7 JUnit Sampler Script

**Development Environment:**

* Apache JMeter 5.1.1 Version
* Open JDK Runtime Environment 11.0.2

This guide has five sub-sections, namely:

.1 Creating a JUnit test project

.2 Creating jar for the JUnit project

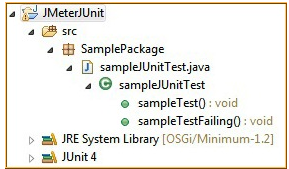
.3 Putting jar in JMeter’s lib/JUnit directory

.4 Running JUnit tests in JMeter

5 Pushing the code to GitHub repositories

**Step .1:** Creating a JUnit test project

* We will create a sample java project having JUnit annotations.
* It contains a test class sampleJUnitTest.java having two test methods.

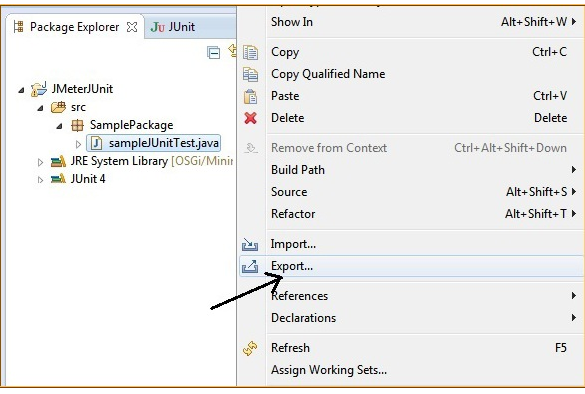


**Note:** For the demo, we have two JUnit tests in sampleJUnitTest.java which are **sampleTestPassing** and **sampleTestFailing**. The test **sampleTestPassing** gets passed when you run it and the test **sampleTestFailing** explicitly fails when it uses the Assert.fail() command.

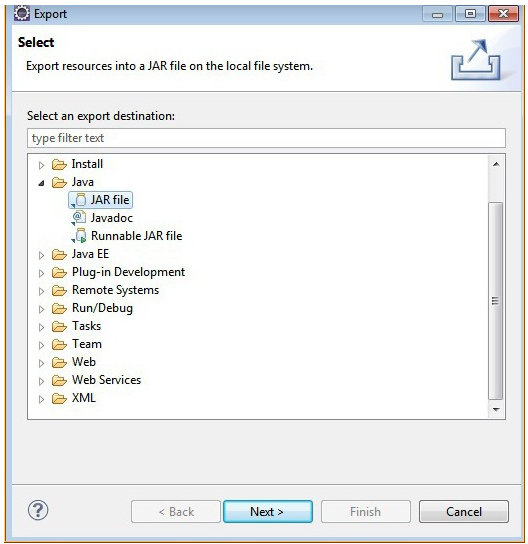


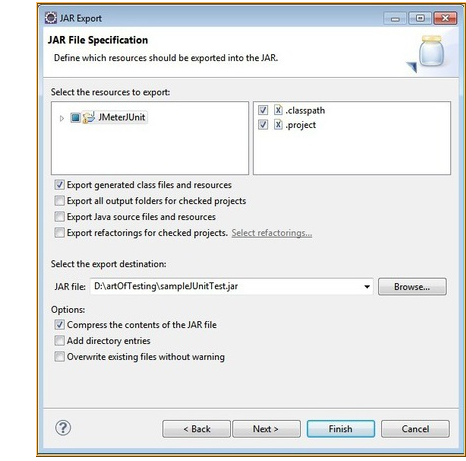
**Step 1.7.2:** Creating jar for the JUnit project

* Right click on the project and click on the Export button.



* Inside java, click on the ‘JAR’ file.



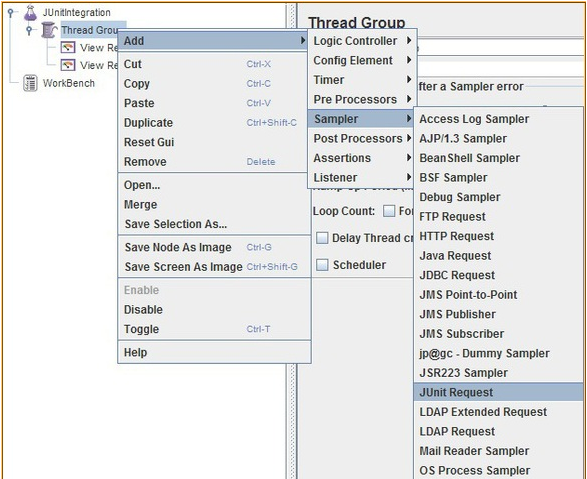
* Select your project and check the resources. Also, provide the path for the generated ‘jar’ file. 

**Step 1.7.3:** Putting jar in JMeter’s lib/JUnit directory

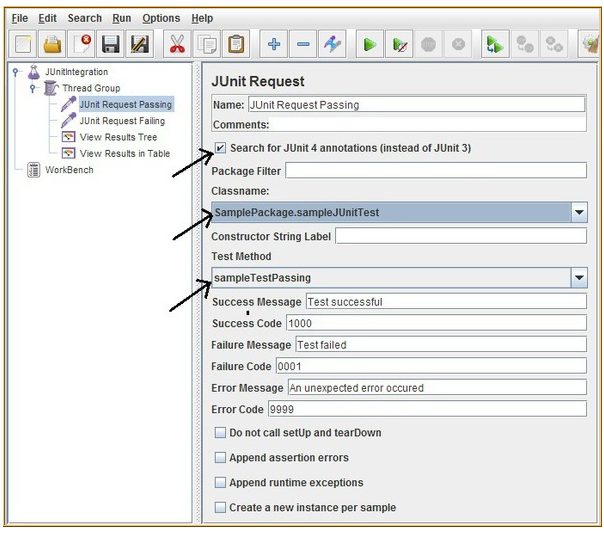
* Put the generated jar file in the JMeter’s lib/JUnit directory and restart JMeter.

**Step 1.7.4:** Running JUnit tests in JMeter

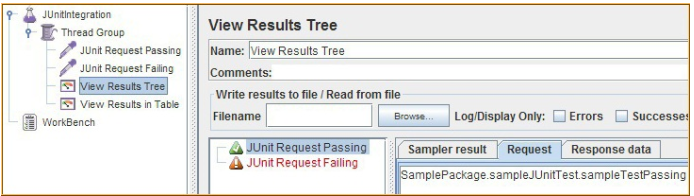
* First, add a “JUnit Request” to a thread group.



* Check the “search for JUnit 4 annotations (instead of JUnit3)” checkbox.
* From the “classname” dropdown, select the JUnit test class created.
* From the “Test Method” dropdown, select the JUnit method/test where you want to load the test.
* Likewise, multiple JUnit requests can be added with each request having a test method.
* In this example, two JUnit requests are added for passing and failing the test.



* Add **Listeners** and run the test.



**Step 1.7.5:** Pushing the code to GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add . 

Commit the changes using the following command:

git commit . -m “Changes have been committed.”

Push the files to the folder you initially created using the following command:

git push -u origin master