

Static Testing Tools & White-box test Design

This section covers static testing tools, white-box test design, code coverage and their contribution to our project.

Static Testing Tools

SonarQube

To ensure code quality, we chose to use SonarQube as our primary static testing tool. SonarQube allows us to identify potential bugs, security risks, code smells, and violations of coding standards before running the application.

By analyzing the codebase statically, we gain insight into areas that require refactoring or further attention, which helps to improve the overall maintainability and structure of our backend.

Usage

Insert screenshot of SonarQube Dashboard, Issues, Code Smells and Complexity

ReSharper

Although ReSharper is an IDE-based tool and therefore not part of our official static testing tools for this section, it was still used by some of us as a supplementary aid during development

ReSharper provides on-the-fly code inspections, refactoring suggestions and quick fixes directly inside the editor. Its feedback helped identify minor issues such as unused variables, inconsistent naming, missing null-checks, and opportunities for simplifying expressions.

Code coverage

By using code coverage, we can identify which parts of the codebase are not covered by tests and which parts may be redundant. As shown in the coverage report below, the majority of the codebase is covered by unit tests for the **TitlesDto** class (report name: **EfCoreModelsLib.DTO.TitlesDto – Coverage Report.pdf**).

In the report, line **104** is not covered by any unit tests. Through white-box testing and debugging, we confirmed that the exception on line **100** is always triggered, regardless of the input. This means that line **104** is never executed in any scenario. Since the line is effectively unreachable, we removed it from the codebase.