



Spring Web

Exercise: Todo REST Features

1. Clone the GitLab repository Todo (<https://gitlab.ti.bfh.ch/cas-sad-hs25/spring-web/todo>).
2. Add the Jackson libraries `tools.jackson.dataformat:jackson-dataformat-xml` and `tools.jackson.dataformat:jackson-dataformat-csv` as Maven dependencies to the REST service.
3. Implement a message converter `TodoCsvMessageConverter` that can write a todo list in CSV format and expand the controller annotations such that a todo list can be exported as XML or CSV.
4. Implement a filter `LoggingFilter` that extends the `HttpFilter` (<https://jakarta.ee/specifications/servlet/6.1/apidocs/jakarta.servlet/jakarta/servlet/http/httpfilter>) class and logs all HTTP requests and responses.
5. Implement a filter `AuthorizationFilter` that verifies that each API request contains a header with an API key.
6. Add the `spring-boot-starter-validation` Maven dependency and provide Bean Validation constraints (<https://jakarta.ee/specifications/bean-validation/3.1/jakarta-validation-spec-3.1#builtinconstraints>) such that identifiers must be positive, todos must have a non-empty title and due dates are in the present or future.
7. Ensure that the REST interface can also be called by JavaScript code originating from any other server.



8. Add the SpringDoc library (<https://springdoc.org/>) as Maven dependency and Swagger annotations (<https://javadoc.io/doc/io.swagger.core.v3/swagger-annotations/latest/index.html>) to the controller, and review the generated OpenAPI documentation.

Solution (<https://gitlab.ti.bfh.ch/cas-sad-hs25/spring-web/todo/-/commits>)

© Stephan Fischli, <https://distsys.bfh.ch/fischli> (<https://distsys.bfh.ch/fischli>)