Configuration of JaCoCo using IntelliJ Idea.

Step 1: Make sure the project is building and test cases are running successfully.

Step 2: Create new Run/Debug configuration. Go to Run > Create/Edit configuration. As shown in Figure 1-1, choose "All in package" as test kind. Also make sure to choose the configuration for JaCoCo run as shown in Figure 1-2.

Step 3: Apply the newly created run profile and run JaCoCo with test coverage. Go to Run > "Run JaCoCo with Coverage".

Step 4: Export the generated coverage results as shown in Figure 1-3.

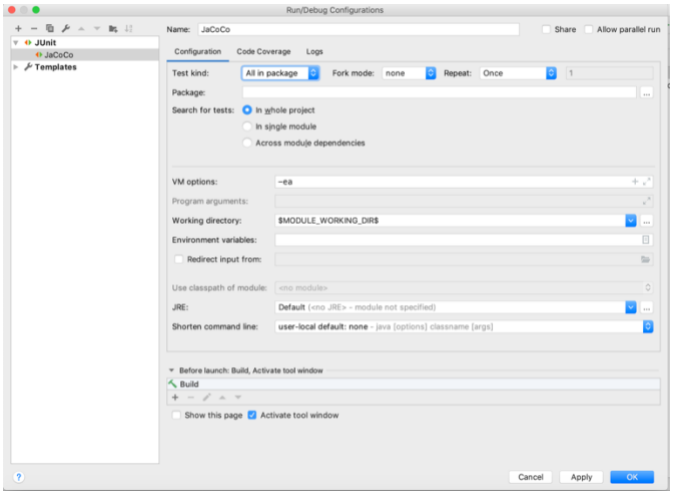


*Figure 1-**Error! No text of specified style in document.**-1: JaCoCo Configuration in IntelliJ Idea IDE*

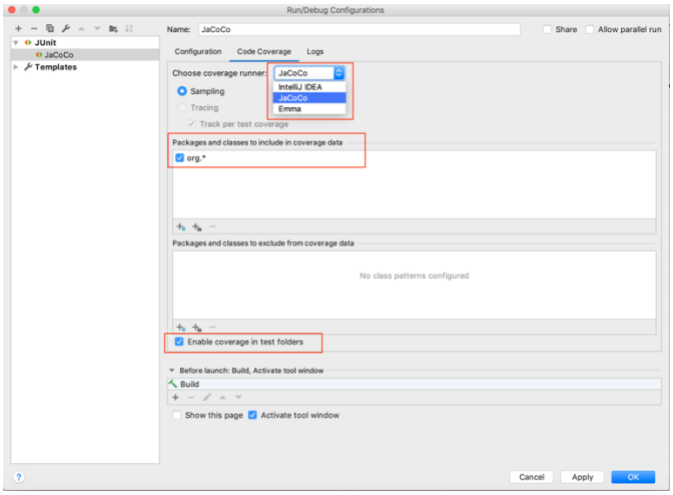


*Figure 1-**Error! No text of specified style in document.**-2: JaCoCo Configuration in IntelliJ Idea IDE*

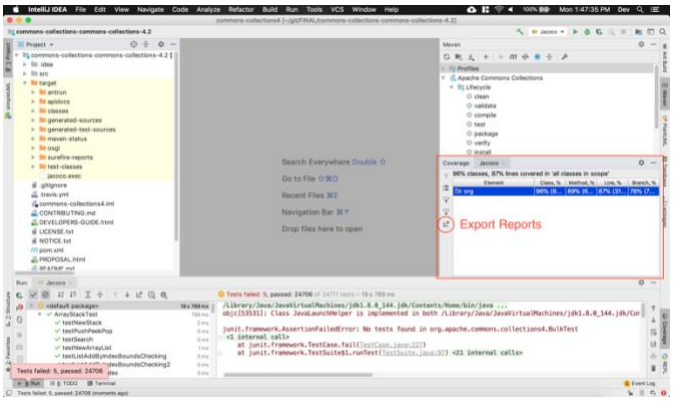


*Figure 1-**Error! No text of specified style in document.**-3: JaCoCo Configuration in IntelliJ Idea IDE*

While running the JaCoCo, sometimes it was producing the incorrect reports due to our mistakes in run configuration, we fixed the, resulting the correct statement and branch coverage reports (Figure 1-4).

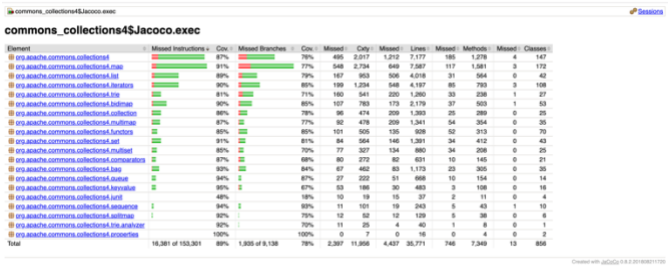


*Figure 1-**Error! No text of specified style in document.**-4 Sample JaCoCo report*

Configuration of PIT - Mutation Testing using IntelliJ Idea

Metric 3 is about test suit effectiveness, and we chose PIT tool (http://pitest.org) to calculate mutation score.IntelliJ Idea has a plugin that simply provides results when we run mutation test on the selected test suits. The tool adds a 'Run configuration' that allows to execute PIT within IDE.

Usage: Run->Edit Configurations->Defaults->Pit Runner or just simply right click on project's parent directory and choose to run PITest from the context menu as shown in Figure 2-1. Note here, it will apply the mutation on all the classes in the project since we are choosing the same option.

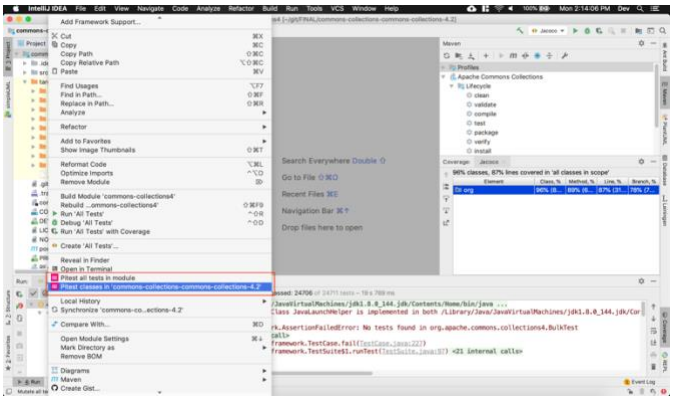
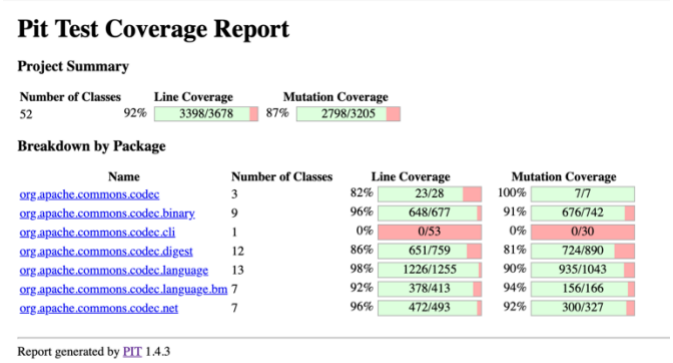


Figure 2-1 PIT Configuration and running

After completion of running mutation testing, it will generate the test reports like shown in Figure 2-2.



Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage		Mutation Coverage	
52	92%	3398/3678	87%	2798/3205

Breakdown by Package

Name	Number of Classes	Line Coverage		Mutation Coverage	
org.apache.commons.codec	3	82%	23/28	100%	7/7
org.apache.commons.codec.binary	9	96%	648/677	91%	676/742
org.apache.commons.codec.cli	1	0%	0/53	0%	0/30
org.apache.commons.codec.digest	12	86%	651/759	81%	724/890
org.apache.commons.codec.language	13	98%	1226/1255	90%	935/1043
org.apache.commons.codec.language.bm	7	92%	378/413	94%	156/166
org.apache.commons.codec.net	7	96%	472/493	92%	300/327

Report generated by PIT 1.4.3

Figure 2-2 PIT test coverage report

Configuration of MetricsReloaded plugin in IntelliJ Idea IDE for static code analysis.

MetricsReloaded plugin is used for static code analysis. We used this plugin to get Halstead Volume in order to calculate Metric 5.

The tool itself is very easy to use, we just need to install the plugin in IntelliJ Idea (https://plugins.jetbrains.com/plugin/93-metricsreloaded), specify the metrics and metrics calculation scope, and run it (Figure 3).

Usage: Choose Analyze > “Calculate Metric” from the menu.

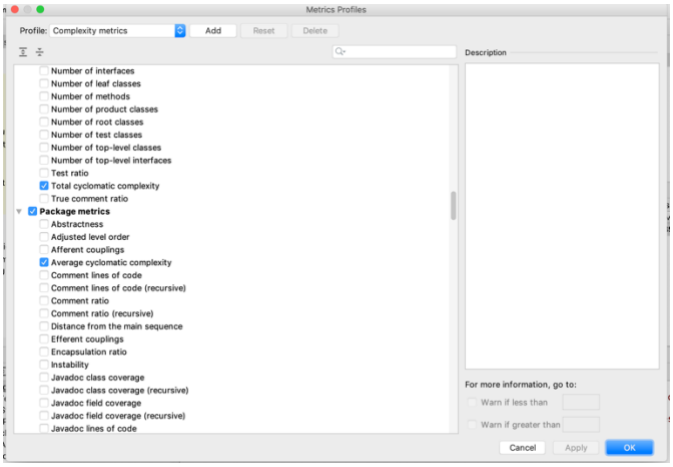


Figure 3 MetricsReloaded plugin configuration and usage