself together with all reports of severity ACCIDENT. Use data-driven Spring security 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. Hints:

- You will need to introduce the (`@Transactional`) `OccurrenceReportService.update` method inherited from the `AbstractRepositoryService` in order to annotate it.
- You will need to modify the `RestExceptionHandler` class to pass the Forbidden (403) status generated by Spring to the React client.

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