

Beaconfire Inc, Home Work, Week5 Day21.

By (Ping) Nalongsone Danddank.

ndanddank@gmail.com

wechatID: ndanddank

Short Answer:

1. Use your own word to explain TDD and why use TDD.

-> TDD, Test Driven Development, is the process that we write the test cases first and test each function which we want to implement. Then, we go to write the new code to develop and implement for each function to pass the test and making code simple and bug free. So because TDD approach help us to get fewer bugs and errors. When our code has fewer bugs, we will spend less time fixing them. And TDD produces a higher overall test coverage so TDD provide a better quality of the final product.

2. What is Mockito and why use it?

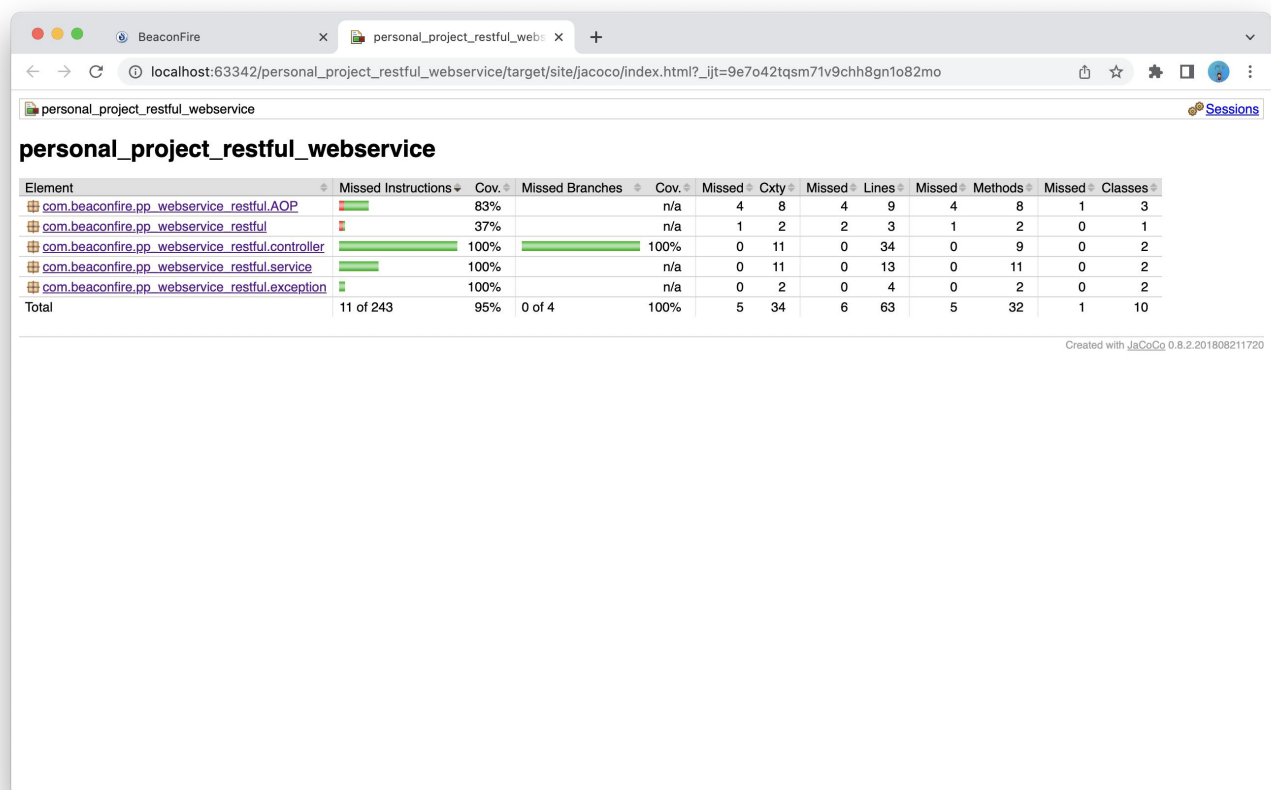
-> Mocking is a process used in unit testing when the unit being tested has external dependencies. The purpose of mocking is to isolate and focus on the code being tested and not on the behavior or state of external dependencies.

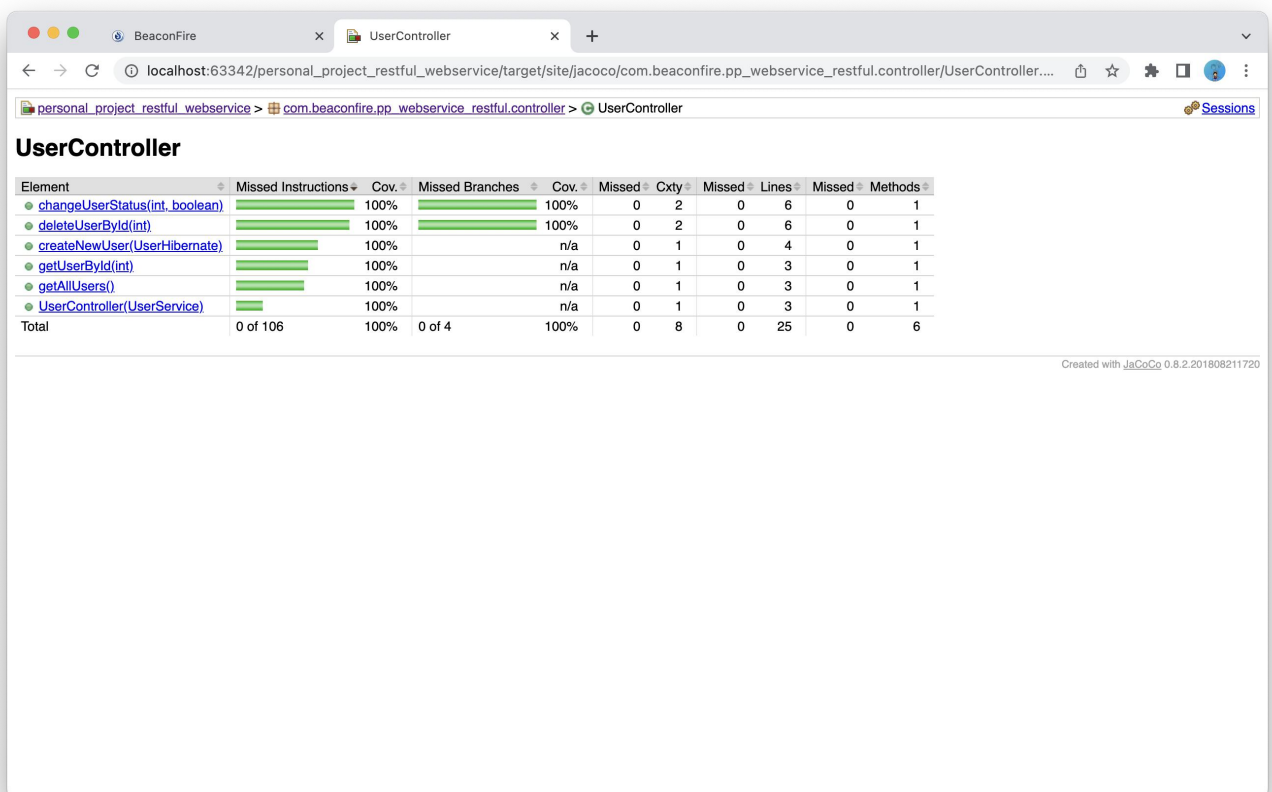
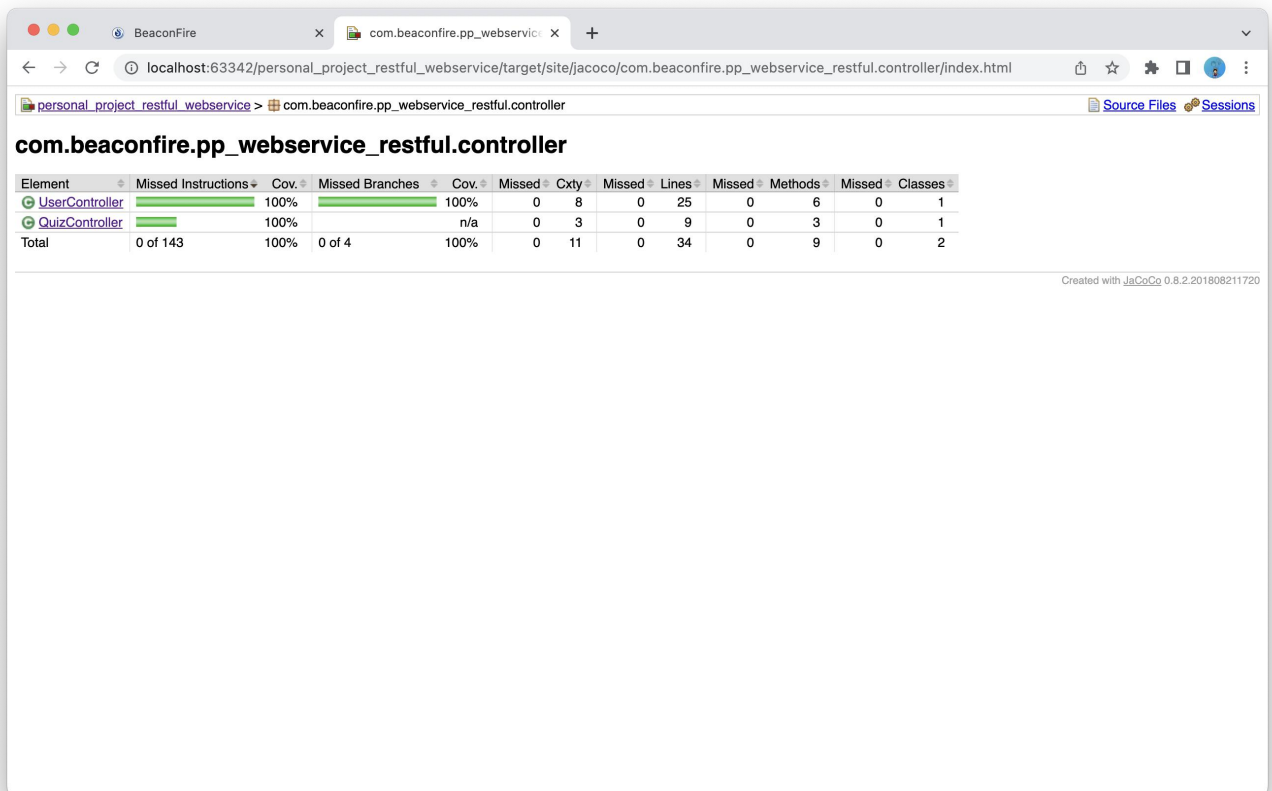
-> Mockito Is a java based mocking framework, used in conjunction with other testing frameworks such as JUnit and TestNG. It internally uses Java Reflection API and allows to create objects of a service. A mock object returns a dummy data and avoids external dependencies.

JACOCO

• Code coverage is a software metric used to measure how many lines of our code are executed during automated tests.

JACOCO Screen shortcut





```
35. }
36.
37. @GetMapping("/{userId}")
38. @PreAuthorize("hasAnyAuthority('read')")
39. public ResponseEntity<UserResponse> getUserById(@PathVariable int userId) throws UserNotFoundException {
40.     return ResponseEntity.ok(UserResponse.builder().status(ResponseStatus.builder()
41.         .success(true).message("Returning the User By Id.").build())
42.         .user(userService.getUserById(userId)).build());
43. }
44.
45. @PostMapping()
46. @PreAuthorize("hasAnyAuthority('write')")
47. public ResponseEntity<UserResponse> createNewUser(@RequestBody UserHibernate user) throws UserNotFoundException {
48.     User newUser = userService.createNewUser(user);
49.     return ResponseEntity.ok(UserResponse.builder().status(ResponseStatus.builder()
50.         .success(true).message("Created a new User.").build()).user(newUser).build());
51. }
52. }
53.
54. @DeleteMapping("/{id}")
55. @PreAuthorize("hasAnyAuthority('delete')")
56. public ResponseEntity<UserResponse> deleteUserById(@PathVariable("id") int id) throws UserNotFoundException {
57.     User newUser = userService.deleteUserById(id);
58.     if(newUser == null){
59.         throw new UserNotFoundException("Some thing wrong!");
60.     }
61.     return ResponseEntity.ok(UserResponse.builder().status(ResponseStatus.builder()
62.         .success(true).message("Delete User success.").build()).user(newUser).build());
63. }
64. }
65.
66. @PatchMapping("/{id}/status")
67. @PreAuthorize("hasAnyAuthority('update')")
68. public ResponseEntity<UserResponse> changeUserStatus(@PathVariable int id, @RequestParam(value="activate", required = true) boolean activate) throws
69.
70.     User newUser = userService.changeUserStatus(id, activate);
71.     if(newUser == null){
72.         throw new UserNotFoundException("Some thing wrong!");
73.     }
74.     return ResponseEntity.ok(UserResponse.builder().status(ResponseStatus.builder()
75.         .success(true).message("User Statue changed success.").build()).user(newUser).build());
76. }
77. }
78. }
```

BeaconFire x QuizController x +

localhost:63342/personal_project_restful_webservice/target/site/jacoco/com.beaconfire.pp_webservice_restful.controller/QuizController.h...

personal_project_restful_webservice > com.beaconfire.pp_webservice_restful.controller > QuizController Sessions

QuizController

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Cxty	Missed Lines	Missed Methods
getAlIQuizzesByUserId(int)	<div></div>	100%		n/a	0 1	0 3	0 1
getAlIQuizzes()	<div></div>	100%		n/a	0 1	0 3	0 1
QuizController(QuizService)	<div></div>	100%		n/a	0 1	0 3	0 1
Total	0 of 37	100%	0 of 0	n/a	0 3	0 9	0 3

Created with JaCoCo 0.8.2.201808211720

