Consultant Tracker Testing Policy

Edited By

Sibekezelo Mamba 16095414
Johan de Waal 16155140
Stephen Munro 16024479
Hulisani Mudimeli 16073364
Ngonidzashe Mujuru 16285256
Tatenda Mafunga 16094965

University of Pretoria
2018

Contents

1 Introduction

Travis CI (Continuous Integration) improves development process by automatically building and testing code modifications which in turn provides instant response or feedback on the status of the modification.

Travis CI is linked with our GitHub repository that contains our project source code. Travis was chosen because it makes use of continuous integration which is a development practice that joins small code alterations or modifications regularly instead of merging a large change when development cycle is completed.

2 Overview

Travis CI use its own system to perform tests. The automation of executions does not depend on the speed or performance of an individual's machine. Results of the tests are sent via email and shown in the Travis job status section.

3 Variances

Multiple conditions and events were detected during the test. This includes the condition of having to delete something that never existed. Travis helped integrate the test steps to avoid that risky condition from happening. The test cases below will elaborate further on how this was solved.

Most test cases were successful only when the tomcat server and the database server where running, otherwise, it would produce an error because of failure to communicate with the database.

4 Assessment

Java was used for back-end by the implementation of servlets that exchanges data with the front-end of our system. Individual units of source code of the system, associated with usage methods, data control and operating procedures will be tested using Unit Testing to ensure that the code works as it is intended to. Testing

Framework: Travis Continuous Integration
Module Tested: Java Test Files that in turn tests the servlets
Test Cases:

a) Creating Projects
b) Adding Consultants
c) Removing Consultants
d) Deleting Projects

5 Results

The following results shows the executed test cases' output and shows that all test cases have passed the test. All test cases used separate java files to test the individual parts of the system.

5.1 Test Case 1: Creating a project

After the execution of test case one, the database was populated with an extra Project called Dummy-Project. This test case was successful only when the tomcat server and the database server where running.

```
compile:
    [javac] Compiling 4 source files to /home/travis/build/hulisani-mudimeli/Test/testing-out

test1:
     [echo] Test Case 1 executing...
     [java] ==> Creating Dummy Project...
     [java] ==> Dummy Project Created...
     [echo] Test Case 1 finished...

BUILD SUCCESSFUL
Total time: 2 seconds


The command "ant test1" exited with 0.
```

Figure 1: Creating Project

5.2 Test Case 2: Adding Consultants to a project

Test Case 2 requires Test Case 1 to be successful so that consultants can be added to the desired Project. Upon execution test case 2, Dummy-Project had a

consultant added to it. This test case was successful only when the tomcat server and the database server where running.

```
$ ant test2
Picked up _JAVA_OPTIONS: -Xmx2048m -Xms512m
Buildfile: /home/travis/build/hulisani-mudimeli/Test/build.xml

compile:

test2:
     [echo] Test Case 2 executing...
     [java] ==> Adding Dummy Consultant to Dummy Project...
     [java] ==> Dummy Consultant Added...
     [echo] Test Case 2 finished...

BUILD SUCCESSFUL
Total time: 1 second


The command "ant test2" exited with 0.
```

Figure 2: Add Consultant to Project

5.3 Test Case 3: Removing a consultant

Test Case 3 requires Test Case 2 to be successful so that a consultant can be removed from the Dummy Project. Upon execution test case 3, Dummy-Project had the dummy consultant removed from it. This test case was successful only when the tomcat server and the database server where running.

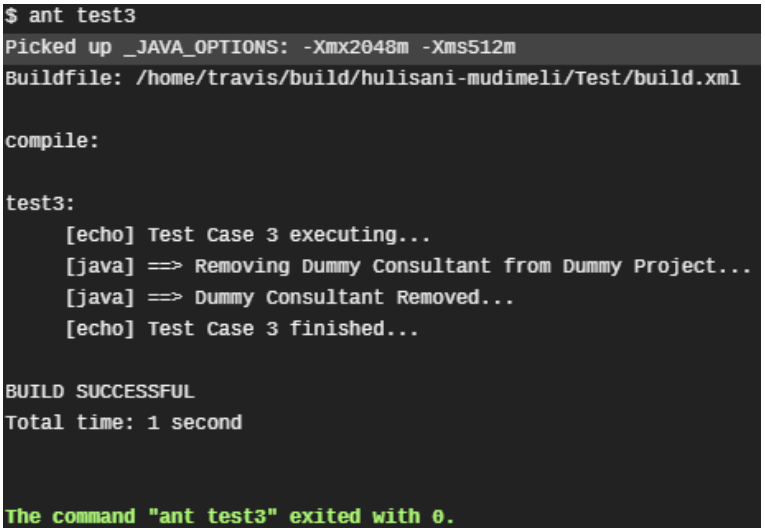

Figure 3: Remove Consultant from Project

5.4 Test Case 4: Deleting a Project

Test Case 4 requires Test Case 1 to be successful so that the created dummy project exists for removal. Upon execution test case 4, Dummy-Project was removed from the database. This test case was successful only when the tomcat server and the database server where running.

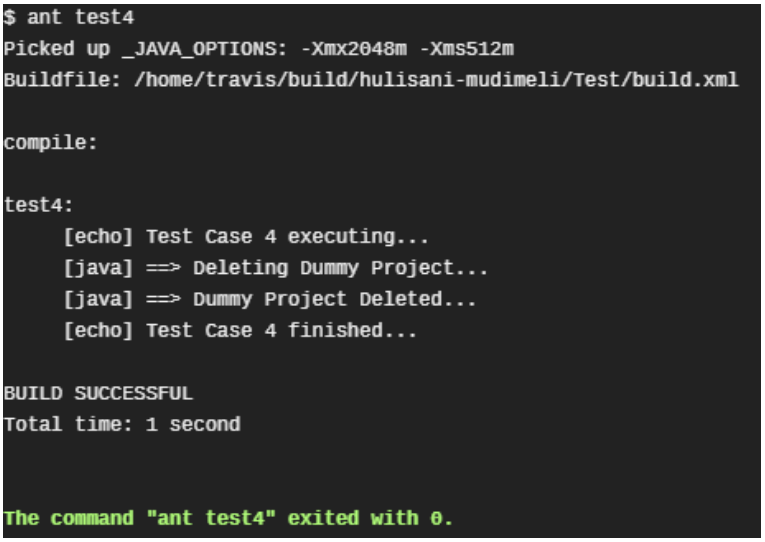

Figure 4: Delete Project

6 Evaluation

All four test cases for Consultant Tracking went through extensive automated testing by the help of the testing tool, Travis CI. After the execution of all this four test cases, errors would have occurred only when trying to remove a consultant that was never created and when trying to delete a project that never existed. But our automation process was programmed to follow the no erroneous steps such that no consultant or project is removed without their existence. Therefore, no errors where produced by our automated testing

7 Summary of Activities

The automated testing execution only took, 32 seconds to run all the configurations that have been specified in the .travis.yml file.

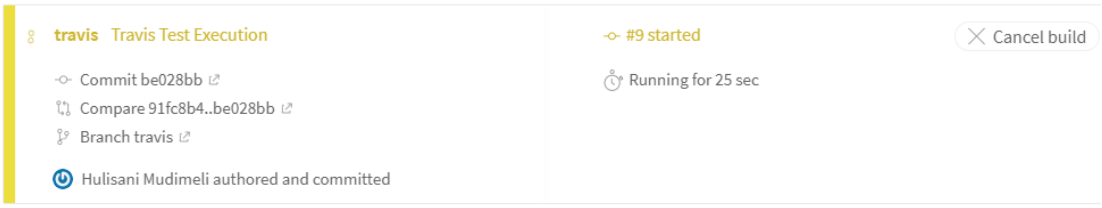

Figure 5: Testing in progress

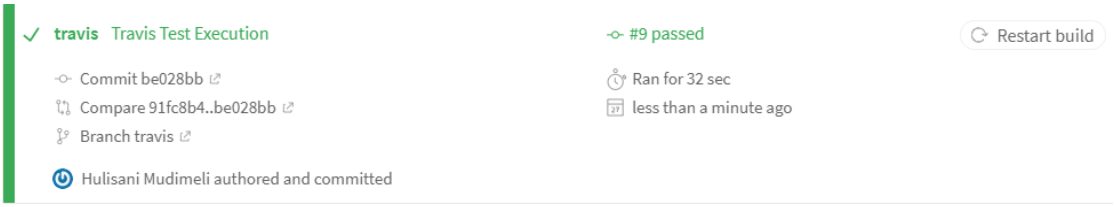

Figure 6: Testing completed