Delft University of Technology

Software Engineering Methods
CSE2115

Assignment 3 Group 03B

Authors:
Antonios Barotsis
Nathan Klumpenaar
Dan Sochirca
Rado Todorov
Miloš Ristić
January 22, 2022

Introduction

For this assignment mutation testing was used to improve the test suite. The tool chosen for this is pitest[1]. This document will show some improved classes with their mutation score before and after. Included are also the commit SHA signatures in which the changes were made to achieve the new mutation score and next to that a brief explanation of the commit which if clicked with lead you to the gitlab page of the commit.

All images from the report will be given in the following manner where the first section indicates the name of the class, the second section the line coverage, and the third section the mutation coverage.

Name	Line Coverage		Mutation Coverage	
NonTargetedCompanyOfferController.java	100%	29/29	100%	13/13

Improvements

UserLoadEventListener - User microservice

UserLoadEventListener.java	17%	2/12	0%	0/7
UserLoadEventListener.java	100%	12/12	100%	7/7

Commits:

- d47989d24ff09e12818c05ac95de771b434bca24 - Added tests

Description:
3 tests were added to get the line coverage up to 100% which in turn also increased the mutation score from 0% to 100%

AuthService - User microservice

AuthService.java	100%	11/11	67%	4/6
AuthService.java	100%	11/11	100%	6/6

Commits:

- 74d178055d8be28238d4b869fa5a31fe271c3541 - Added tests
- 2775643382e8fc64dff05bc5eccf526a4280f69e - Fixed tests

Description:
Prior to the test improvement only 2 mutants had survived, one that replaced addition with subtraction, and one that returned an empty string instead of the expected string. The first mutant was killed by changing the *System.currentTimeMillis()* to *Instant.now()* in the class method *generateJwtToken()* and mocking it to verify if the expired date of the token was set correctly (System class is more difficult to mock). The second mutant was killed by mocking 1 dependency (JwtConfig), which allowed the *verify()* assertion to check if the call to this class was actually made.

UserService - User microservice

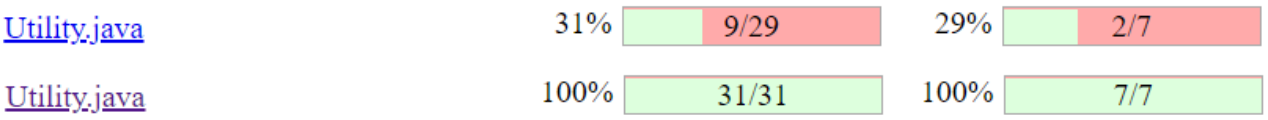
UserService.java	83%	25/30	50%	9/18
UserService.java	100%	30/30	89%	16/18

Commits:

- d47989d24ff09e12818c05ac95de771b434bca24 - Added tests

Description:
To improve the mutation coverage first all calls to the FileLogger and UserRepository were verified to have been executed by using a Mock instance. The AuthService was also mocked and verified to have done its job in the program. Also two additional tests were written.

Utility - Offer microservice

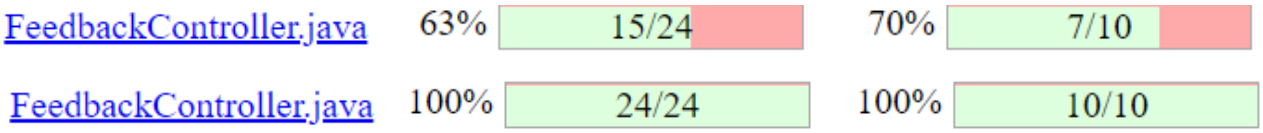


Commits:

- 9ee959d870cd77e8a6a11754a8598302ef7457f2 - Added tests
- 6292a51f73026d28e708c3ff0d1be592584849d1 - Added tests
- d3aee68b0521cd2f250f37a60fc2f8b1815d06cb - Fixed tests

Description:
In order to improve the mutation score tests were added, which cover all branches of the methods. It was discovered that 2 of the mutants were not killed initially because of the fact the it was difficult to test System calls, used for printing in the console. The implementation was then changed to use the FileLogger, which is more robust and easy to test. After the change in the implementation, tests were adjusted and the mutants were successfully killed.

FeedbackController - Feedback microservice



Commits:

- 0741d34564c08bee98d76589a62ecd0498b7a290 - Added tests
- b526ff6d59b59fdb263fb5cfc4563d98ccc44eb0 - Fixed tests

Description:
By increasing branch coverage the mutation score was in turn also increased. Additional tests were written to handle all scenarios in which exceptions are thrown and also one for handling unauthenticated users. For the mutation score it was enough to assert that what was returned was not null.

References

[1] H. Coles, "Java mutation testing tool." https://github.com/hcoles/pitest.