CODECOVER TOOL

Software Engineering Process & Quality Management (SE3010)
Group Assignment

Software Engineering Weekend Batch : **WE_SE_17**

Student ID	Name With Initials
IT17092548	**Kamburugamuwa K.L.A.P.T**
IT17121170	**Ranasinghe N.K**
IT17120012	**T. A. S. Shashiprabha**
IT17119122	**Liyanage I.M**

CONTENT

1. Introduction

What is Code Coverage?

Code coverage is the percentage of code which is covered by automated tests. Code coverage measurement simply determines which statements in a body of code have been executed through a test run, and which statements have not.
Coverage is used by developers and vendors to indicate their confidence in the readiness of their software. Evaluating software coverage and taking proper actions lead to an improvement of software quality.It helps in evaluating the effectiveness of testing providing data on different coverage items.

Code coverage analysis is the process of

1 .Find areas of a program not exercised by a set of test cases
2. Create additional test cases to increase coverage
3. Determine a quantitative measure of code coverage
4. Identify redundant test cases that do not increase coverage.

So there are various code cover tools for java at present. Though this document, we are discussing about the Codecover tool and how it affect the real world scenarios.

Codecover is a extensible open source code cover tool with the following features.

1. Support Statement Coverage.
2. Branch coverage
3. Loop Coverage
4. Strict Condition Coverage

Codecover tool use java-based template engine (velocity) for creating source code and reports. By using this tool, we can get most accurate coverage measurement

Integration

There are three platforms which allows us to do the integration.

• Command line (Linux, Windows, Mac OS)
• Eclipse
• Ant

- integrate codecover tool to your ant script is not easy. But this way is powerful.
- Most powerful way is, using codecover in Eclipse.
- The reason is all the interesting views and convenient testcase buttons can’t be handle in a console.

Reports

There are three types of reports in codecover tool.

- Customizable – summarizes and output its data the way that user wants. Can adapt the report template in order to create text-based output looking can add stylesheets, highlight results the way you want. You can customize the report.

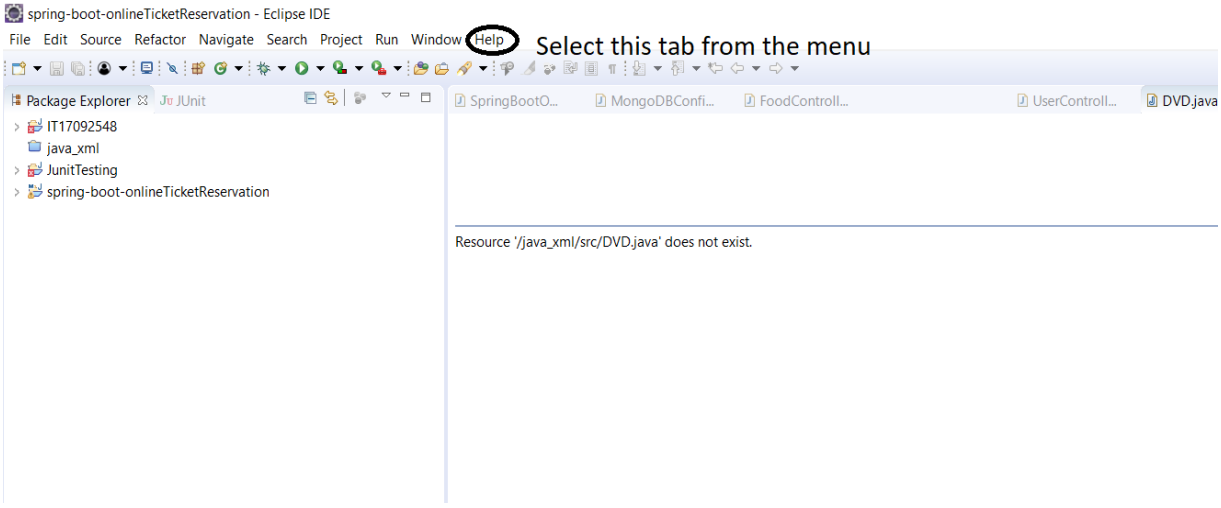
- On-hand – Saves every data of interests. Can create test reports of past test sessions even containing the source code its belonging to. You can notice the undercover parts too.

- Flexible – Outputs CSV and HTML files. By use of java language you can create a new report generator, so you can get the file format you need or you can send the results via mail.

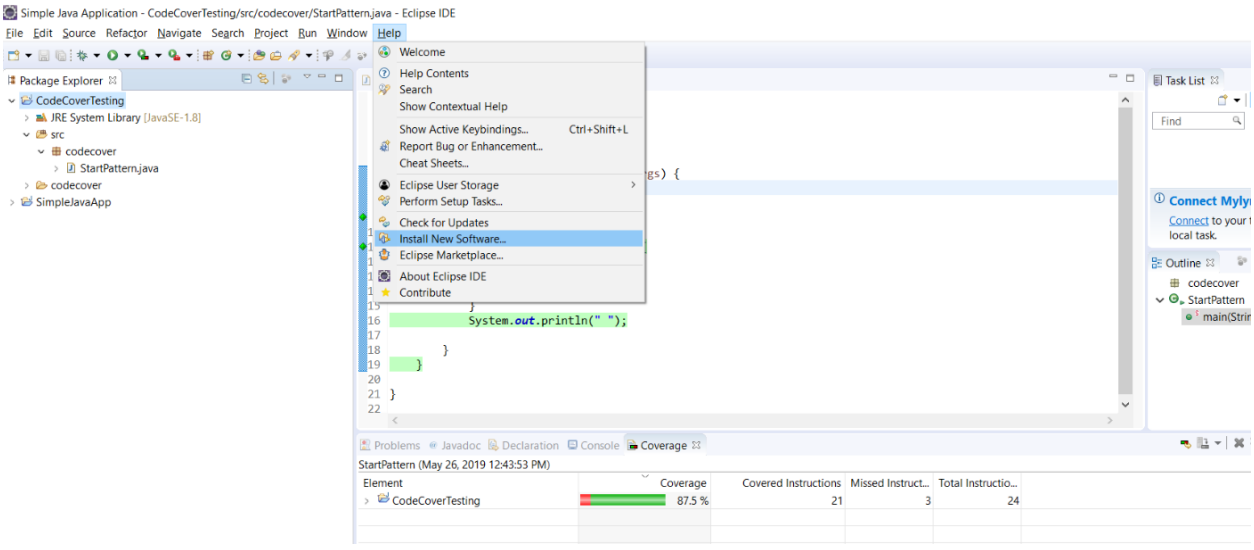
2. Configuration of the tool

2.1 Installation guide of Codecover on eclipse IDE

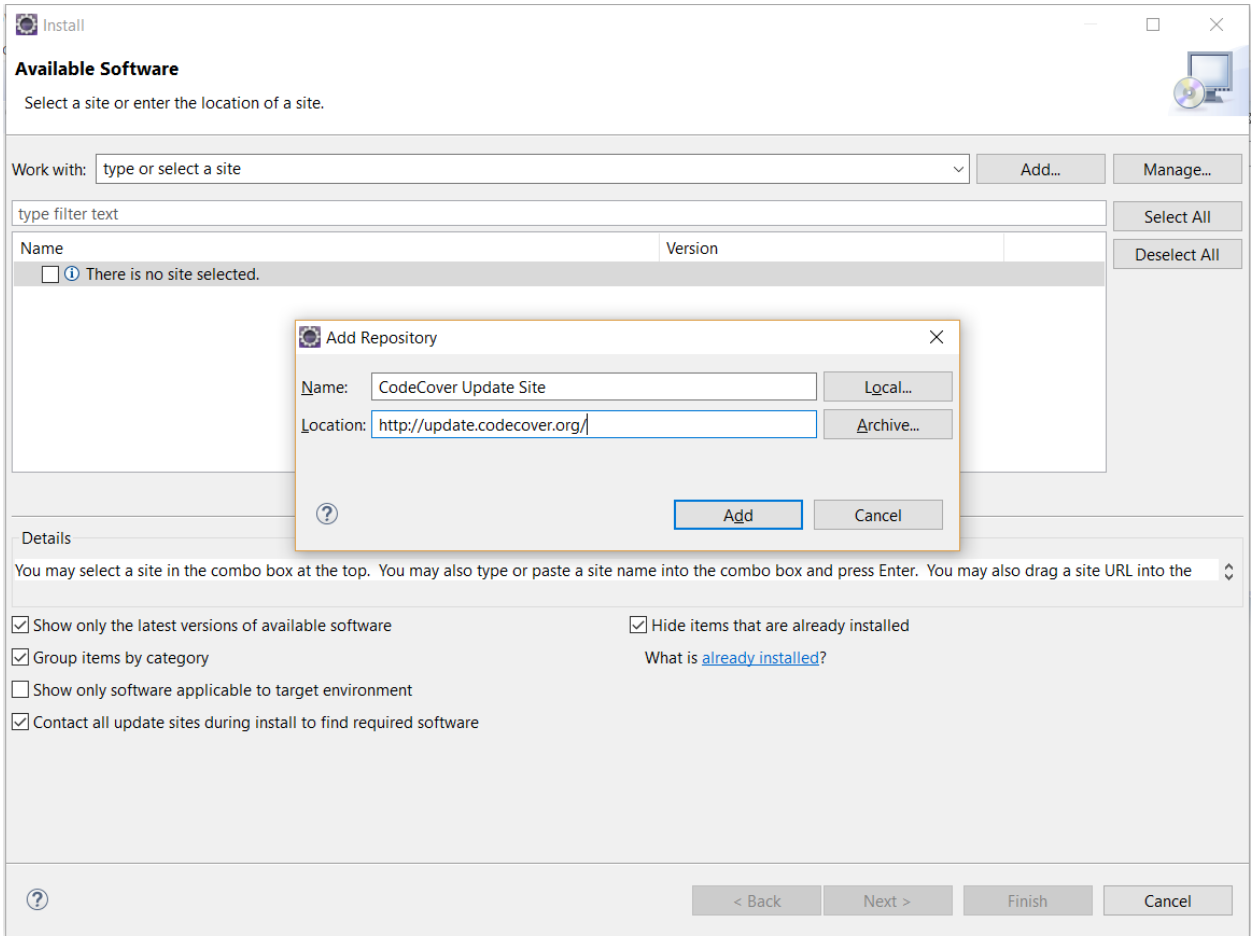
1. Open the eclipse IDE



2. From the main menu, select the “Help”,then select the install new Software under the Help.



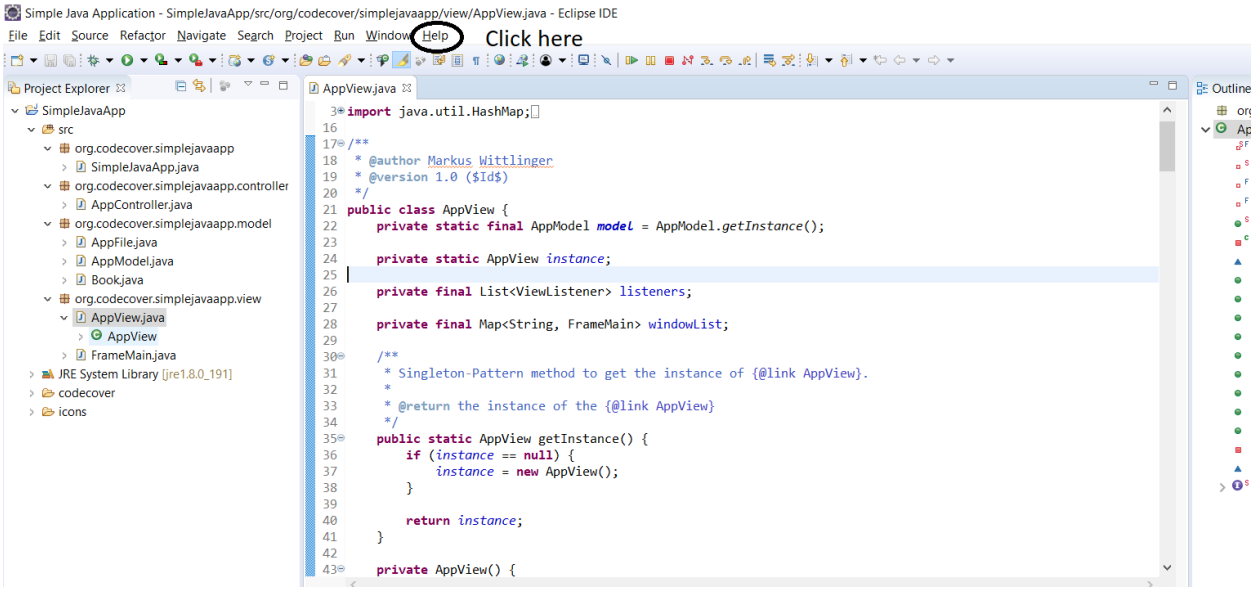
3. Then select the Add button in the pop up window and type the name as “CodeCover Update Site” and also location as “http://update.codecover.org/ “ . Then click Add button.



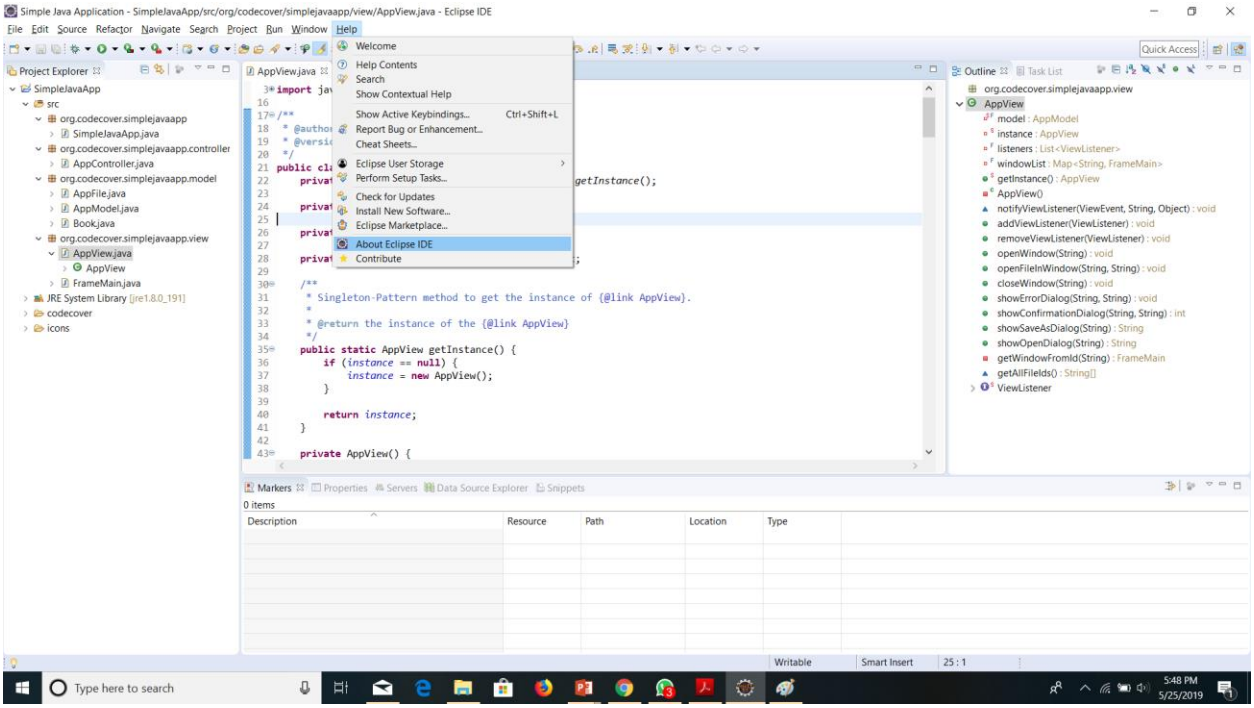
5. Then select the added link and select the version and press **Next**.
6. Again navigate to the next window from the detail window.
7. Accept the terms and conditions and navigate with Next.
8. All the dependencies will be downloaded.

Check whether the CodeCover was installed in eclipse properly.

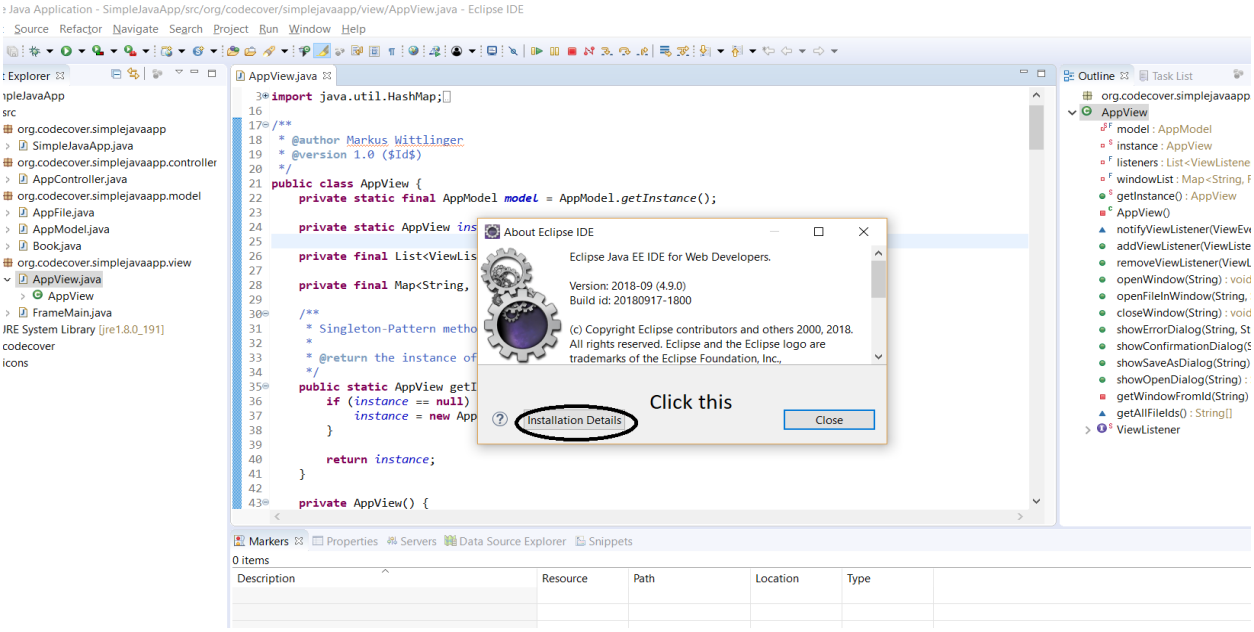
Go to help tab in the main window of the eclipse.



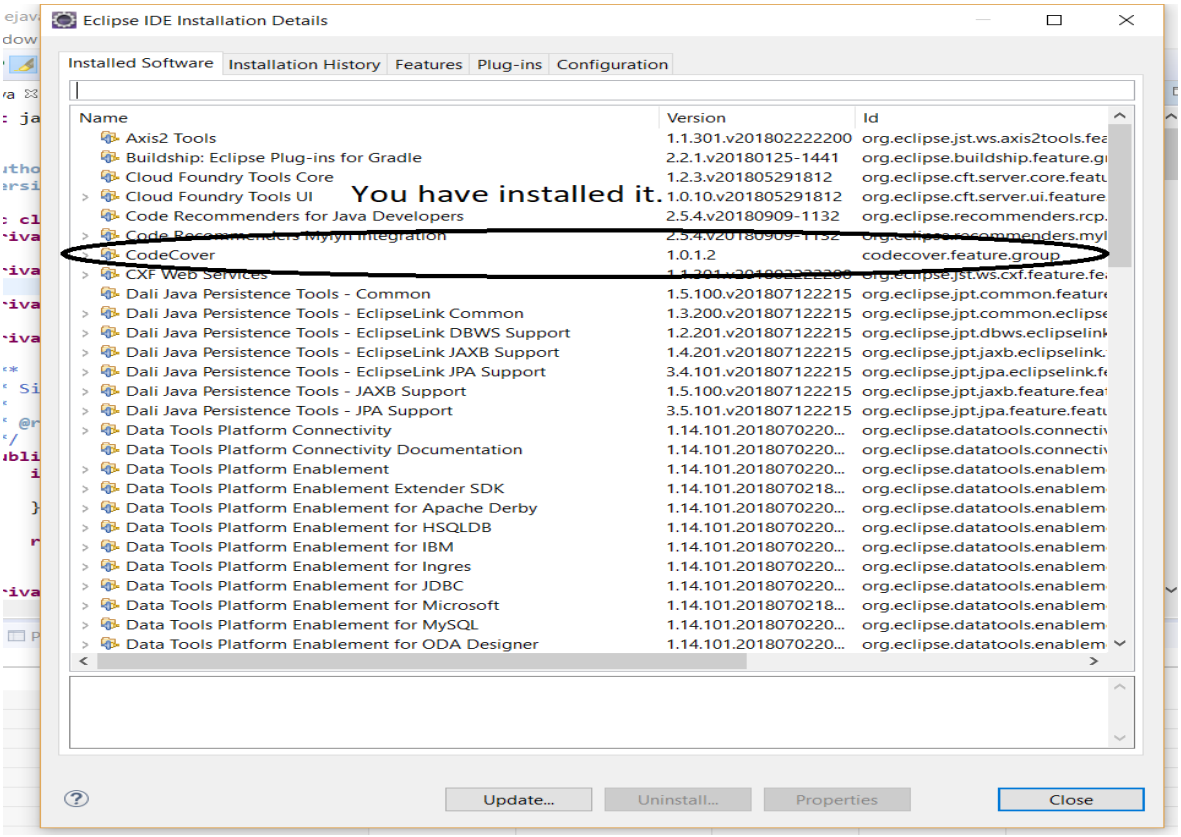
Then select the About Eclipse IDE.



Click on Installation Details.



Now you can see the CodeCover and its version on your Eclipse IDE



2.2 Use standalone CodeCover version

You can download the standalone codecover version from the going through to this URL.
Url :
For command line usage, codecover.sh (Un*x) and codecover.bat (Windows) wrapper scripts are provided. You may want to set your PATH environment variable to include the folder to which you extracted CodeCover, so you don't need to type in the full path to CodeCover every time.

3. Other tools for Code Coverage

EMMA TOOL

This tool is open source as the Code Cover tool. Emma is used for both open-source and commercial development projects. Emma distinguishes itself from other tools by going after a unique feature combination.development while keeping individual developers work fast and and iterative.This tool is essential for detecting dead code and verifying which parts of an application are actually exercised by the test suite and interactive use.
The main features of Emma, which represent its advantages are: Emma can instrument classes for coverage either offline (before they are loaded) or on the fly (using an instrumenting application class loader)
Supported coverage types: class, method, line, basic block; Emma can detect when a single source code line is covered only partially; Output report types: plain text, HTML, XML.

COBERTURA

Cobertura is a free Java code inclusion detailing device, that computes the level of code gotten to by tests. It tends to be utilized to recognize which parts of your Java program are inadequate with regards to test inclusion. It depends on jcoverage. Cobertura should take a shot at any stage with Java 5 or more up to date. The Cobertura download bundles incorporate every one of the conditions required (ASM, log4j, and so forth.)

GRETEL

Gretel is a residual take a look at insurance monitoring tool evolved with the aid of way of the university of Oregon. The maximum trendy version (which turned into again in 2002) presents assertion insurance tracking, identifying which traces of Java have been finished and which code has no longer been touched with the useful resource of checking out.

4. Justification for tool selection with a comparison to few other similar tools.

Code Coverage Tool	**CodeCover**	**Cobertura**	**Emma**	**Gretel**
Type of the software	Open source	Open source	Open source	Open Source
Coverage types that can be covered.	Statement coverage. Branch coverage. Loop coverage. Condition coverage.	Statement coverage. Branch coverage.	Statement coverage. Branch coverage.	Statement coverage. Branch coverage.

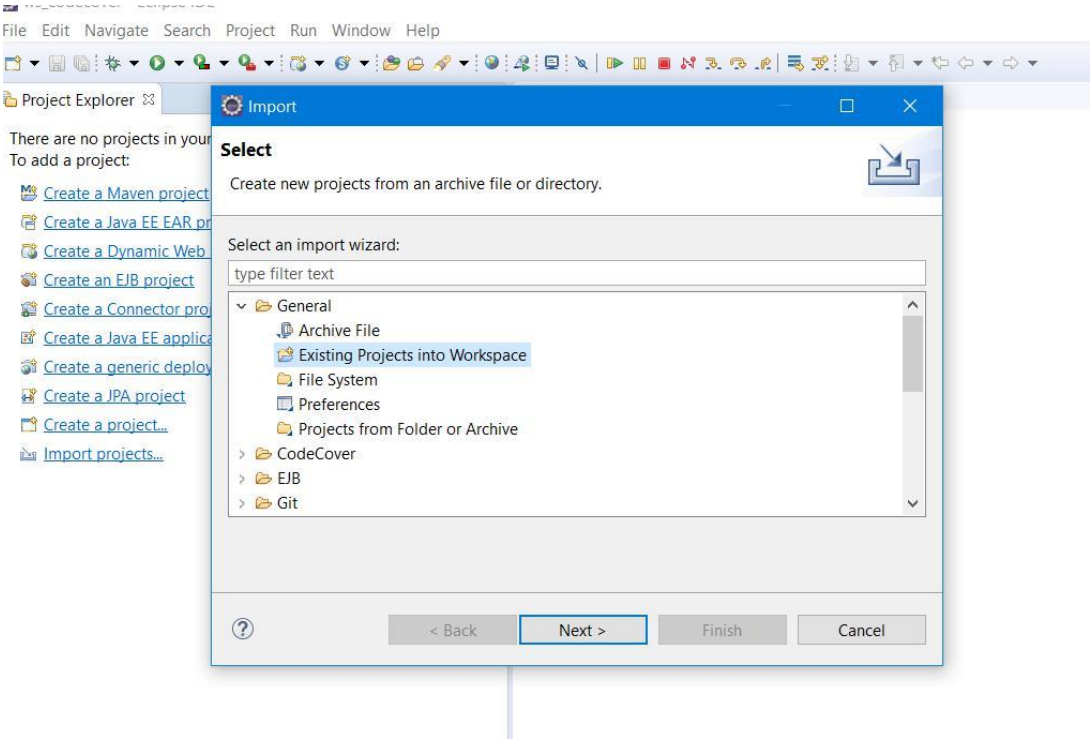
Is GUI support?	Yes	Yes	No Command line interface	No Command line interface
Reports type	Coverage view. Correlation view. Boolean analyzer Views. HTML format.	XML. HTML.	Text file. HTML format.	Line Table
Programming languages	Java COBOL	Java	Java	Java
Instrumentation	Source code instrumentation	Byte code instrumentation	Byte code instrumentation	Source code instrumentation

Comparing to other similar tools like Cobertura, Emma and Gretel, Codecover tool has additional supported coverage called Loop and condition coverage. It's an advantage for a tester. Comparing to the other two tools mentioned above Codecover supports COBOL programming language and it can generate reports as coverage view, correlation view, Boolean analyzer views and in HTMLformat. So when compare with other tools the best tool for the code coverage is Codecover.

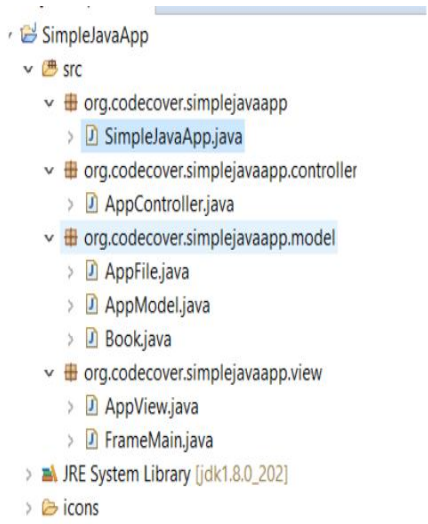
5. Tool Usage Using a Selected Scenario.

Here, we used a simple java project which give in the CodeCover.

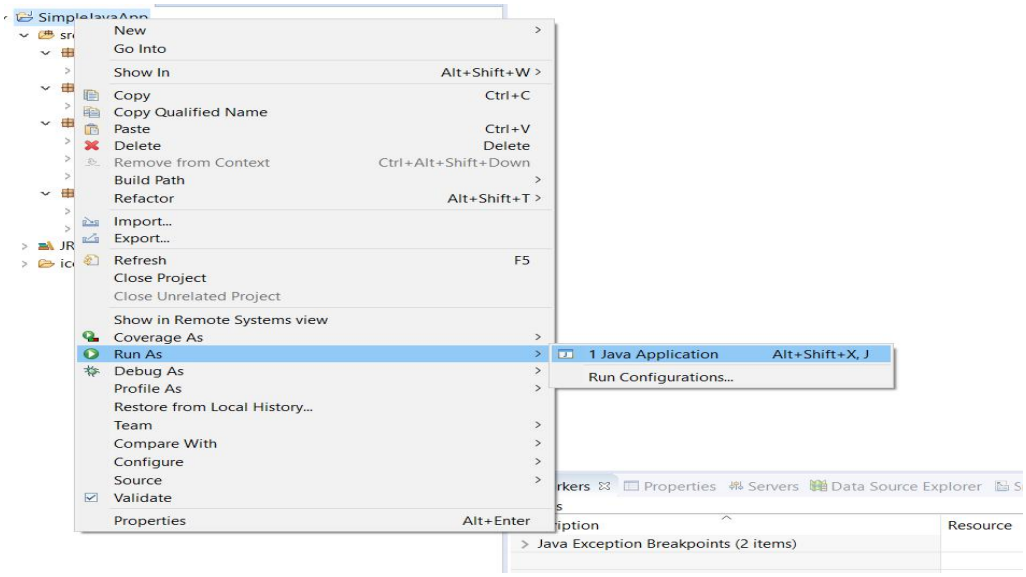
I. Import the project to eclipse workspace.



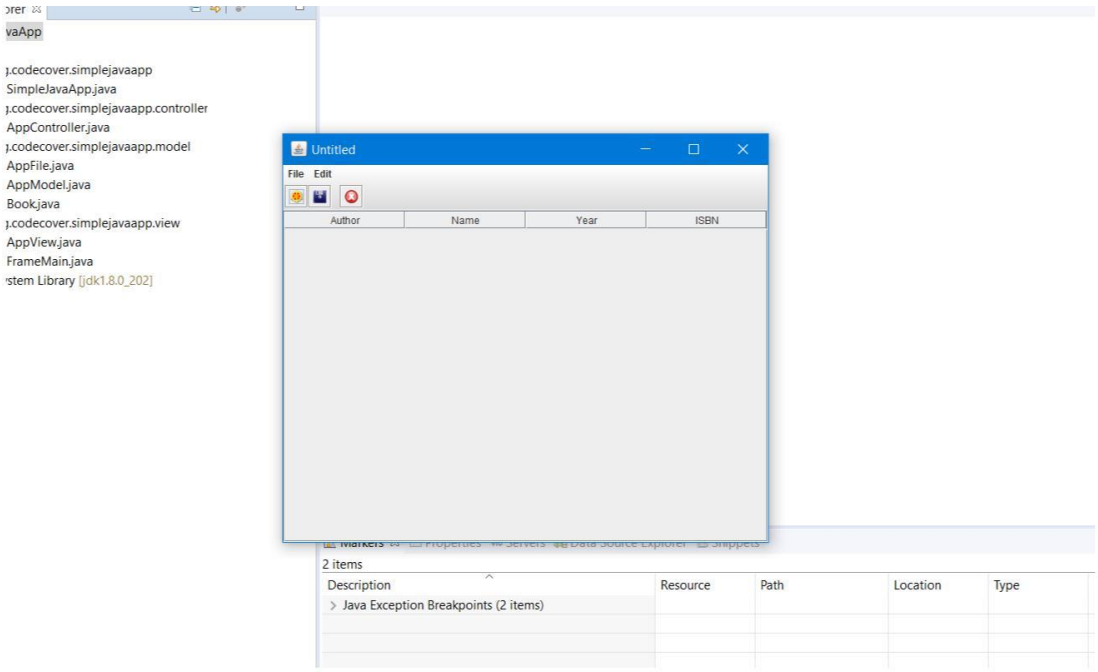
II. After importing.



III. Then, we have to run project as java application. Because we need to get runnable file into run configurations for future processes.



IV. This is the application which is the System Under Test.

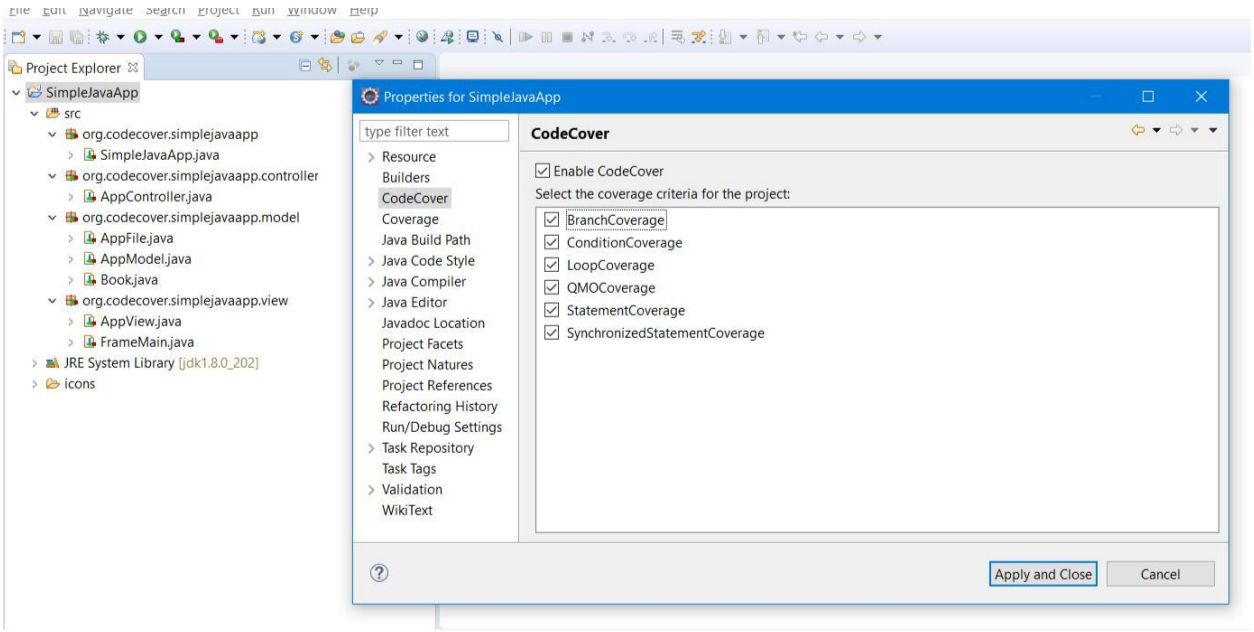


There are three ways to execute the SUT.

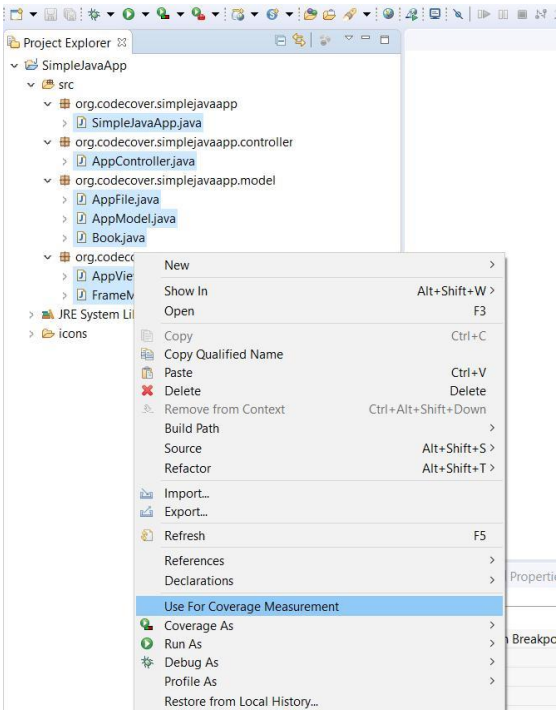
A. **Normal Execution – All the measured data is collected into single test case.**
B. **Live Notification Execution – Record individual test cases.**
C. **Junit Execution – Use existing Junit test suite to define the test case Here we do first two methods.**

A. Normal Execution

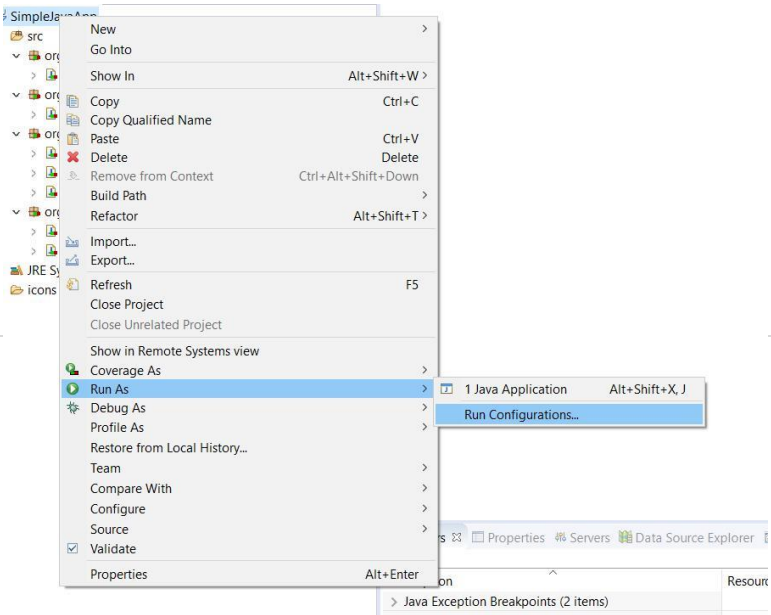
V. First, we need to enable CodeCover for imported project. Go to properties.



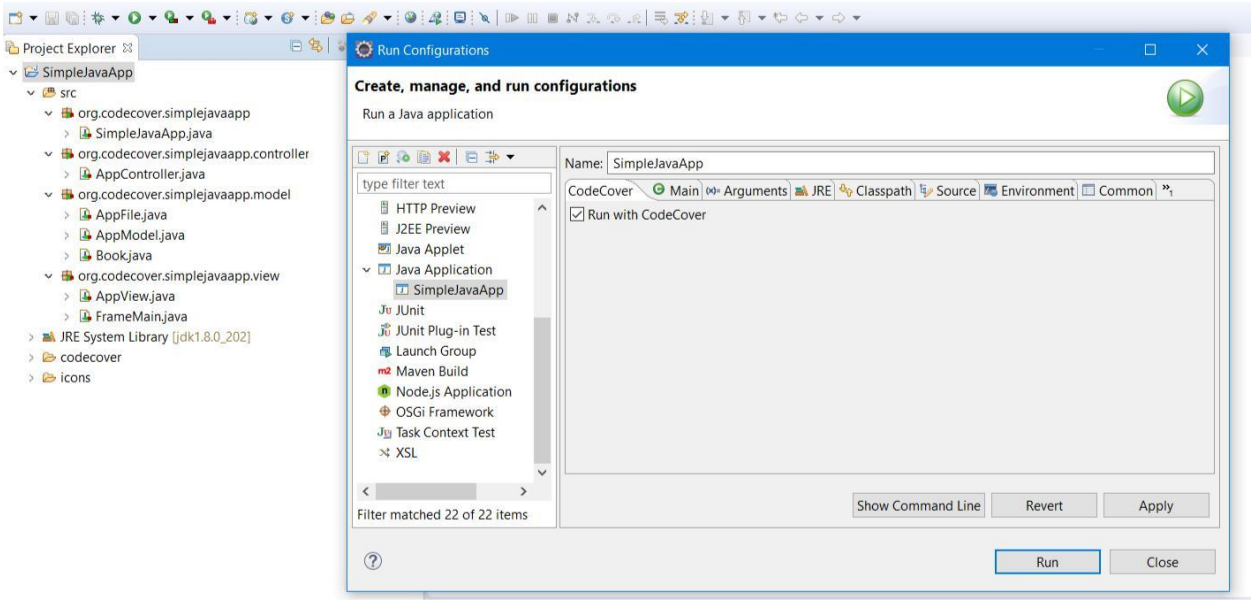
VI. Select all classes which we need to test and right click on those files and select ‘Use for coverage measurement’.



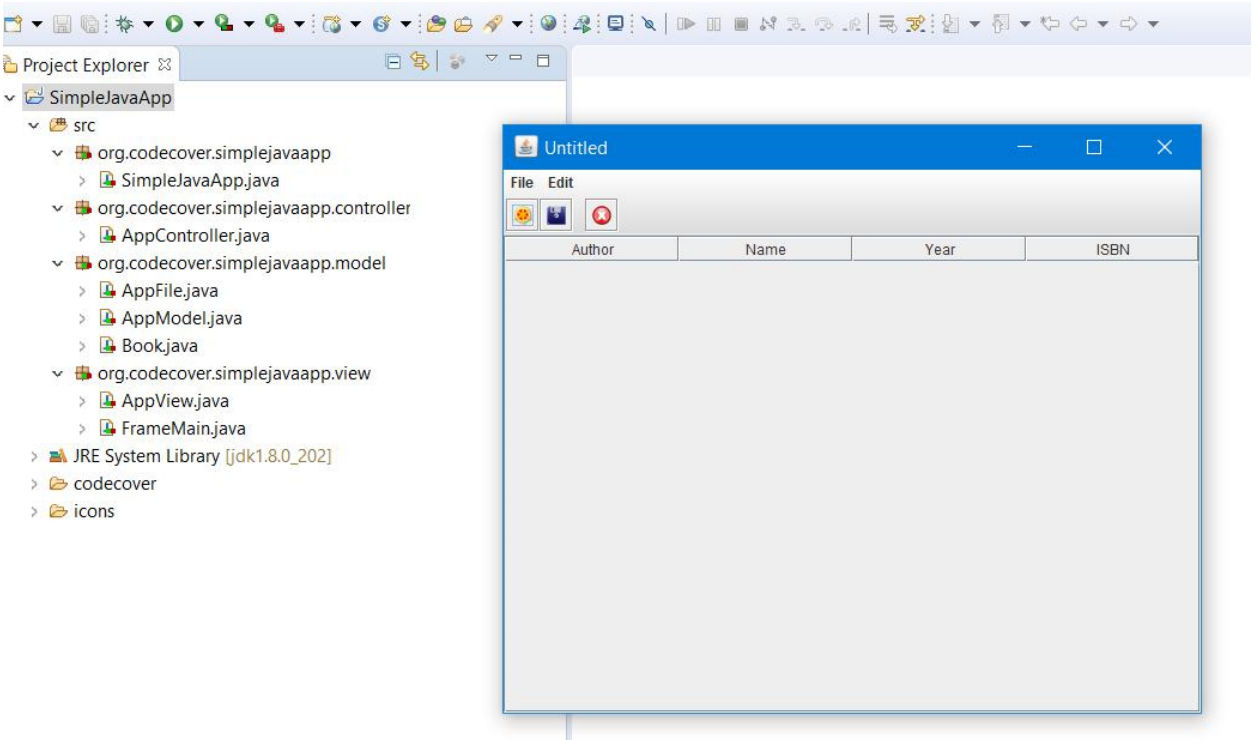
VII. After that, go to ‘run configuration’.



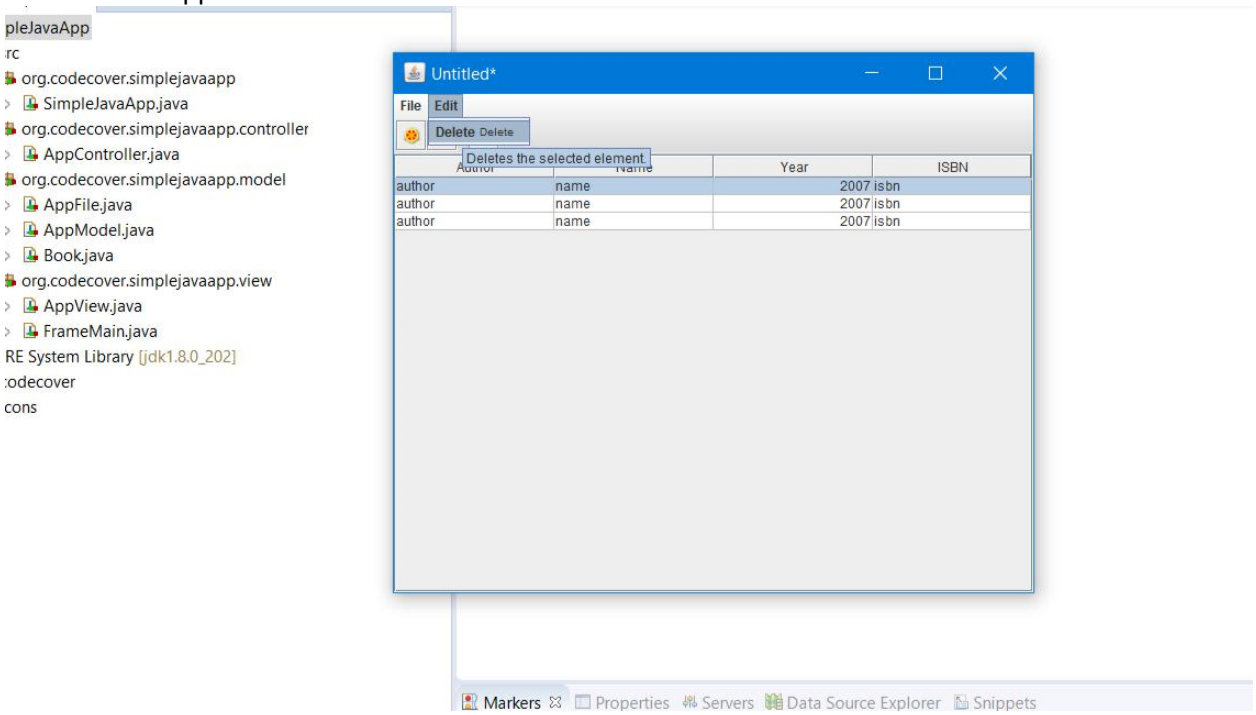
VIII. Then, run with code cover.



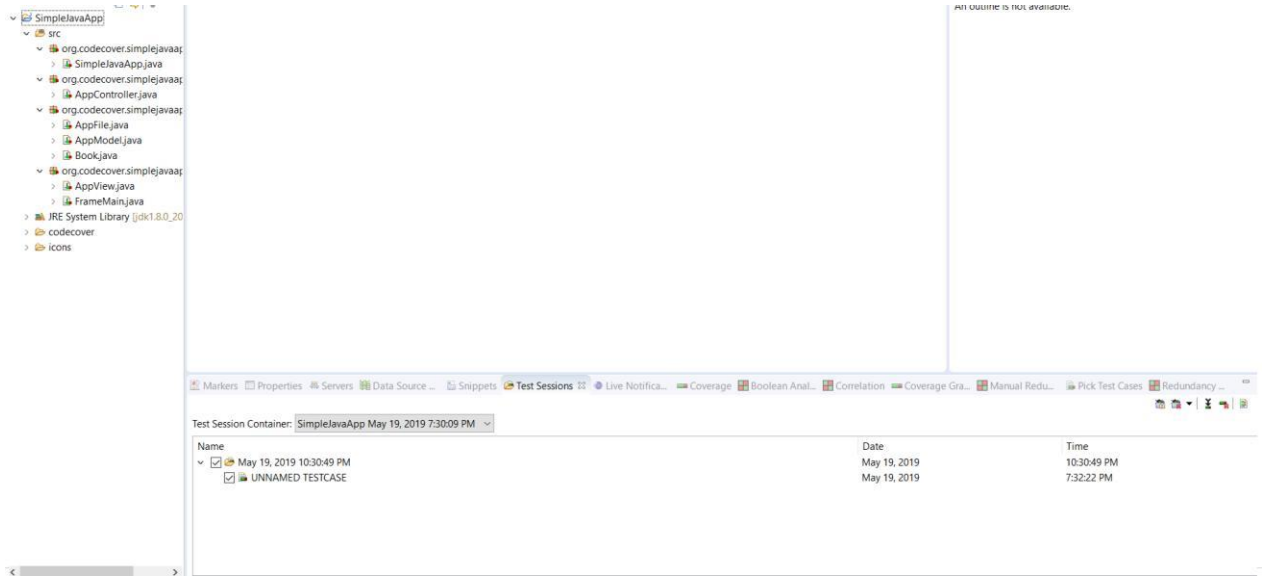
IX. This the running SUT. Now, we are doing on this application will record in testsession.



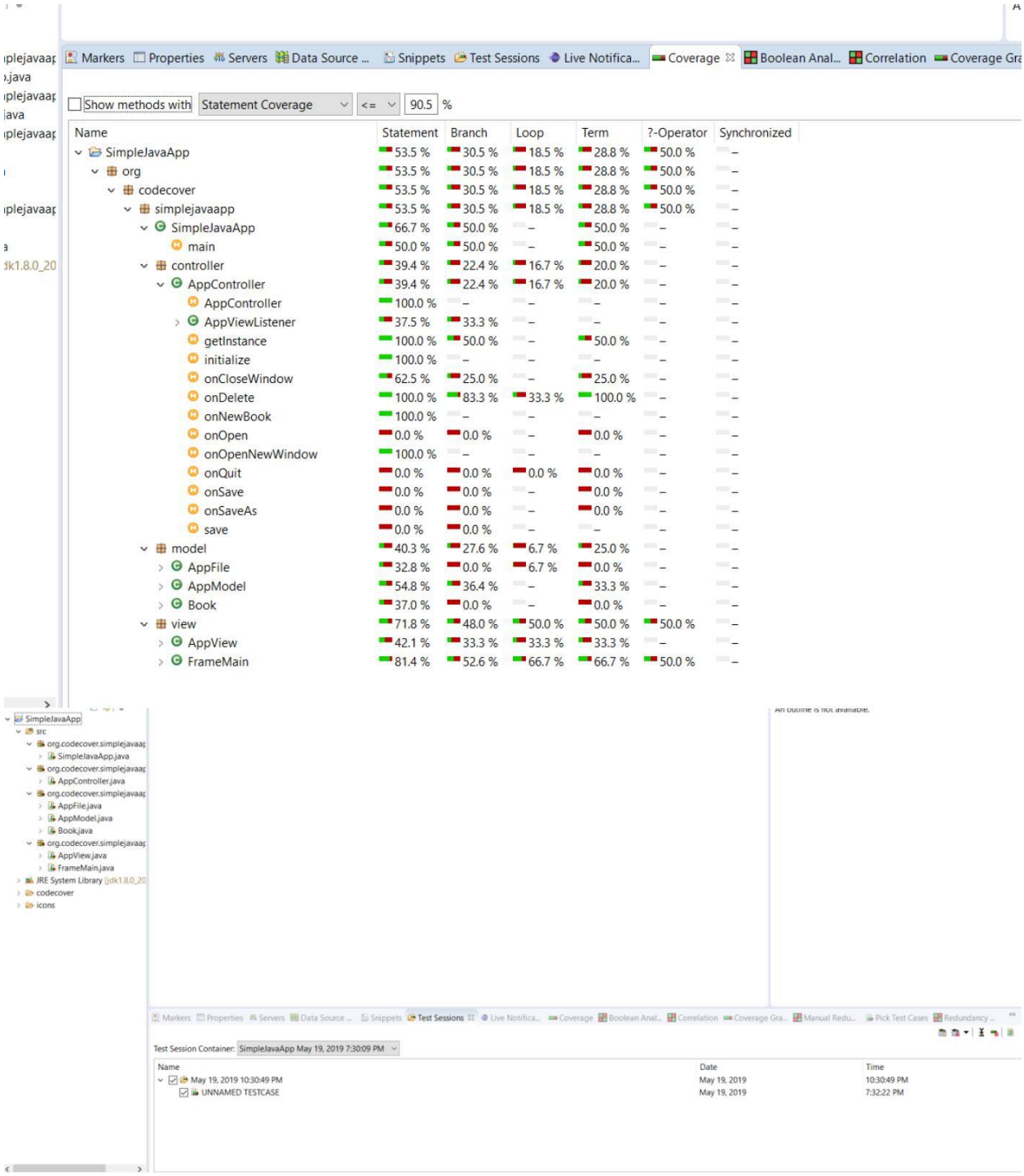
X. Let's add some books and delete some books. After doing what we want to test, close application.

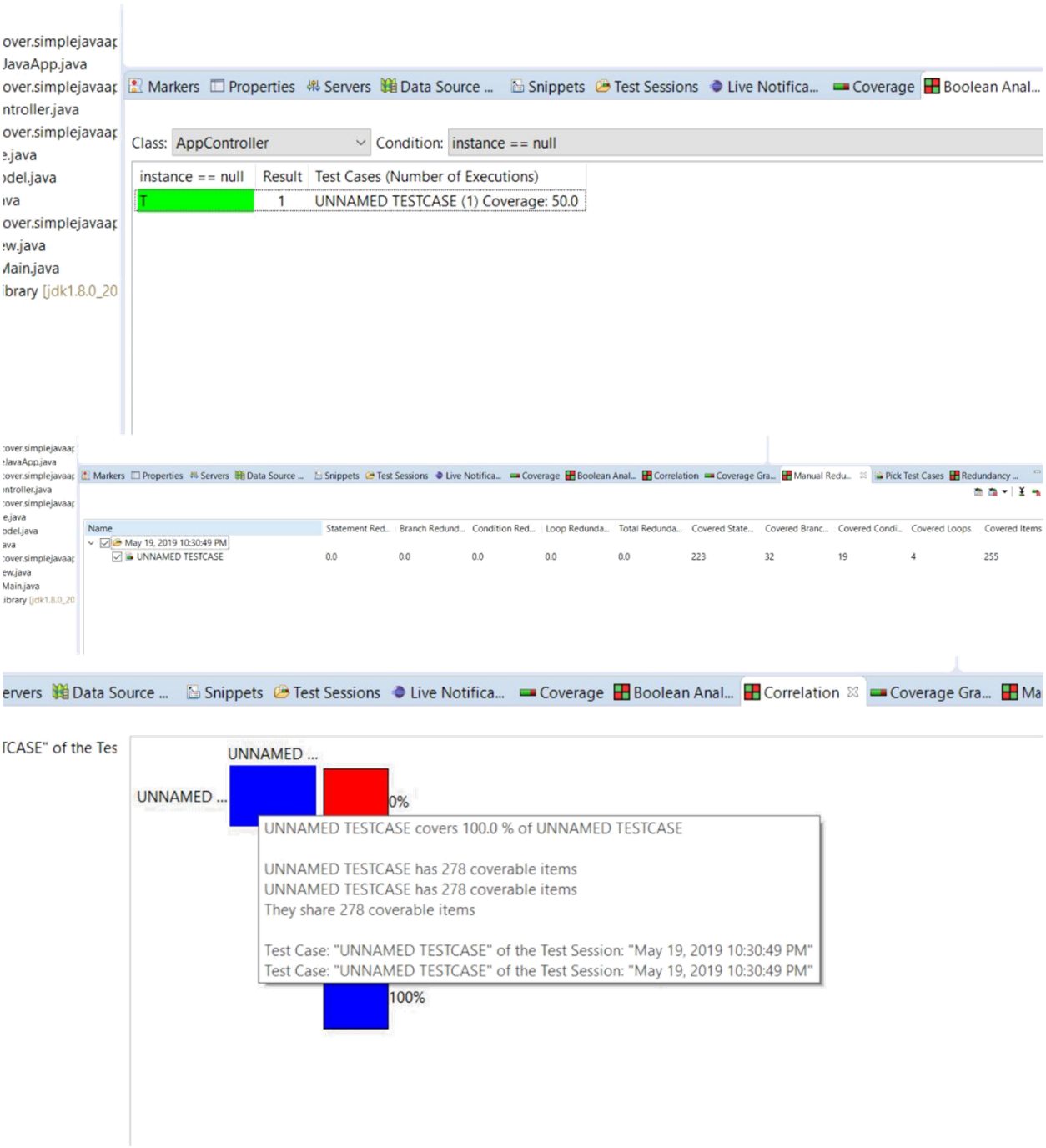


XI . Now we can see the measured data. For that, we have to add Test Sessions View, Coverage View, Correlation View, Boolean Analyzer, Pick Test Case View. Now we can see our test session in test session view.



XII. Let’s see measured data in different views.



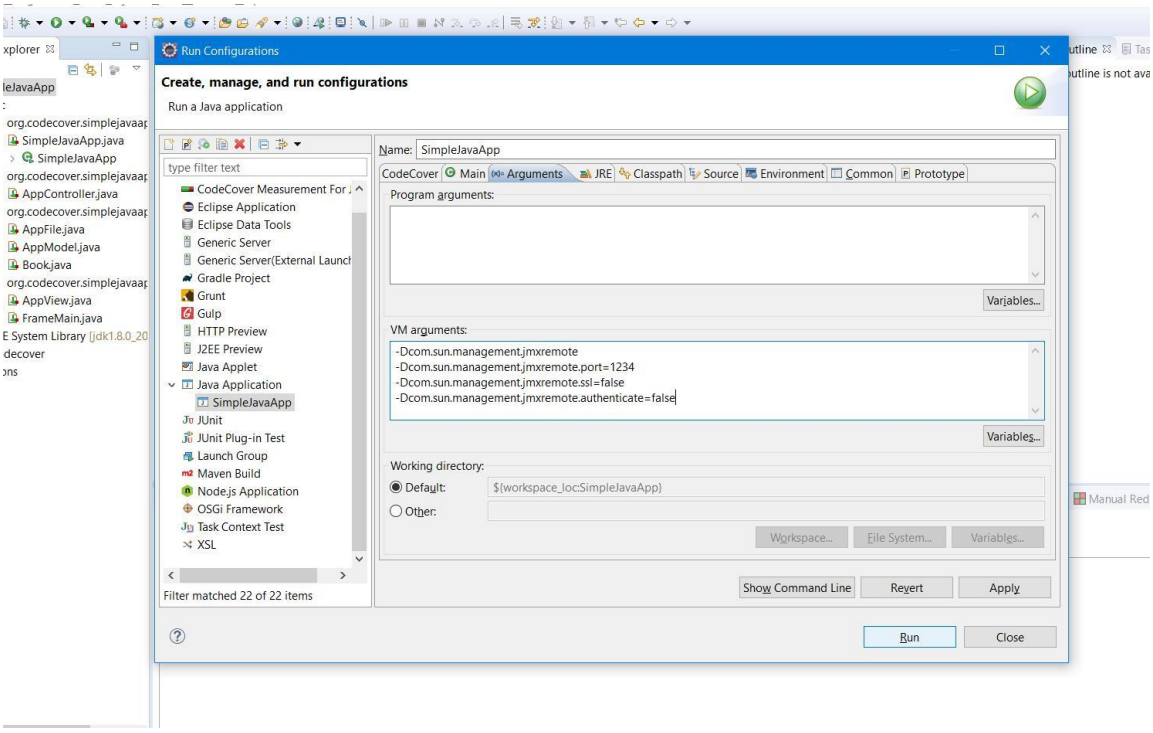

Markers
Properties
Servers
Data Source ...
Snippets
Test Sessions
Live Notifica...
Coverage
Boolean Anal...
Class: AppController
Condition: instance == null
instance == null
Result
Test Cases (Number of Executions)
1
UNNAMED TESTCASE (1) Coverage: 50.0
Correlation
Coverage Gra...
Manual Redu...
Pick Test Cases
Redundancy ...
Name
Statement Red...
Branch Redund...
Condition Red...
Loop Redunda...
Total Redunda...
Covered State...
Covered Branc...
Covered Condi...
Covered Loops
Covered Items
May 19, 2019 10:30:49 PM
UNNAMED TESTCASE
0.0
223
32
19
4
255
UNNAMED ...
0%
UNNAMED TESTCASE covers 100.0 % of UNNAMED TESTCASE
UNNAMED TESTCASE has 278 coverable items
UNNAMED TESTCASE has 278 coverable items
They share 278 coverable items
Test Case: "UNNAMED TESTCASE" of the Test Session: "May 19, 2019 10:30:49 PM"
Test Case: "UNNAMED TESTCASE" of the Test Session: "May 19, 2019 10:30:49 PM"
100%

XIII. Also, we can see source code is highlighted according to measured data.

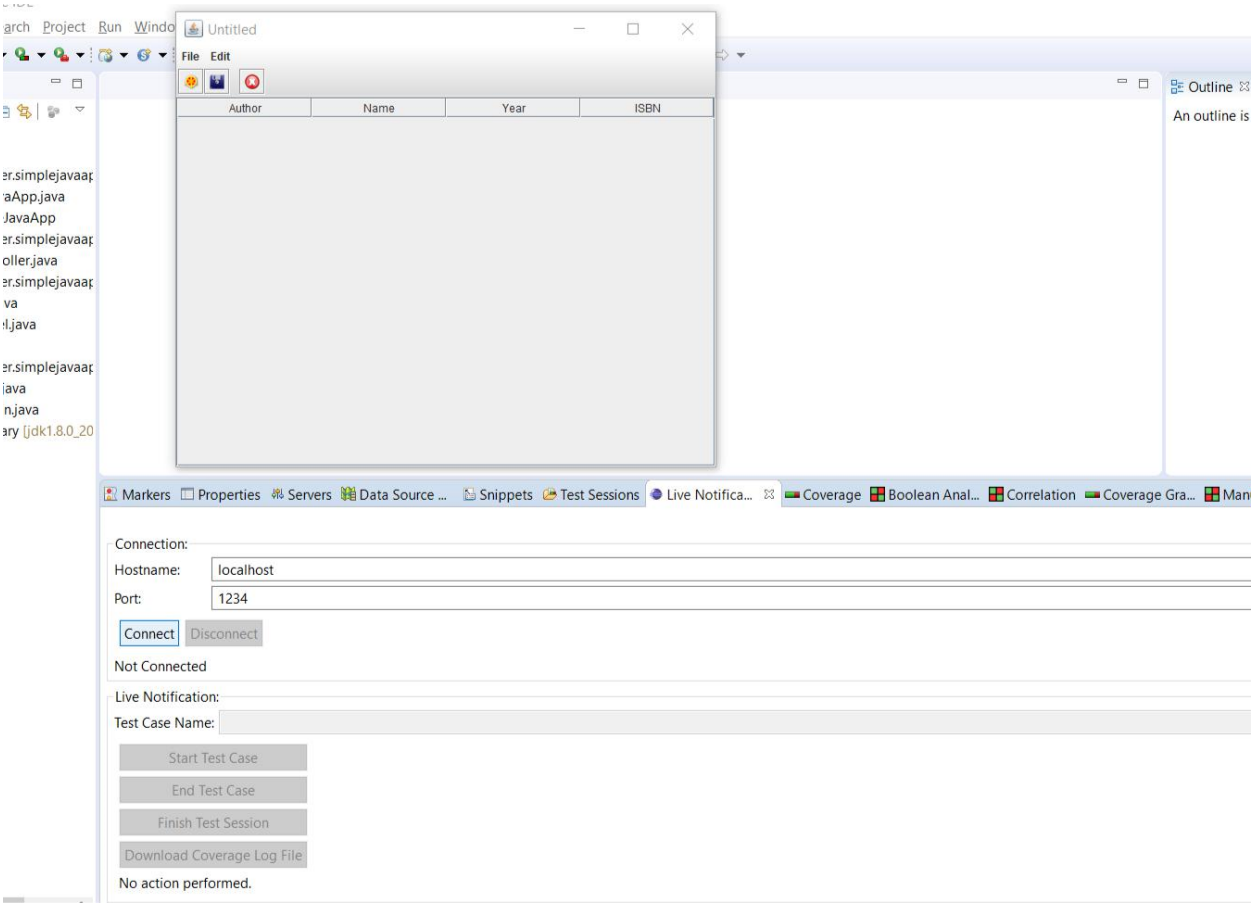
```
import org.codecover.simplejavaapp.controller.AppController;

/**
 * @author Markus Wittlinger
 * @version 1.0 ($Id$)
 */
public class SimpleJavaApp {
    final static boolean thisIsAMac = System.getProperty("mrj.version") != null;

    /**
     * @param args
     */
    public static void main(String[] args) {
        if (thisIsAMac) {
            System.setProperty("apple.laf.useScreenMenuBar", "true");
        }

        AppController.initialize();
    }
}
```

B. Live Notification Execution

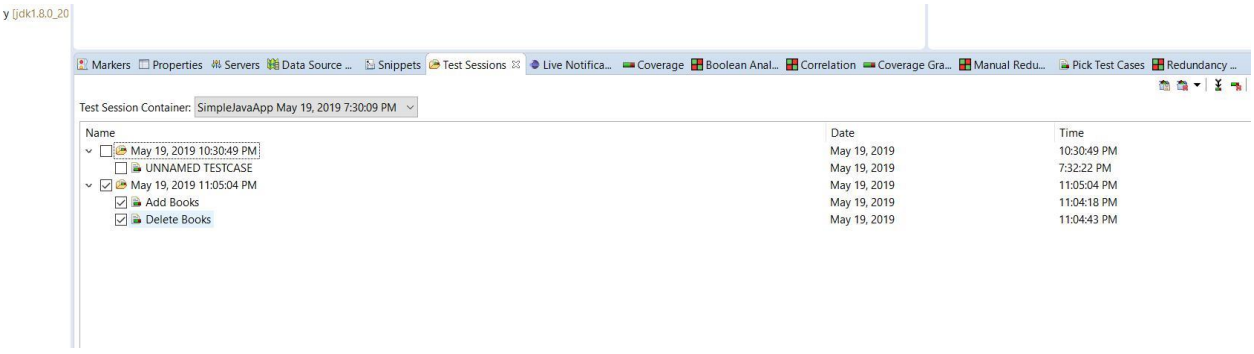
XIV. Here, we need add some parameters to the VM arguments.

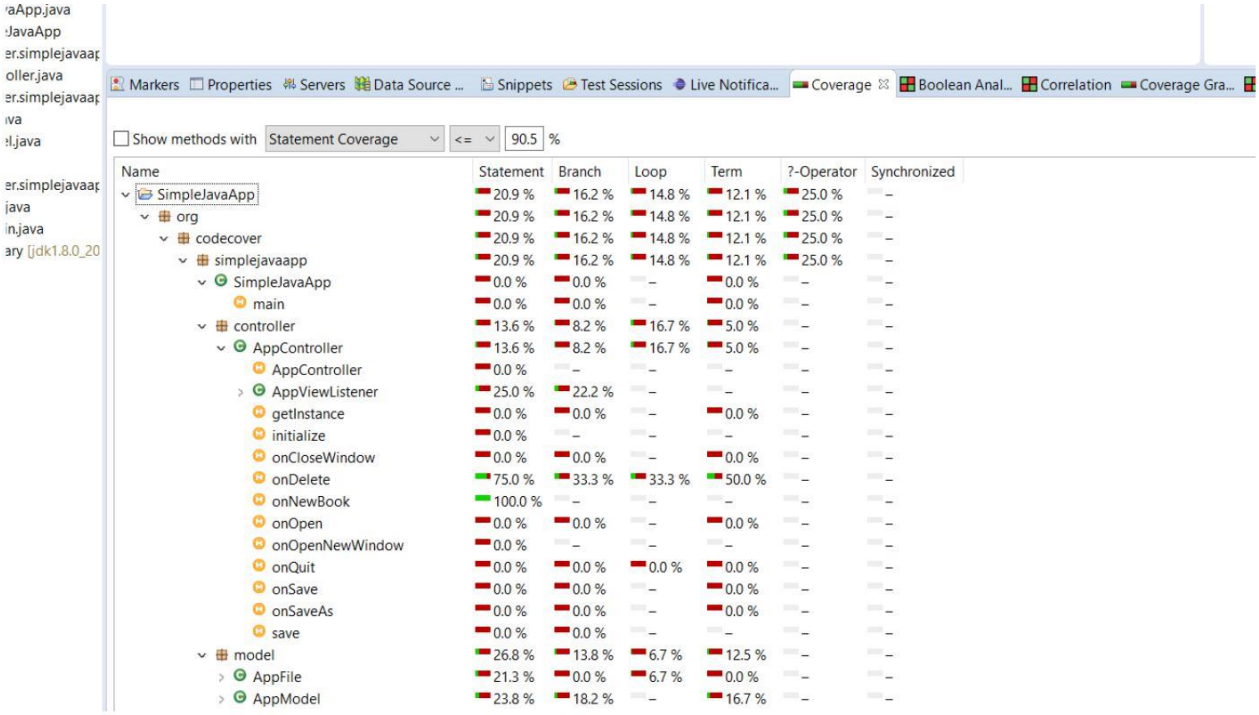


XV. While running the application, we need to go live notification view and add following details and connect.

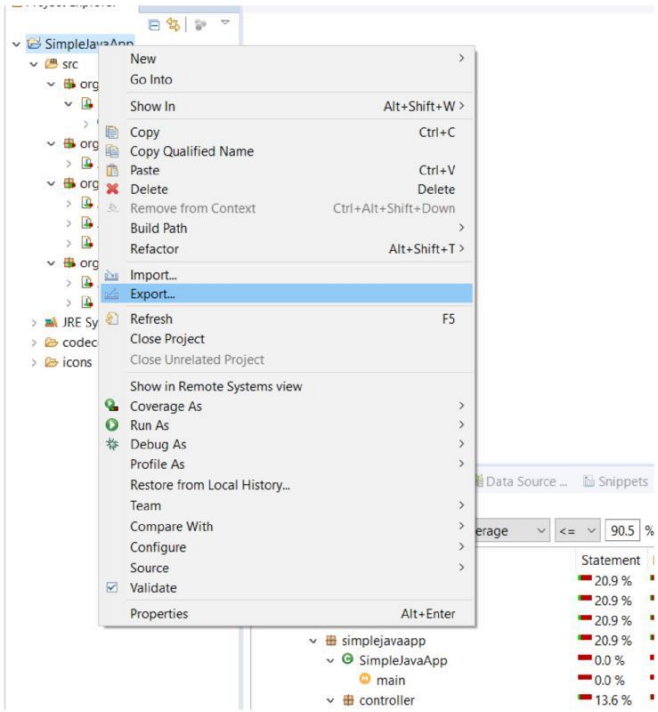


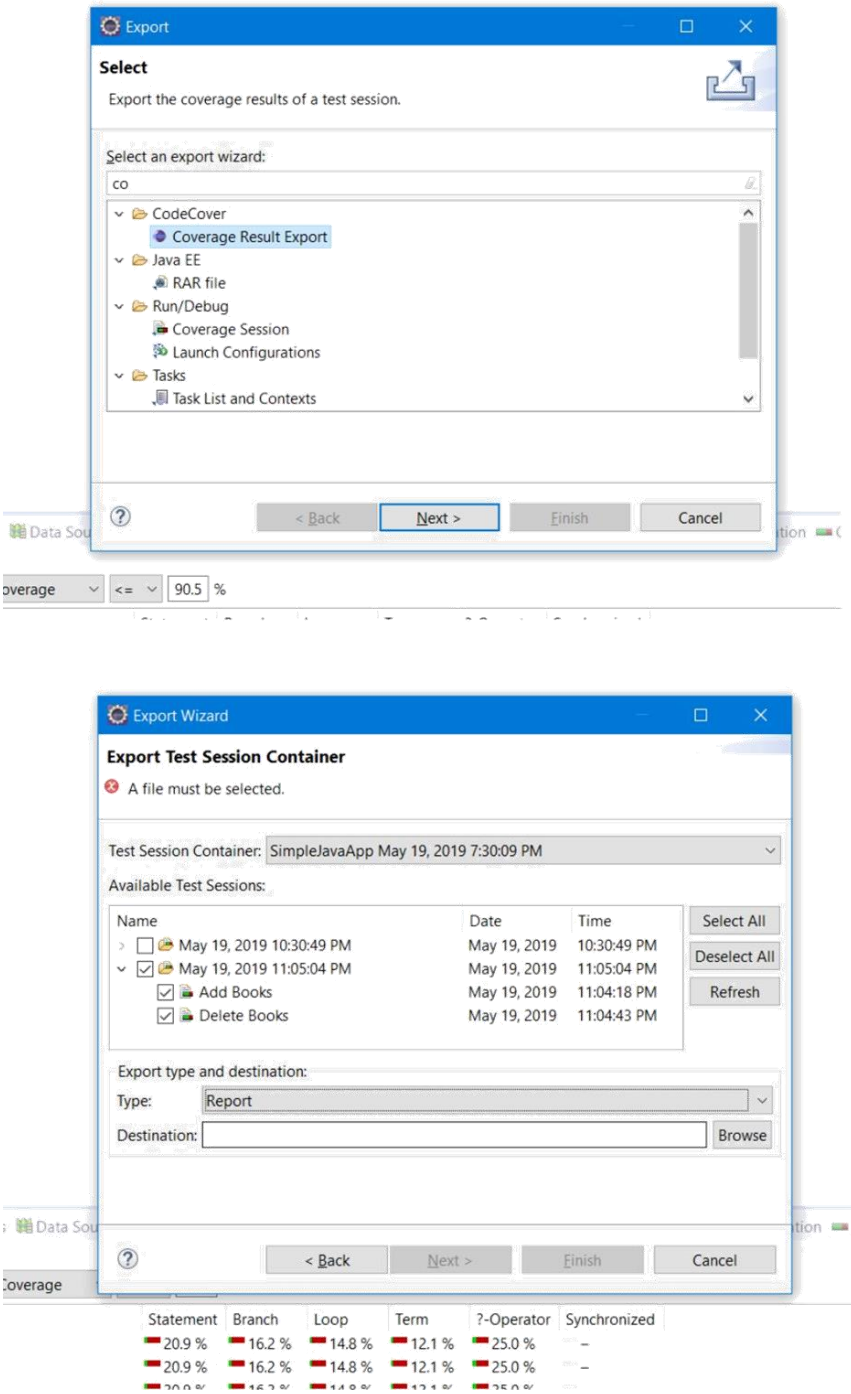
XVI .After, connect we can perform individual test cases by giving name. Enter name, press Start, perform test activities, press End. We can record more test cases repeating the above. To finish recording, use the finish button. Here, I have done two cases called add books and delete books. After pressing finish button, we need to close the running project. In test session view we can see our measured data.





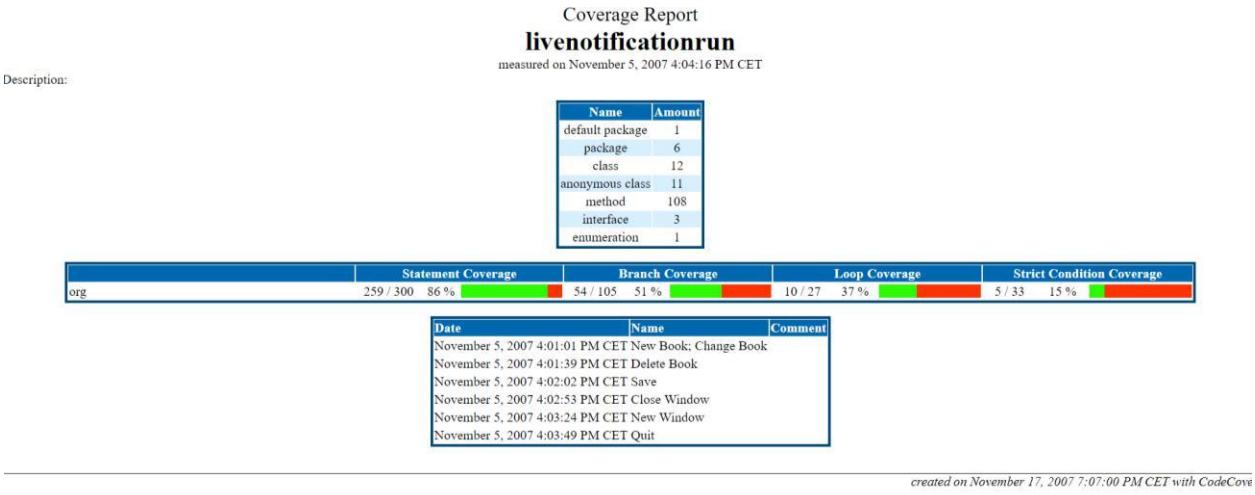
XVII . Also, we can get these details to our customized report. For that, we need a report template xml file. For get report go to export, coverage result export, select test case.




Export
Select
Export the coverage results of a test session.
Select an export wizard:
co
CodeCover
Coverage Result Export
Java EE
RAR file
Run/Debug
Coverage Session
Launch Configurations
Tasks
Task List and Contexts
< Back
Next >
Finish
Cancel
Data Sou
tion
overage
<=
90.5
%

Export Wizard
Export Test Session Container
A file must be selected.
Test Session Container:
SimpleJavaApp May 19, 2019 7:30:09 PM
Available Test Sessions:
Name
Date
Time
May 19, 2019 10:30:49 PM
May 19, 2019 10:30:49 PM
May 19, 2019 11:05:04 PM
May 19, 2019 11:05:04 PM
Add Books
May 19, 2019 11:04:18 PM
Delete Books
May 19, 2019 11:04:43 PM
Select All
Deselect All
Refresh
Export type and destination:
Type:
Report
Destination:
Browse
< Back
Next >
Finish
Cancel
Data Sou
tion
Coverage Gra
overage
Statement
Branch
Loop
Term
?-Operator
Synchronized
20.9 %
16.2 %
14.8 %
12.1 %
25.0 %
20.9 %
16.2 %
14.8 %
12.1 %
25.0 %

XVIII. This is an example report which is not relevant to our scenario.



Coverage Report

livenotificationrun

measured on November 5, 2007 4:04:16 PM CET

Description:

Name	Amount
default package	1
package	6
class	12
anonymous class	11
method	108
interface	3
enumeration	1

	Statement Coverage	Branch Coverage	Loop Coverage	Strict Condition Coverage
org	259 / 300 86 %	54 / 105 51 %	10 / 27 37 %	5 / 33 15 %

Date	Name	Comment
November 5, 2007 4:01:01 PM CET	New Book; Change Book	
November 5, 2007 4:01:39 PM CET	Delete Book	
November 5, 2007 4:02:02 PM CET	Save	
November 5, 2007 4:02:53 PM CET	Close Window	
November 5, 2007 4:03:24 PM CET	New Window	
November 5, 2007 4:03:49 PM CET	Quit	

created on November 17, 2007 7:07:00 PM CET with CodeCove