# Integration of Jenkins with Selenium WebDriver: Step-by-Step Tutorial

Reference - http://www.softwaretestinghelp.com/integration-of-jenkins-with-selenium-webdriver/

**Step by step guide to Setup and Configure Jenkins with Selenium:**

Jenkins is an open source tool written in Java. It provides continuous delivery and continuous integration service for software development. It automates your manual task of code deployment process from development box – QA – Stage – Production.

Jenkins supports many plugins which you can integrate such as Git, SVN, build pipeline and many more.

The basic functionality of Jenkins is to execute a predefined list of steps on the basis of time and some events.

**For example,** when you want to base the execution on time you can run a job every 20 minutes or if you want to base it on an event you can do so after a new commit in a Git repository.

A Step by Step Guide to
Integrate Jenkins with Selenium WebDriver
Jenkins
Selenium
© www.SoftwareTestingHelp.com

**In this article we covered the below points:**

- Jenkins usage and integration with selenium
- Creating a batch file and using it Jenkins
- Scheduling Jenkins job and added email notification
- And running selenium script from command line

**Advantages of using Jenkins are:**

- It is a cross-platform and can be used on Windows, Linux, Mac OS, and Solaris environments
- It is a free and open source tool
- Widely used and well documented
- Integration with wide variety of tool and technologies

**Apart from Jenkins, we have many more tools in the market such as:**

- Anthill
- Bamboo
- Cruise Control
- Team City and many more.

**What You Will Learn:** [show]

**Jenkins usage and integration with selenium**

**Follow the below step-by-step procedure to use Jenkins with Selenium**

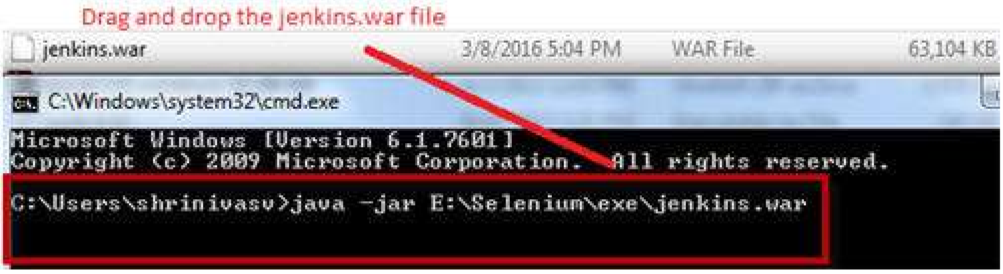
**Step #1:**

Download Jenkins from the official website of Jenkins – **Jenkins**. Download the latest .war file. Jenkins can be started via the command line or can run in a web application server.

Refer to the below steps for the execution through the command line:

**1)** Open the command prompt and type ***java –jar*** and enter path of .war file

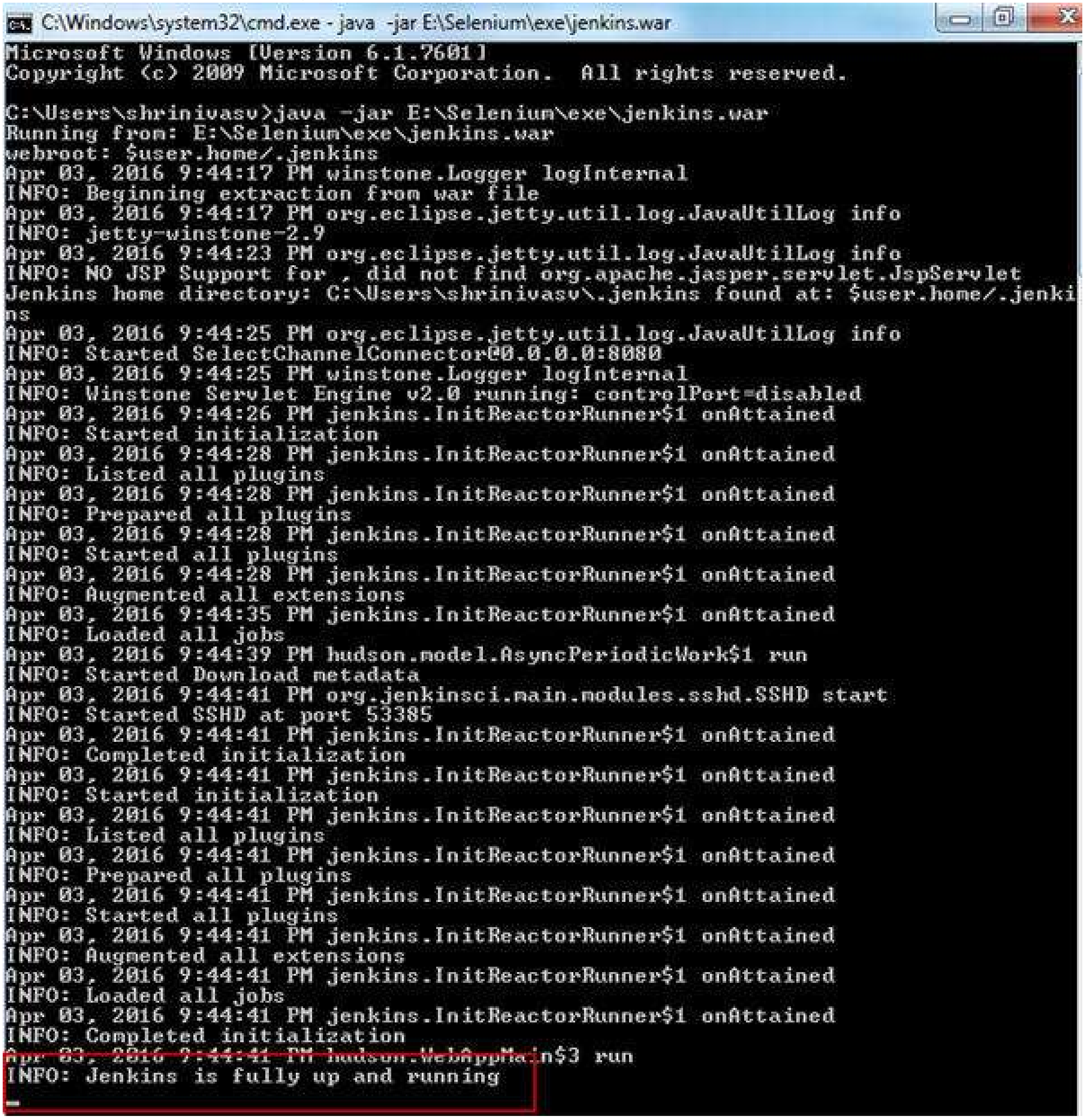
**(Note: Click on any image for enlarged view)**


Drag and drop the jenkins.war file
jenkins.war
3/8/2016 5:04 PM
WAR File
63,104 KB
C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\shrinivasv>java -jar E:\Selenium\exe\jenkins.war
```

**2)** Press enter and check if your Jenkins.war file started to run and check the status information on the command prompt console.

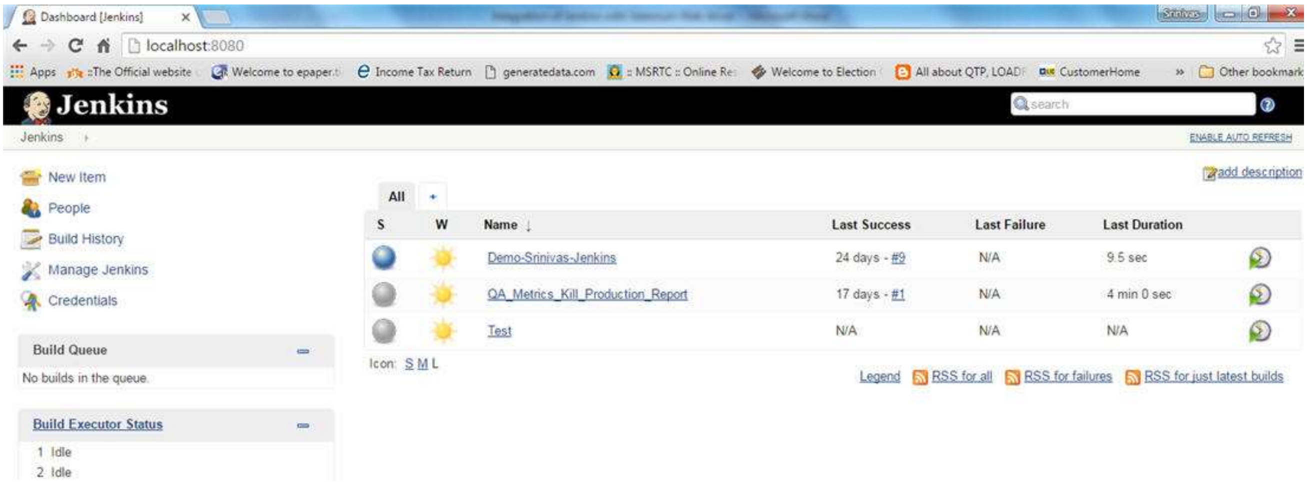
It should show – **Jenkins is fully up and running**


C:\Windows\system32\cmd.exe - java  -jar E:\Selenium\exe\jenkins.war

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\shrinivasv>java -jar E:\Selenium\exe\jenkins.war
Running from: E:\Selenium\exe\jenkins.war
webroot: $user.home/.jenkins
Apr 03, 2016 9:44:17 PM winstone.Logger logInternal
INFO: Beginning extraction from war file
Apr 03, 2016 9:44:17 PM org.eclipse.jetty.util.log.JavaUtilLog info
INFO: jetty-winstone-2.9
Apr 03, 2016 9:44:23 PM org.eclipse.jetty.util.log.JavaUtilLog info
INFO: NO JSP Support for , did not find org.apache.jasper.servlet.JspServlet
Jenkins home directory: C:\Users\shrinivasv\.jenkins found at: $user.home/.jenki
ns
Apr 03, 2016 9:44:25 PM org.eclipse.jetty.util.log.JavaUtilLog info
INFO: Started SelectChannelConnector@0.0.0.0:8080
Apr 03, 2016 9:44:25 PM winstone.Logger logInternal
INFO: Winstone Servlet Engine v2.0 running: controlPort=disabled
Apr 03, 2016 9:44:26 PM jenkins.InitReactorRunner$1 onAttained
INFO: Started initialization
Apr 03, 2016 9:44:28 PM jenkins.InitReactorRunner$1 onAttained
INFO: Listed all plugins
Apr 03, 2016 9:44:28 PM jenkins.InitReactorRunner$1 onAttained
INFO: Prepared all plugins
Apr 03, 2016 9:44:28 PM jenkins.InitReactorRunner$1 onAttained
INFO: Started all plugins
Apr 03, 2016 9:44:28 PM jenkins.InitReactorRunner$1 onAttained
INFO: Augmented all extensions
Apr 03, 2016 9:44:35 PM jenkins.InitReactorRunner$1 onAttained
INFO: Loaded all jobs
Apr 03, 2016 9:44:39 PM hudson.model.AsyncPeriodicWork$1 run
INFO: Started Download metadata
Apr 03, 2016 9:44:41 PM org.jenkinsci.main.modules.sshd.SSHD start
INFO: Started SSHD at port 53385
Apr 03, 2016 9:44:41 PM jenkins.InitReactorRunner$1 onAttained
INFO: Completed initialization
Apr 03, 2016 9:44:41 PM jenkins.InitReactorRunner$1 onAttained
INFO: Started initialization
Apr 03, 2016 9:44:41 PM jenkins.InitReactorRunner$1 onAttained
INFO: Listed all plugins
Apr 03, 2016 9:44:41 PM jenkins.InitReactorRunner$1 onAttained
INFO: Prepared all plugins
Apr 03, 2016 9:44:41 PM jenkins.InitReactorRunner$1 onAttained
INFO: Started all plugins
Apr 03, 2016 9:44:41 PM jenkins.InitReactorRunner$1 onAttained
INFO: Augmented all extensions
Apr 03, 2016 9:44:41 PM jenkins.InitReactorRunner$1 onAttained
INFO: Loaded all jobs
Apr 03, 2016 9:44:41 PM jenkins.InitReactorRunner$1 onAttained
INFO: Completed initialization
Apr 03, 2016 9:44:41 PM hudson.WebAppMain$3 run
INFO: Jenkins is fully up and running
_
```

**3)** Now check whether your Jenkins is ready to use; by default, it uses port 8080.

Type "***http://localhost:8080***" in the browser and press enter. It will show you Jenkins UI.



| S | W | Name ↓ | Last Success | Last Failure | Last Duration |
|---|---|---|---|---|---|
| ● | ☀ | Demo-Srinivas-Jenkins | 24 days - #9 | N/A | 9.5 sec |
| ● | ☀ | QA_Metrics_Kill_Production_Report | 17 days - #1 | N/A | 4 min 0 sec |
| ● | ☀ | Test | N/A | N/A | N/A |

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

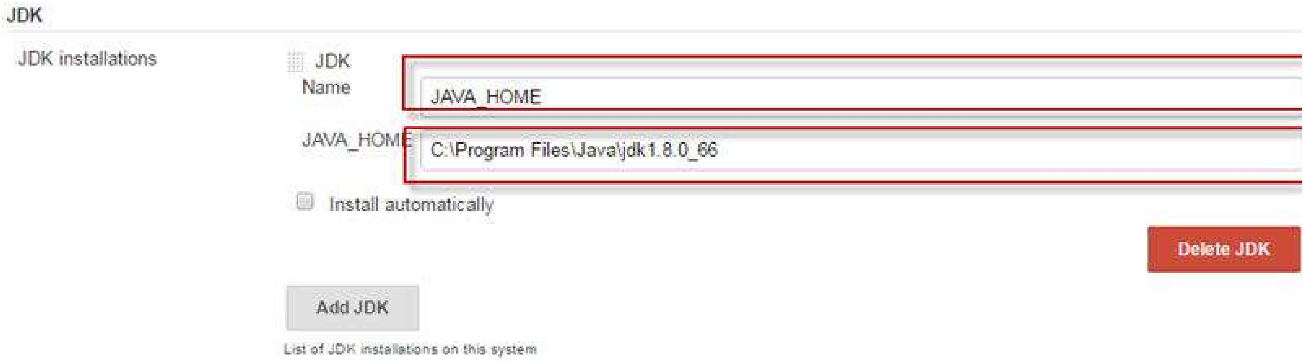
It will load the Jenkins dashboard empty by default. I created some Jenkins job in the above screenshot as an example and hence, it did not load empty.

**Step #2:**

To use Selenium with Jenkins you need to configure Jenkins with Selenium.

**Follow the below steps:**

- Go to Jenkins dashboard
- Click on manage Jenkins
- Click on configure Jenkins
- Click on JDK installation – In JDK name section enter the name, under Java Home section – give your java path


JDK

JDK installations

JDK Name: JAVA_HOME

JAVA_HOME: C:\Program Files\Java\jdk1.8.0_66

□ Install automatically

List of JDK installations on this system

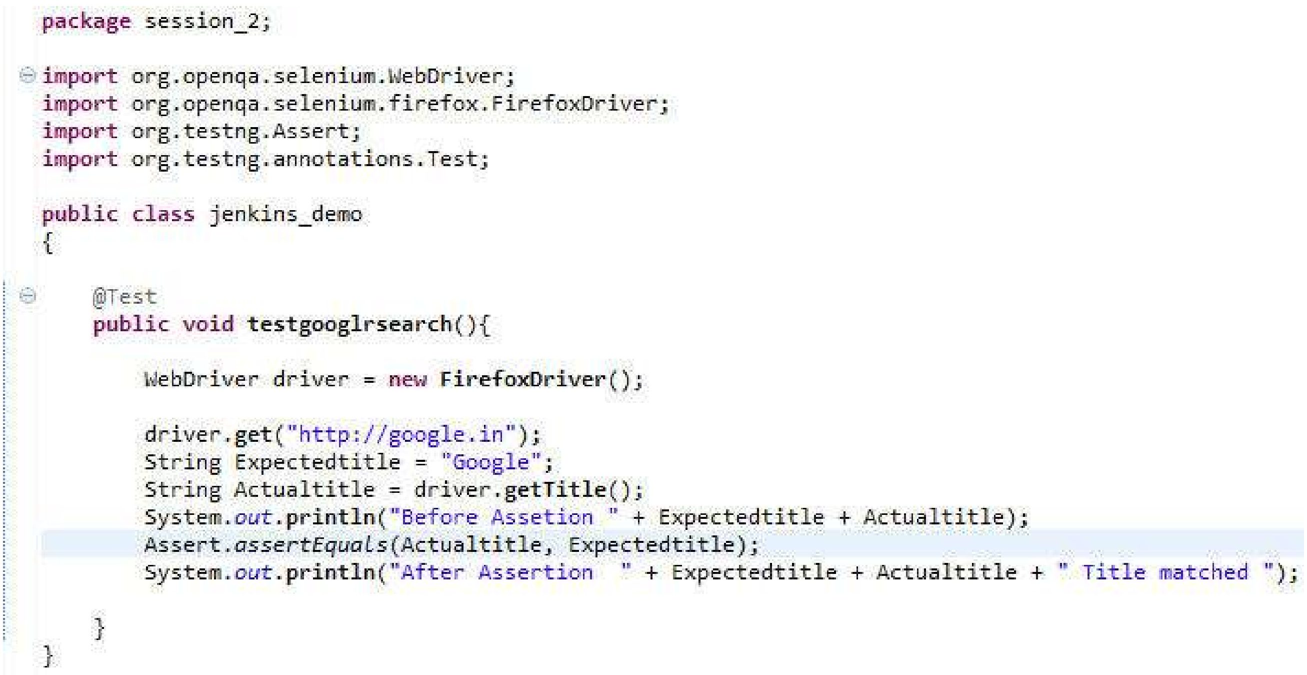
The radio button, **Install automatically** is checked by default. You need to uncheck it because it will automatically update with the new Java version and there might be a possibility that Selenium doesn't support the new Java version. It is better to uncheck it. Now click on apply and save.

Your Jenkins is configured with Selenium and is now ready to be used with Selenium. Both Jenkins and Selenium code is written in Java. Hence, if you give the Java path then internally it will communicate and process your job.

**Step #3:**

Now, create a Selenium script and a TestNG XML file. This TestNG XML file will be required for creating a batch file and we will use that batch file in Jenkins. Refer below TestNG code:

**Refer below TestNG code:**



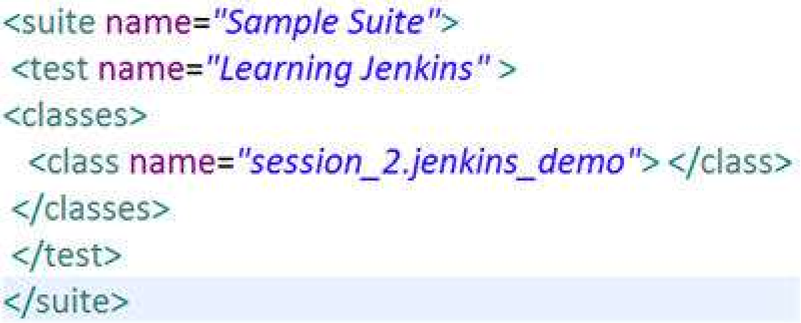
```
package session_2;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.Assert;
import org.testng.annotations.Test;

public class jenkins_demo
{

    @Test
    public void testgooglrsearch(){

        WebDriver driver = new FirefoxDriver();

        driver.get("http://google.in");
        String Expectedtitle = "Google";
        String Actualtitle = driver.getTitle();
        System.out.println("Before Assetion " + Expectedtitle + Actualtitle);
        Assert.assertEquals(Actualtitle, Expectedtitle);
        System.out.println("After Assertion  " + Expectedtitle + Actualtitle + " Title matched ");

    }
}
```

```
package session_2;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.Assert;
import org.testng.annotations.Test;

public class jenkins_demo
{

@Test
publicvoidtestgooglrsearch(){

WebDriver driver = newFirefoxDriver();
//it will open the goggle page
driver.get("http://google.in");
//we expect the title “Google “ should be present
String Expectedtitle = "Google";
//it will fetch the actual title
String Actualtitle = driver.getTitle();
System.out.println("Before Assetion " + Expectedtitle + Actualtitle);
//it will compare actual title and expected title
Assert.assertEquals(Actualtitle, Expectedtitle);
//print out the result
System.out.println("After Assertion " + Expectedtitle + Actualtitle + " Title matched ");
 }
}
```

**Output**: Before Assertion GoogleGoogle

After Assertion, GoogleGoogle Title matched
PASSED: testgooglrsearch

**Create a TestNG xml file, refer below code:**

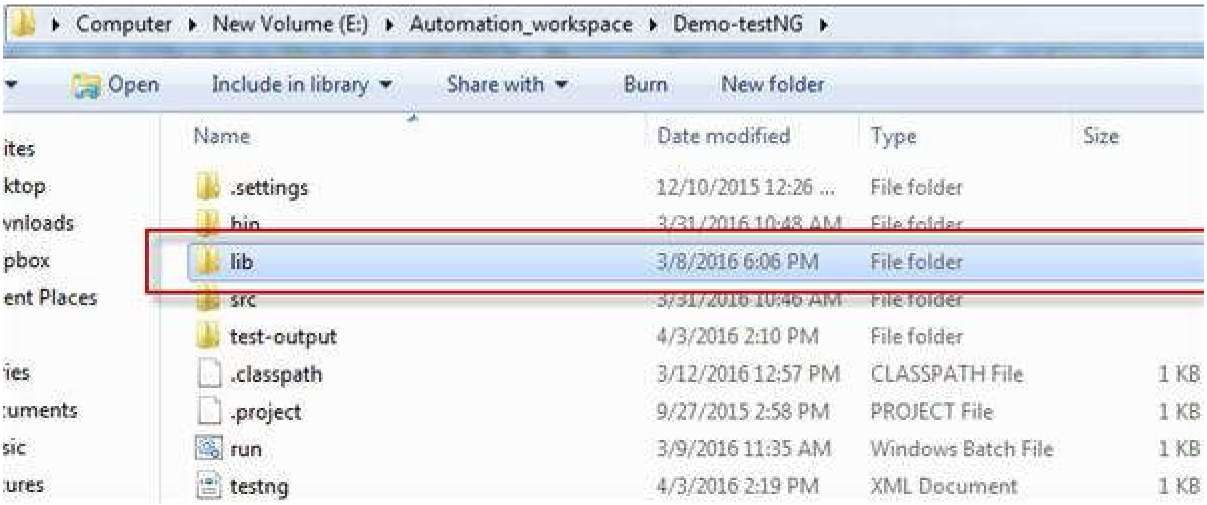


```
<suite name="Sample Suite">
 <test name="Learning Jenkins" >
<classes>
  <class name="session_2.jenkins_demo"> </class>
 </classes>
 </test>
</suite>
```

**Step #4:**

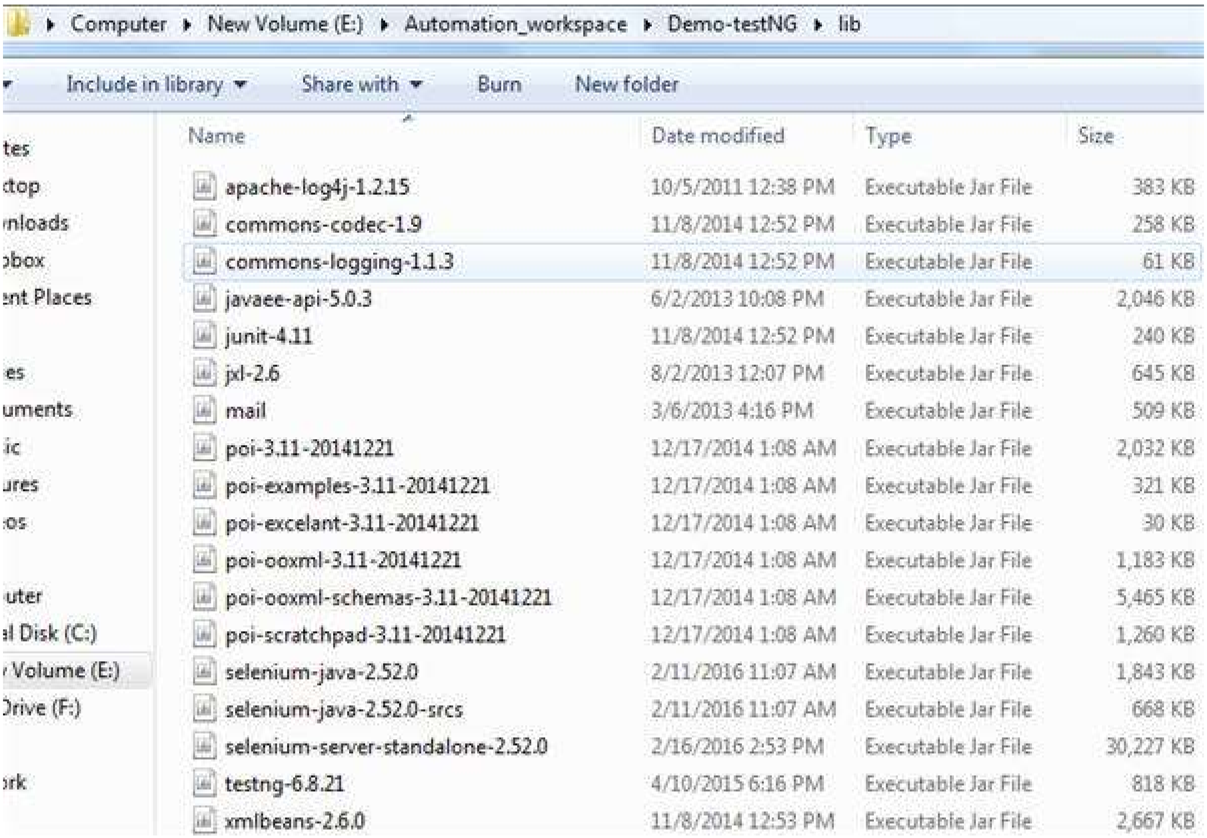
Go to your project root directory and create a library folder.

**Refer the screenshot below:**



| Name | Date modified | Type | Size |
|---|---|---|---|
| .settings | 12/10/2015 12:26 ... | File folder | |
| bin | 3/31/2016 10:48 AM | File folder | |
| lib | 3/8/2016 6:06 PM | File folder | |
| src | 3/31/2016 10:46 AM | File folder | |
| test-output | 4/3/2016 2:10 PM | File folder | |
| .classpath | 3/12/2016 12:57 PM | CLASSPATH File | 1 KB |
| .project | 9/27/2015 2:58 PM | PROJECT File | 1 KB |
| run | 3/9/2016 11:35 AM | Windows Batch File | 1 KB |
| testng | 4/3/2016 2:19 PM | XML Document | 1 KB |

Now, add all your jar files which are required for running your Selenium script:



| Name | Date modified | Type | Size |
|---|---|---|---|
| apache-log4j-1.2.15 | 10/5/2011 12:38 PM | Executable Jar File | 383 KB |
| commons-codec-1.9 | 11/8/2014 12:52 PM | Executable Jar File | 258 KB |
| commons-logging-1.1.3 | 11/8/2014 12:52 PM | Executable Jar File | 61 KB |
| javaee-api-5.0.3 | 6/2/2013 10:08 PM | Executable Jar File | 2,046 KB |
| junit-4.11 | 11/8/2014 12:52 PM | Executable Jar File | 240 KB |
| jxl-2.6 | 8/2/2013 12:07 PM | Executable Jar File | 645 KB |
| mail | 3/6/2013 4:16 PM | Executable Jar File | 509 KB |
| poi-3.11-20141221 | 12/17/2014 1:08 AM | Executable Jar File | 2,032 KB |
| poi-examples-3.11-20141221 | 12/17/2014 1:08 AM | Executable Jar File | 321 KB |
| poi-excelant-3.11-20141221 | 12/17/2014 1:08 AM | Executable Jar File | 30 KB |
| poi-ooxml-3.11-20141221 | 12/17/2014 1:08 AM | Executable Jar File | 1,183 KB |
| poi-ooxml-schemas-3.11-20141221 | 12/17/2014 1:08 AM | Executable Jar File | 5,465 KB |
| poi-scratchpad-3.11-20141221 | 12/17/2014 1:08 AM | Executable Jar File | 1,260 KB |
| selenium-java-2.52.0 | 2/11/2016 11:07 AM | Executable Jar File | 1,843 KB |
| selenium-java-2.52.0-srcs | 2/11/2016 11:07 AM | Executable Jar File | 668 KB |
| selenium-server-standalone-2.52.0 | 2/16/2016 2:53 PM | Executable Jar File | 30,227 KB |
| testng-6.8.21 | 4/10/2015 6:16 PM | Executable Jar File | 818 KB |
| xmlbeans-2.6.0 | 11/8/2014 12:53 PM | Executable Jar File | 2,667 KB |

**Step #5:**

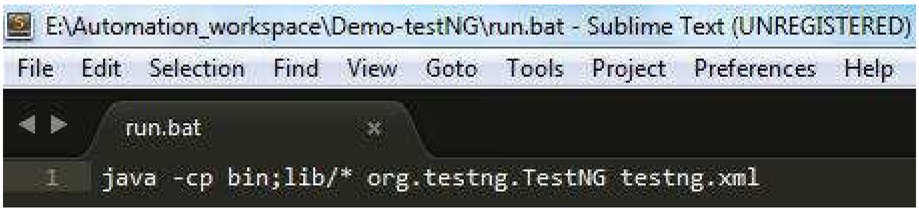
**Creating a batch file and using it in Jenkins**

**Create a batch file by following the below steps:**

**1)** Open the notepad and type-: *Java –cp* bin;lib/* *org.testng.TestNG testng.xml*

By doing this, **Java –cp** will compile and execute .class file which is located at **bin** directory and all our executable jar file are located at **lib** directory and we are using TestNG framework so specify **org.testng.TestNG.** Also, specify the **name of xml** file which will trigger the expected TestNG script.

**2)** Save the file with .bat extension and check the type of file. It should be “windows batch file”. To cross check whether the batch file is created properly, double click the batch file and it will execute the code. Refer the below code of batch file:


E:\Automation_workspace\Demo-testNG\run.bat - Sublime Text (UNREGISTERED)
File
Edit
Selection
Find
View
Goto
Tools
Project
Preferences
Help
run.bat

```
java -cp bin;lib/* org.testng.TestNG testng.xml
```

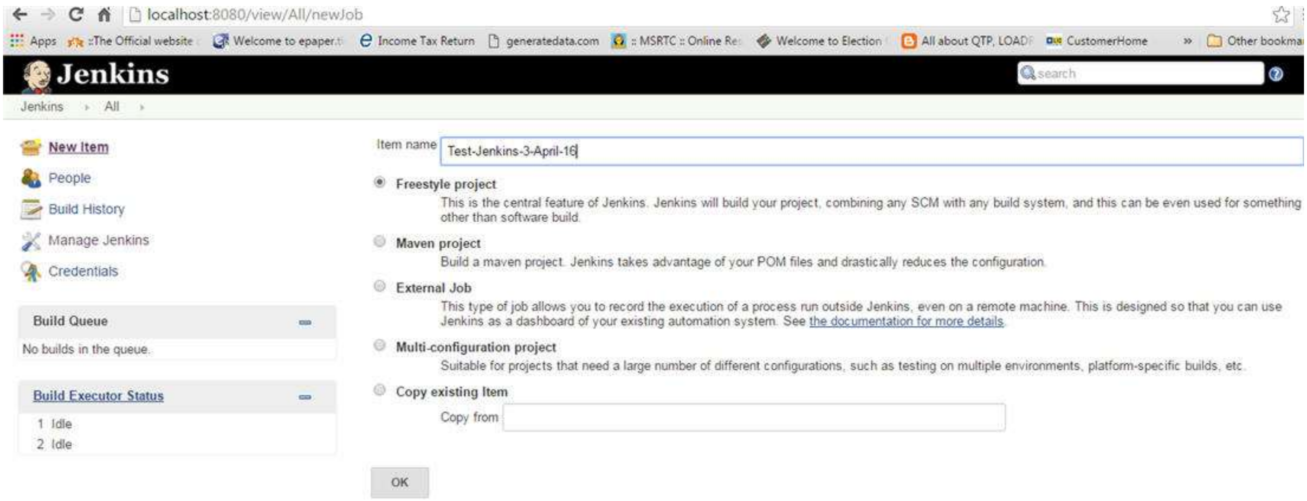
**Step #6:**

Next, we need to add a batch file in Jenkins.

**For adding the batch file follow the below steps:**

**1)** Go to the Jenkins dashboard, create a new job in Jenkins

**2)** Click on new item and enter the item name and check the freestyle project radio button



Item name: Test-Jenkins-3-April-16

● Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

○ Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

○ External Job
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system. See the documentation for more details.

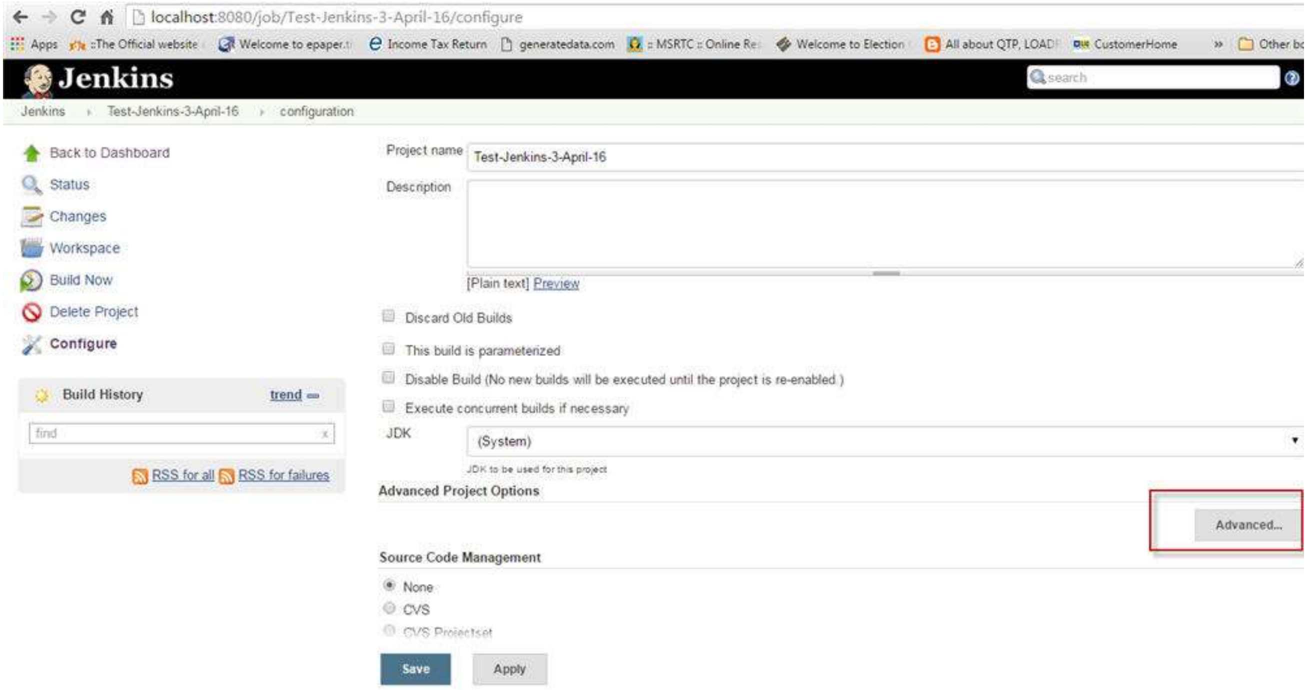
○ Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

○ Copy existing Item
Copy from

Build Queue
No builds in the queue.

Build Executor Status
1 Idle
2 Idle

**3)** Click Advance options



Project name: Test-Jenkins-3-April-16

Description

[Plain text] Preview

□ Discard Old Builds

□ This build is parameterized

□ Disable Build (No new builds will be executed until the project is re-enabled.)

□ Execute concurrent builds if necessary

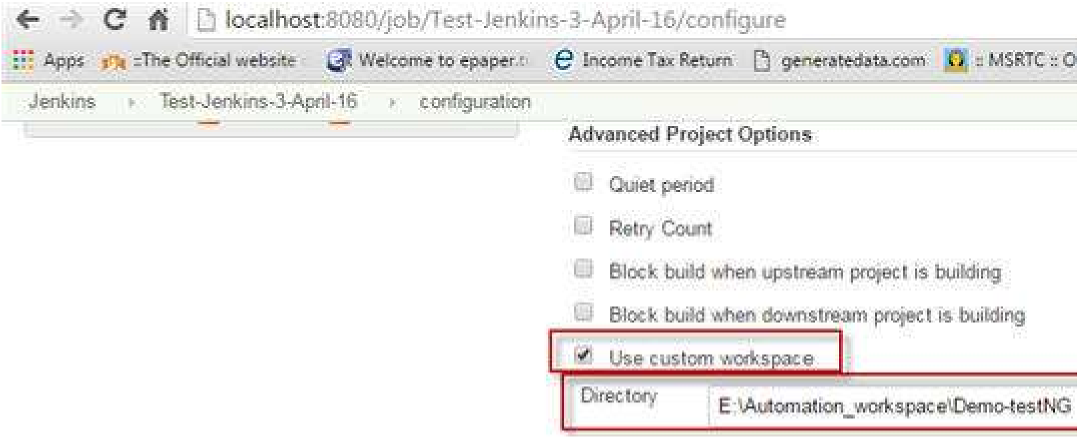
JDK: (System)
JDK to be used for this project

Advanced Project Options

Source Code Management

● None
○ CVS
○ CVS Projectset

**4)** Click on use custom workspace and give your Selenium script project workspace path: "*E:\Automation_workspace\Demo-testNG*"



Advanced Project Options

□ Quiet period

□ Retry Count

□ Block build when upstream project is building

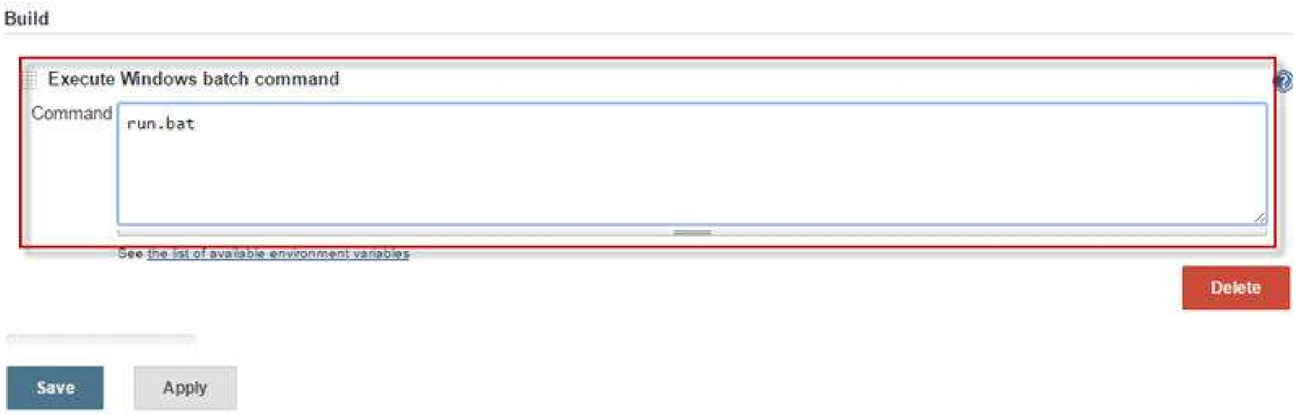
□ Block build when downstream project is building

☑ Use custom workspace

Directory: E:\Automation_workspace\Demo-testNG

**5)** Then go to Build and Select option from drop down box, execute your build through Windows batch command

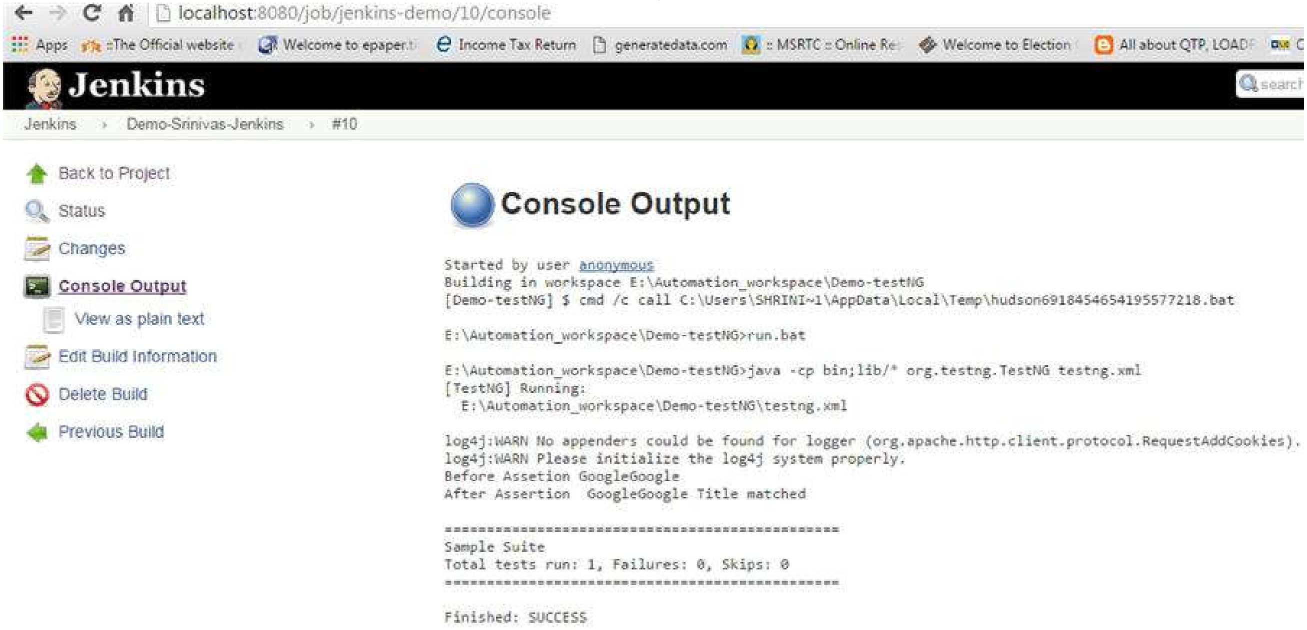
**6)** And give your batch file name here – "*run.bat*"


Build
Execute Windows batch command
Command
See the list of available environment variables
Delete
Save
Apply

```
run.bat
```

**7)** Click on apply and save

**8)** Click on build now and see the build result on console output



```
Started by user anonymous
Building in workspace E:\Automation_workspace\Demo-testNG
[Demo-testNG] $ cmd /c call C:\Users\SHRINI~1\AppData\Local\Temp\hudson6918454654195577218.bat

E:\Automation_workspace\Demo-testNG>run.bat

E:\Automation_workspace\Demo-testNG>java -cp bin;lib/* org.testng.TestNG testng.xml
[TestNG] Running:
  E:\Automation_workspace\Demo-testNG\testng.xml

log4j:WARN No appenders could be found for logger (org.apache.http.client.protocol.RequestAddCookies).
log4j:WARN Please initialize the log4j system properly.
Before Assetion GoogleGoogle
After Assertion  GoogleGoogle Title matched

===============================================
Sample Suite
Total tests run: 1, Failures: 0, Skips: 0
===============================================

Finished: SUCCESS
```

**So far we have learned:**

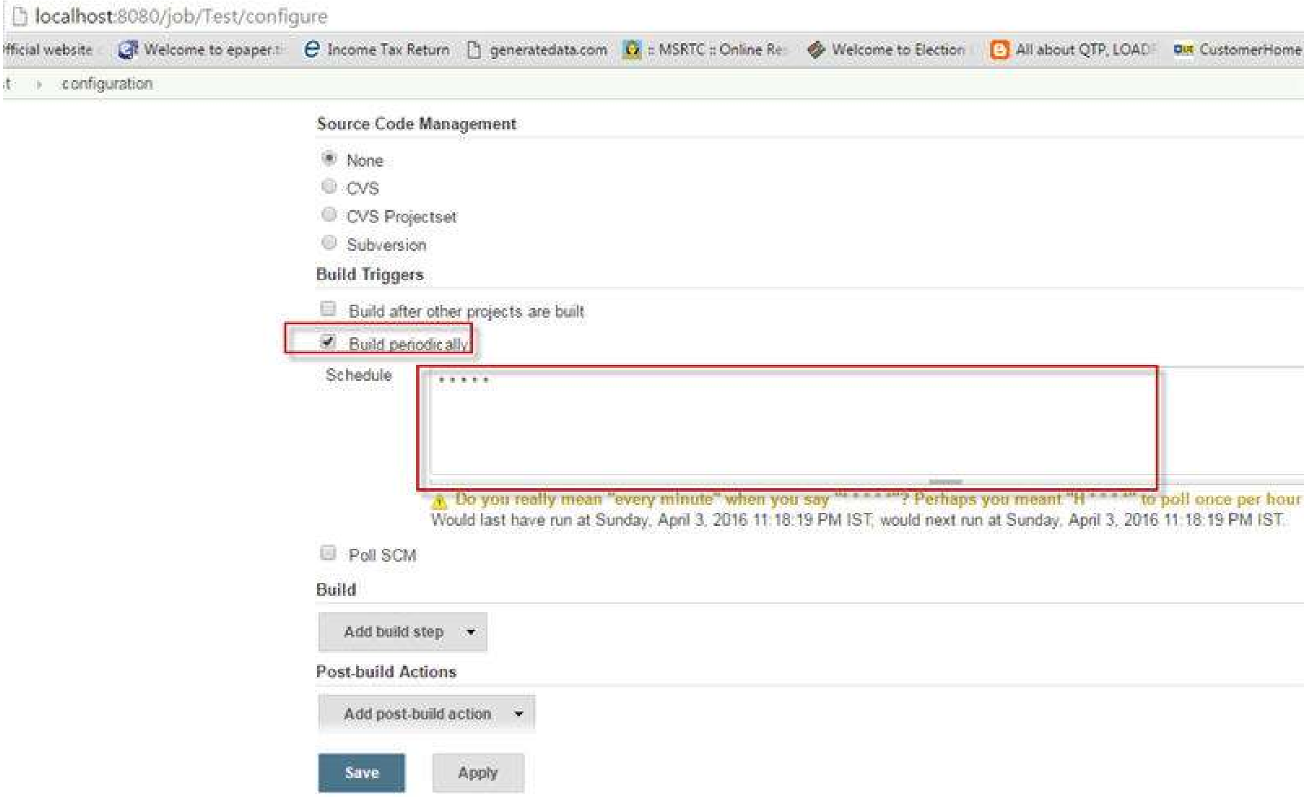
- How to start Jenkins
- How to configure Jenkins with Selenium
- Creating your batch file and executing it through Jenkins.

As you all are aware Jenkins is a very powerful tool which is mainly used for running nightly builds. Hence, we shall now learn how to schedule your build and send email notifications to the concerned team.

**Scheduling  Jenkins job**

**For scheduling you batch file, perform the below steps:**

- Go to dashboard and click on the Jenkins job
- Click on configure and then on advanced  option
- Then go to **Build triggers** and select **Build periodically** option and enter your cron job pattern



Source Code Management

● None
○ CVS
○ CVS Projectset
○ Subversion

Build Triggers

□ Build after other projects are built
☑ Build periodically

Schedule: * * * * *

⚠ Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour
Would last have run at Sunday, April 3, 2016 11:18:19 PM IST; would next run at Sunday, April 3, 2016 11:18:19 PM IST.

□ Poll SCM

Build

Post-build Actions

- To understand cron job pattern follow this wiki link

I entered * * * * * which means it will run my job every minute

- Click on apply and save

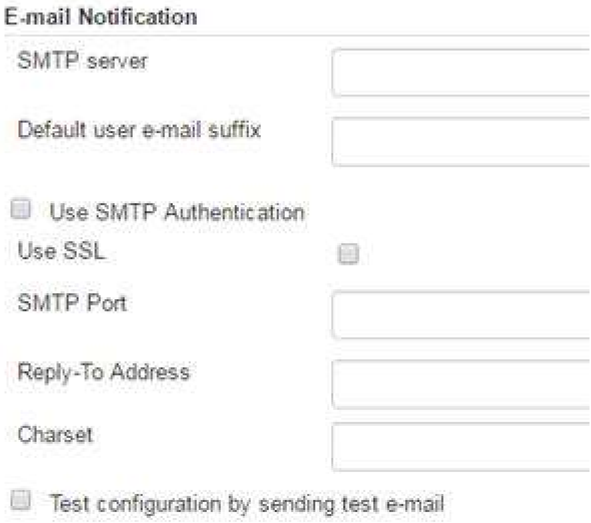
There is no manual intervention. After scheduling the script, it will run on the scheduled time.

**How to add email notifications**

**Next, we will cover how to add email notifications.**

**Refer the below steps:**

- Go to the section ‘Manage Jenkins’
- Click on configure system
- Select Email notification



E-mail Notification

SMTP server

Default user e-mail suffix

□ Use SMTP Authentication

Use SSL □

SMTP Port

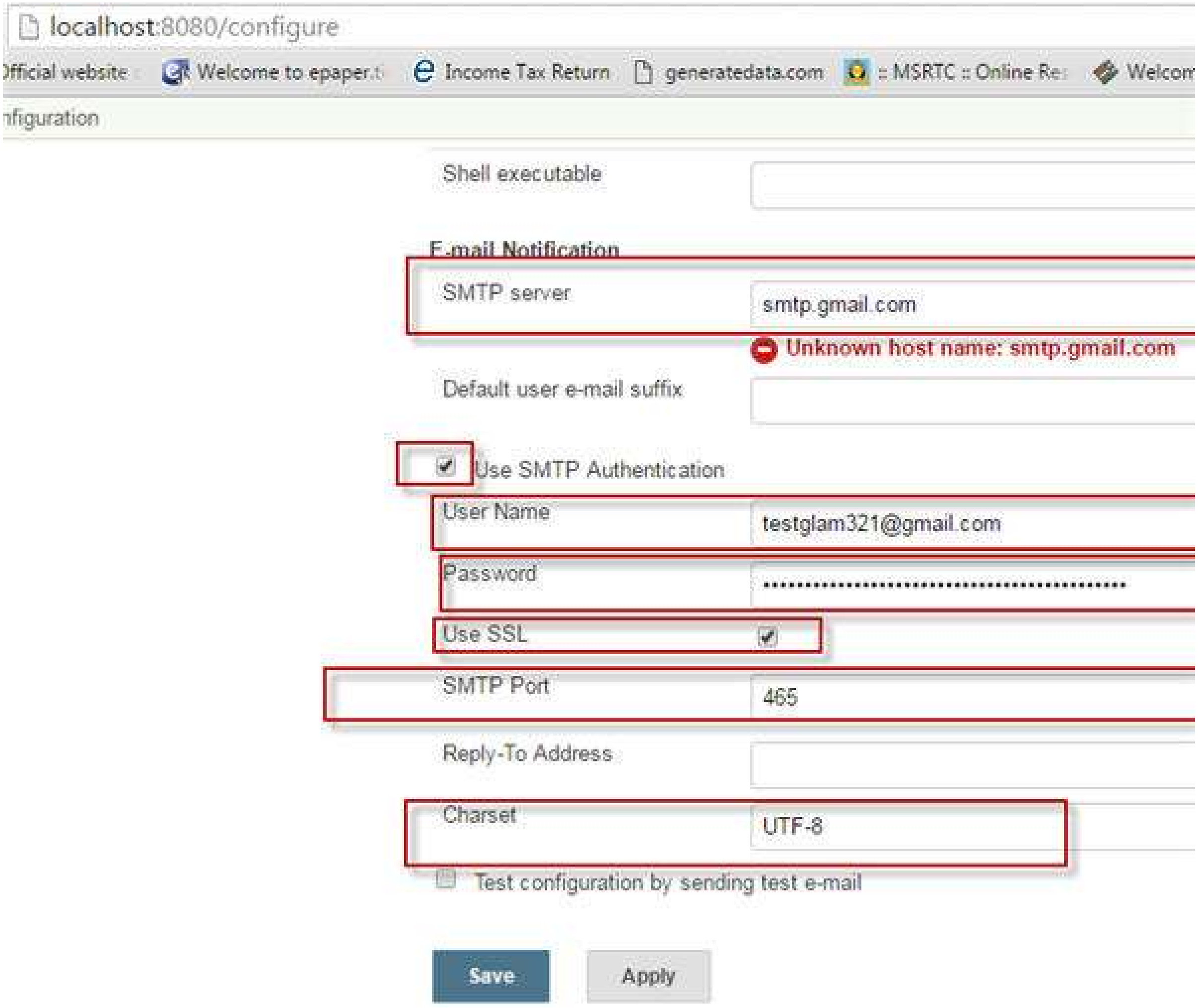
Reply-To Address

Charset

□ Test configuration by sending test e-mail

- Give your SMTP server address. I am using Gmail, as I can’t mention my official server address. To know your official server address, contact your network support team
- I entered SMTP server name = **smtp.gmail.com**
- Click on the advance link and check Use SMTP Authentication check box

- Provide username, password and SMTP port number; it is 465 for Gmail. Check charset and make sure it is = UTF-8



Shell executable

E-mail Notification

SMTP server: smtp.gmail.com

Unknown host name: smtp.gmail.com

Default user e-mail suffix

☑ Use SMTP Authentication

User Name: testglam321@gmail.com

Password: ••••••••••••••••••••••••••••••••••••••••••••••••

Use SSL ☑

SMTP Port: 465

Reply-To Address

Charset: UTF-8

□ Test configuration by sending test e-mail

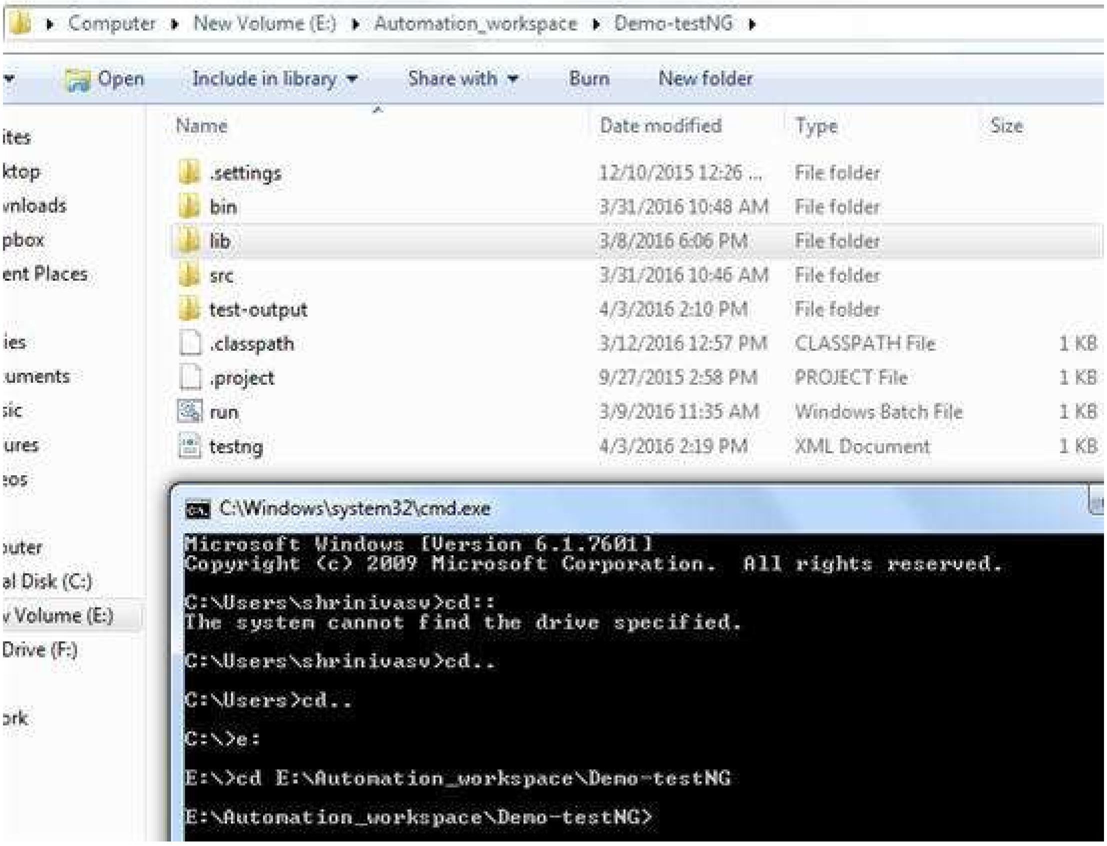
- Check your email configuration settings by clicking on Test configuration button.
- So, whenever the build passes or fails you will get the email notification.

**Running Selenium script through command line**

We will now see **how we can run Selenium script through command prompt**. This part has nothing to do with Jenkins. I am sharing this to give extra insights on Selenium.

**Follow below steps:**

- Open your command prompt and go to your project base path

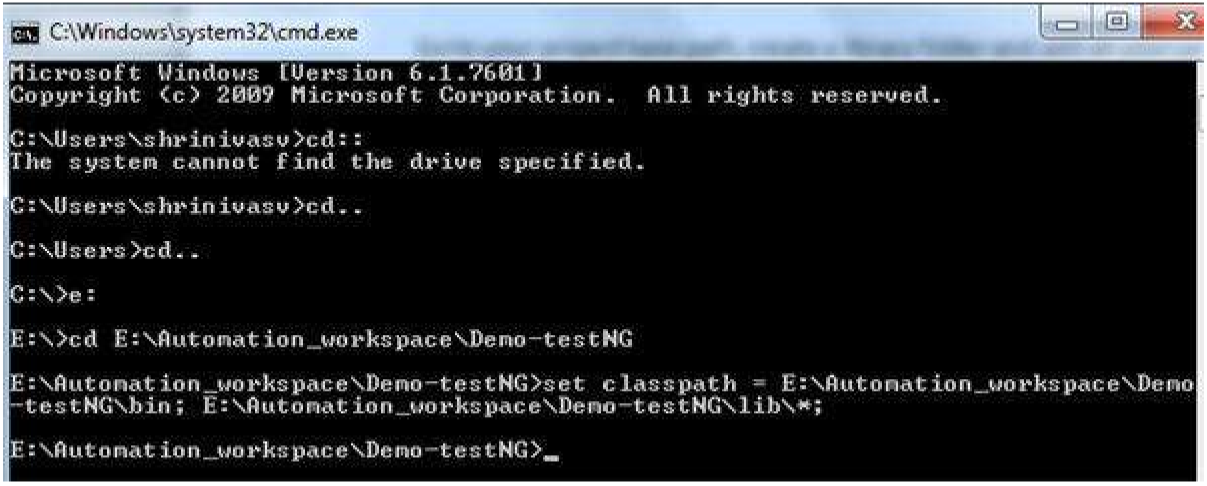


| Name | Date modified | Type | Size |
|---|---|---|---|
| .settings | 12/10/2015 12:26 ... | File folder | |
| bin | 3/31/2016 10:48 AM | File folder | |
| lib | 3/8/2016 6:06 PM | File folder | |
| src | 3/31/2016 10:46 AM | File folder | |
| test-output | 4/3/2016 2:10 PM | File folder | |
| .classpath | 3/12/2016 12:57 PM | CLASSPATH File | 1 KB |
| .project | 9/27/2015 2:58 PM | PROJECT File | 1 KB |
| run | 3/9/2016 11:35 AM | Windows Batch File | 1 KB |
| testng | 4/3/2016 2:19 PM | XML Document | 1 KB |

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\shrinivasv>cd::
The system cannot find the drive specified.

C:\Users\shrinivasv>cd..

C:\Users>cd..

C:\>e:

E:\>cd E:\Automation_workspace\Demo-testNG

E:\Automation_workspace\Demo-testNG>
```

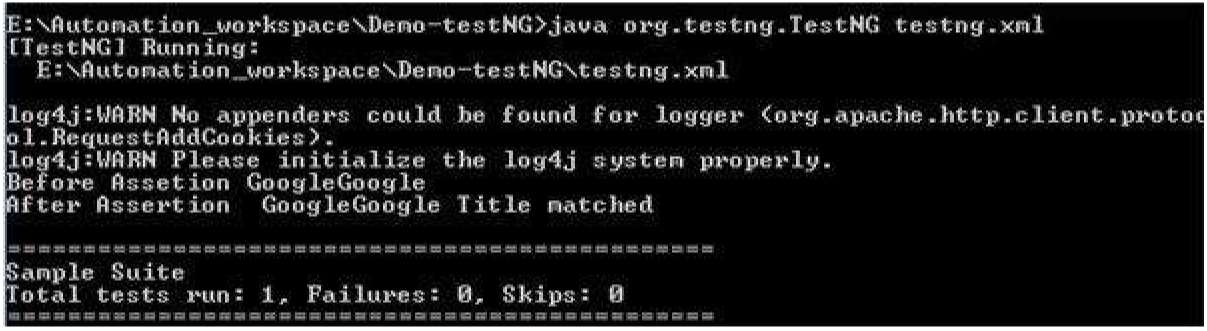
- Set class path for your script file; which means we are specifying that our binary and library files are stored in this location

*E:\Automation_workspace\Demo-testNG > set classpath = E:\Automation_workspace\Demo-testNG\bin;E:\Automation_workspace\Demo-testNG\lib\*;*



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\shrinivasv>cd::
The system cannot find the drive specified.

C:\Users\shrinivasv>cd..

C:\Users>cd..

C:\>e:

E:\>cd E:\Automation_workspace\Demo-testNG

E:\Automation_workspace\Demo-testNG>set classpath = E:\Automation_workspace\Demo
-testNG\bin; E:\Automation_workspace\Demo-testNG\lib\*;

E:\Automation_workspace\Demo-testNG>_
```

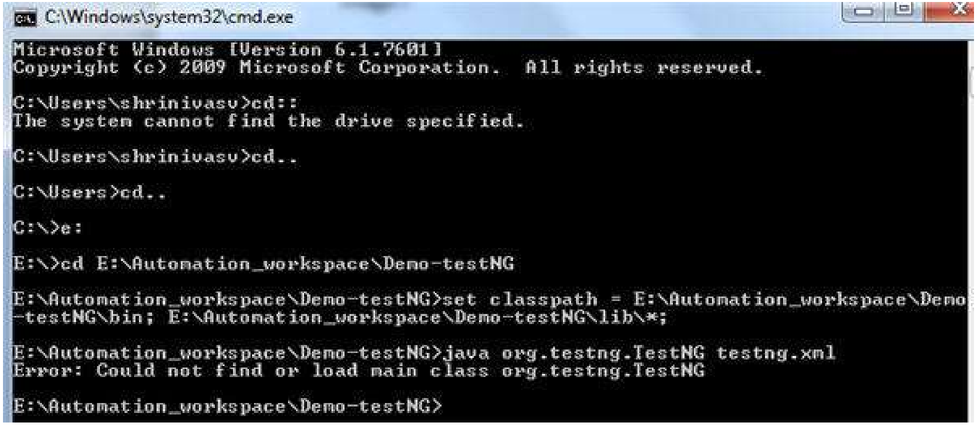
- Execute your testng.xml file by typing the command – ***java org.testng.TestNG testng.xml***



```
E:\Automation_workspace\Demo-testNG>java org.testng.TestNG testng.xml
[TestNG] Running:
  E:\Automation_workspace\Demo-testNG\testng.xml

log4j:WARN No appenders could be found for logger (org.apache.http.client.protoc
ol.RequestAddCookies).
log4j:WARN Please initialize the log4j system properly.
Before Assetion GoogleGoogle
After Assertion  GoogleGoogle Title matched

===============================================
Sample Suite
Total tests run: 1, Failures: 0, Skips: 0
===============================================
```

- When you press enter your script will start executing and you can see the test result in the UI

Sometimes while executing your script you may face error which says, "Could not find or load main class org.testng.TestNG"



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\shrinivasv>cd::
The system cannot find the drive specified.

C:\Users\shrinivasv>cd..

C:\Users>cd..

C:\>e:

E:\>cd E:\Automation_workspace\Demo-testNG

E:\Automation_workspace\Demo-testNG>set classpath = E:\Automation_workspace\Demo
-testNG\bin; E:\Automation_workspace\Demo-testNG\lib\*;

E:\Automation_workspace\Demo-testNG>java org.testng.TestNG testng.xml
Error: Could not find or load main class org.testng.TestNG

E:\Automation_workspace\Demo-testNG>
```

Then you need to close your command prompt and again set your class path as mentioned above and repeat the same steps. Your error will get resolved and the script will run.

**Conclusion**

Integration of Jenkins with selenium provides you to run your script each time there is any change in software code and deploy the code in a new environment. With Jenkins, you can save execution history and test reports.

In short, Jenkins is very useful when you have test cases ready and you want them to run using a single click. We can create or schedule a build to run the test cases using a batch file.

**Further reading => Integrate Selenium with Maven project**