6.1 CI with Junit in Jenkins

This section will guide you to:

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

This guide has four subsections, namely:

6.1.1 Login to Jenkins

6.1.2 Add Junit dependencies and classes in Maven project

6.1.3 Create Jenkins job for Maven

6.1.4 Push code to GitHub repositories

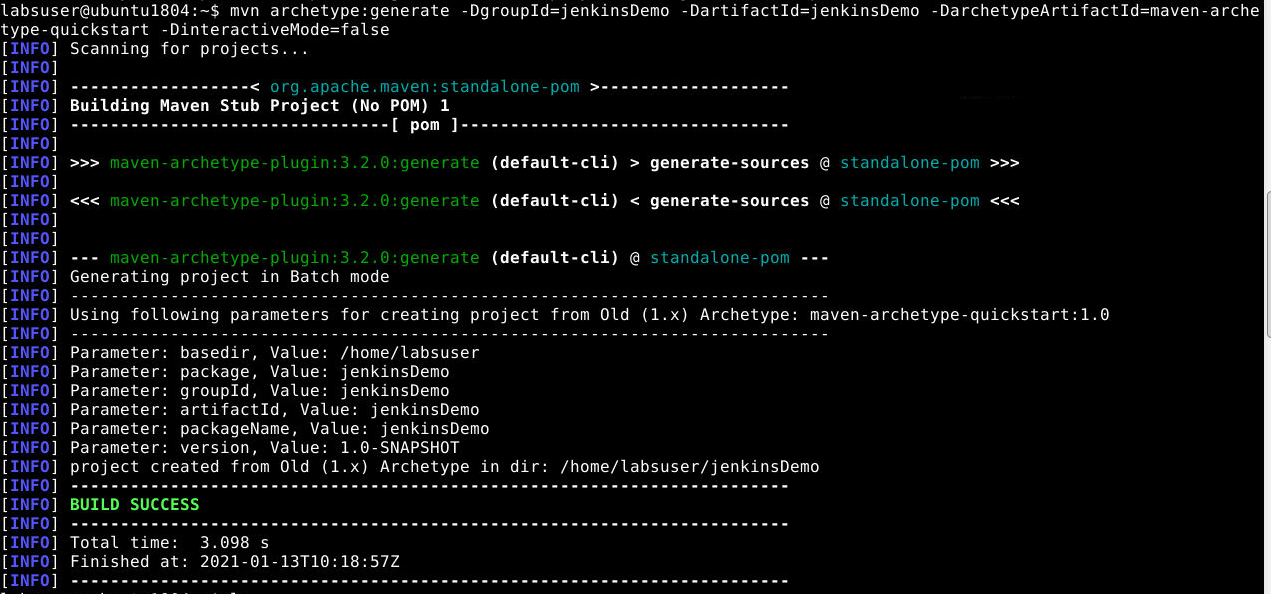
**Step 6.1.1:** Login to Jenkins

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

**Step 6.1.2:** Add Junit dependencies and classes in Maven project.

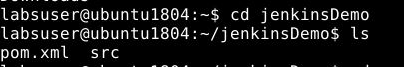
Create a maven project by executing the following command

**mvn archetype:generate -DgroupId=jenkinsDemo -DartifactId=jenkinsDemo -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false**



Run the below command to navigate inside the maven project.

**cd jenkinsDemo**



* Add the below code in the **pom.xml** file of your Maven project.

|  |  |
| --- | --- |
|  | |
| **<dependencies>** | |
|  | **<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>** |
|  | **</dependencies>** |
|  |  |

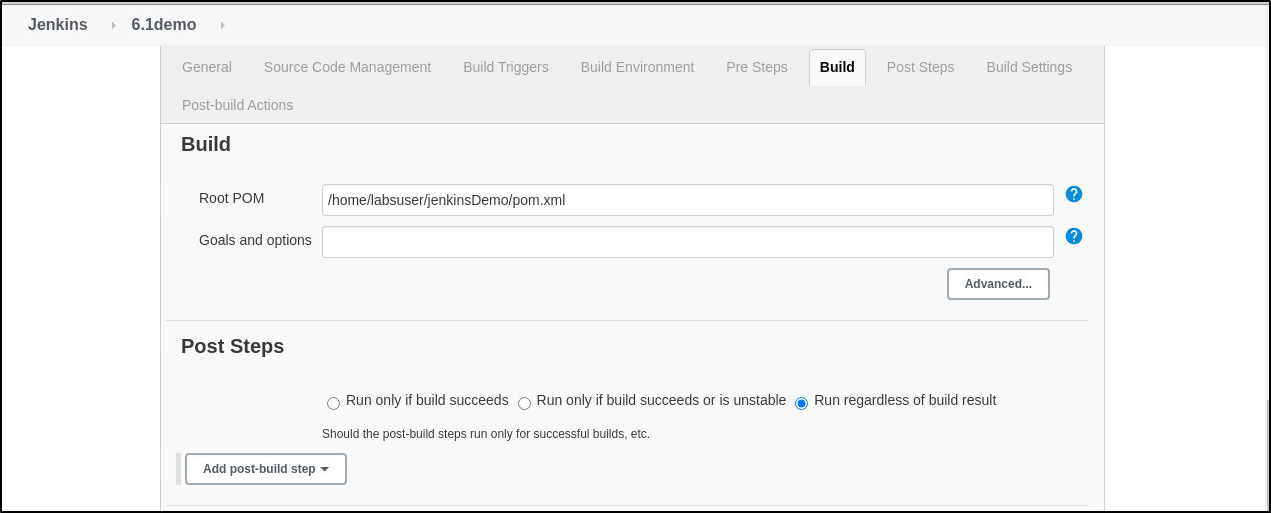
* Delete the **src/main** folder and Navigate to **src/test/java** and create a file **JenkinsDemo.java**
* Add the following code in **JenkinsDemo.java**

**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());  
 }  
}**

Your directory structure should only have these files. Delete any unnecessary files or folder structures.

**Step 6.1.3:** Create Jenkins job for Maven

* To create a new job in Jenkins, open the Jenkins dashboard with your Jenkins URL. For example, http://localhost:8081/.
* Click on **Create New Job**. Enter the item name, select **Maven Project,** and click **OK**.
* Once you click **OK,**the page will be redirected to its project form. Here, you will need to enter the project information.
* In the **Build** section of your job, give the path of the pom.xml in your local system. ie. /*home/labuser/jenkinsDemo/pom.xml*



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

**Step 6.1.4:** Push the code to your 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**