CI with Junit in Jenkins

(Check repo CiwithJunitinjenkins )

This section will guide you to:

* Connect Git and GitHub repository with Jenkins along with Junit tests

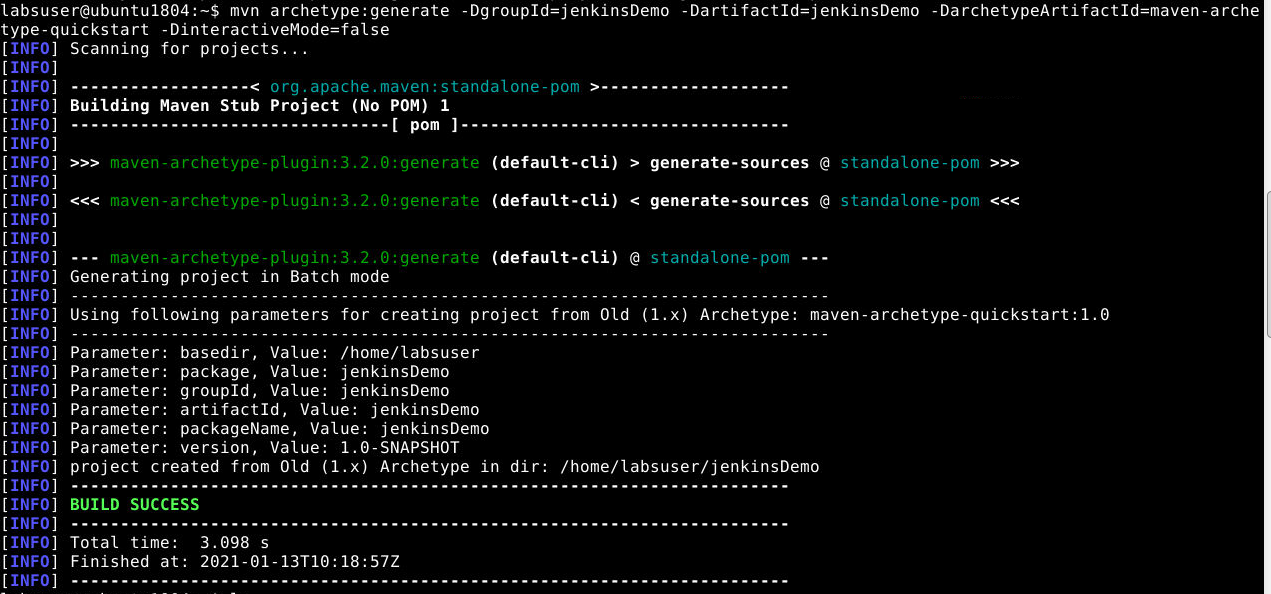
This guide has four subsections, namely:

1. Logging into Jenkins
2. Adding Junit dependencies and classes in Maven project
3. Creating a Jenkins job for Maven project
4. Building the Jenkins job

**Step 1:** Logging into Jenkins

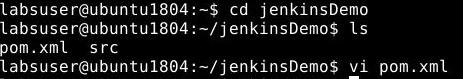
* Open your browser and navigate to **localhost:8081**
* Provide your username and password and click on **Login**

**Step 2:** Adding Junit dependencies and classes in Maven project

* Create a maven project by executing the following command in the terminal of your lab:  
  **mvn archetype:generate -DgroupId=jenkinsDemo -DartifactId=jenkinsDemo -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false**
* Run the below commands to navigate inside the maven project:

**cd /home/labsuser/jenkinsDemo**

**vi pom.xml**

****

* Add the code below versions in pom file

<properties>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

</properties>

* Add the below code in the <dependencies> section of the **pom.xml** file of your Maven project

**<dependency>**

**<groupId>junit</groupId>**

**<artifactId>junit</artifactId>**

**<version>4.12</version>**

**</dependency>**

**<dependency>**

**<groupId>org.seleniumhq.selenium</groupId>**

**<artifactId>selenium-java</artifactId>**

**<version>3.10.0</version>**

**</dependency>**

* Save the file and exit using the command **[esc] shift+:wq**
* Delete the src/main folder using the commands given below:

**cd /home/labsuser/jenkinsDemo/src**

**rm -r main**

* Considering you are in the src folder, navigate to src/test/java, and create a file JenkinsDemo.javausing the following commands:

**cd /home/labsuser/jenkinsDemo/src/test/java**

**vi JenkinsDemo.java**

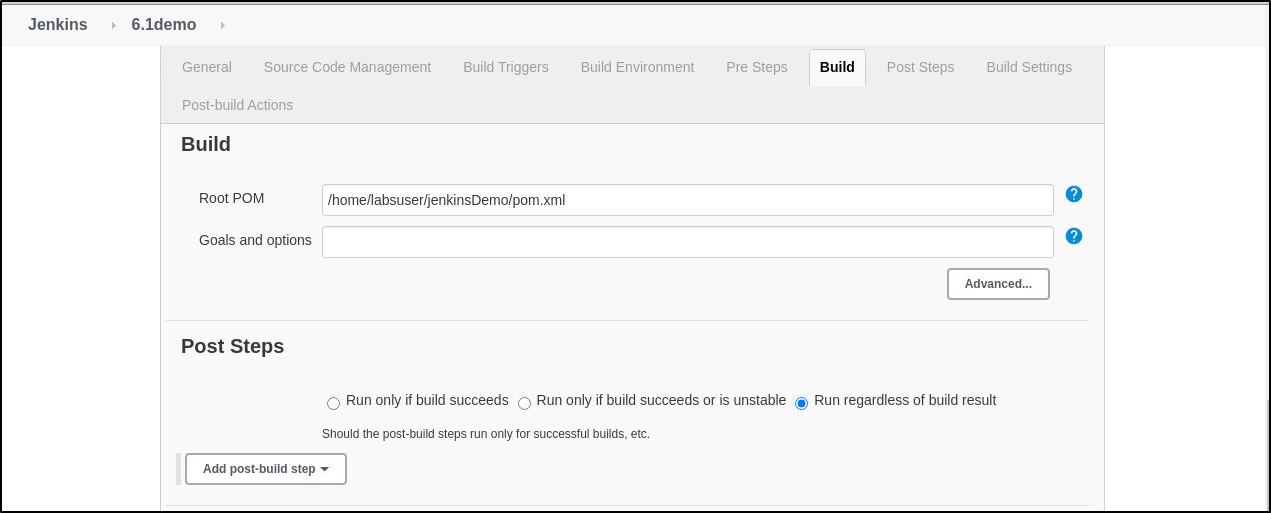
* Add the following code in **JenkinsDemo.java:**

**import org.junit.After;  
import org.junit.Assert;  
import org.junit.Before;  
import org.junit.Ignore;  
import org.junit.Test;  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;  
  
/\*\*  
 \* Created by   
 \* Sample junit test code to integrate by  
 \*/  
public class JenkinsDemo  
{  
 private static String Base\_Url = "https://www.facebook.com";  
 private WebDriver driver;  
  
 @Before  
 public void setUp()  
 {  
 driver = new ChromeDriver();  
 driver.get(Base\_Url);  
 }  
  
 @After  
 public void after()  
 {  
 driver.quit();  
 }  
  
 @Test  
 public void testCasePassed()  
 {  
 Assert.assertTrue(driver.findElement(By.xpath("//form[@id='login\_form']")).isDisplayed());  
 }  
  
 @Test  
 public void testCaseFailed()  
 {  
 Assert.assertTrue(driver.findElement(By.xpath("//form[@id='failed case']")).isDisplayed());  
 }  
  
 @Ignore  
 @Test  
 public void testCaseIgnored()  
 {  
 Assert.assertTrue(driver.findElement(By.xpath("//form[@id='ignored case']")).isDisplayed());  
 }  
}**

* Save the file and exit using the command **[esc] shift+:wq**
* Your directory structure should only have these files.
* Run the following command to delete any unnecessary files or folder structures:  
  **rm -r jenkinsDemo**

**Step 3:** Creating a Jenkins job for Maven project

* To create a new job in Jenkins, open the Jenkins dashboard with your Jenkins URL.   
  For example, http://localhost:8081/
* Click on **New Item**. Enter the item name, select **Maven Project,** and click **OK**.
* Once you click **OK,**the page will be redirected to its project form.
* In the **Build** section of your job, for Root POM give the path of the pom.xml in your local system as shown:   
  **/*home/labsuser/jenkinsDemo/pom.xml***



**Step 4:** Building the Jenkins job

* Build the job and check the status on the console for the test cases executed.

