

# Jenkins installation with FIRE

INDEX.....	1
<a href="#">Jenkins installation .....</a>	<a href="#">2</a>
<a href="#">Configure Jenkins with selenium .....</a>	<a href="#">7</a>
JENKINS CONFIGURATION.....	11
<a href="#">Integration of FIRE scripts to JENKINS .....</a>	<a href="#">11</a>
<a href="#">Configure FIRE .....</a>	<a href="#">16</a>

## Chapter 1

### Jenkins installation

Steps need to follow:

#### Step # 1:

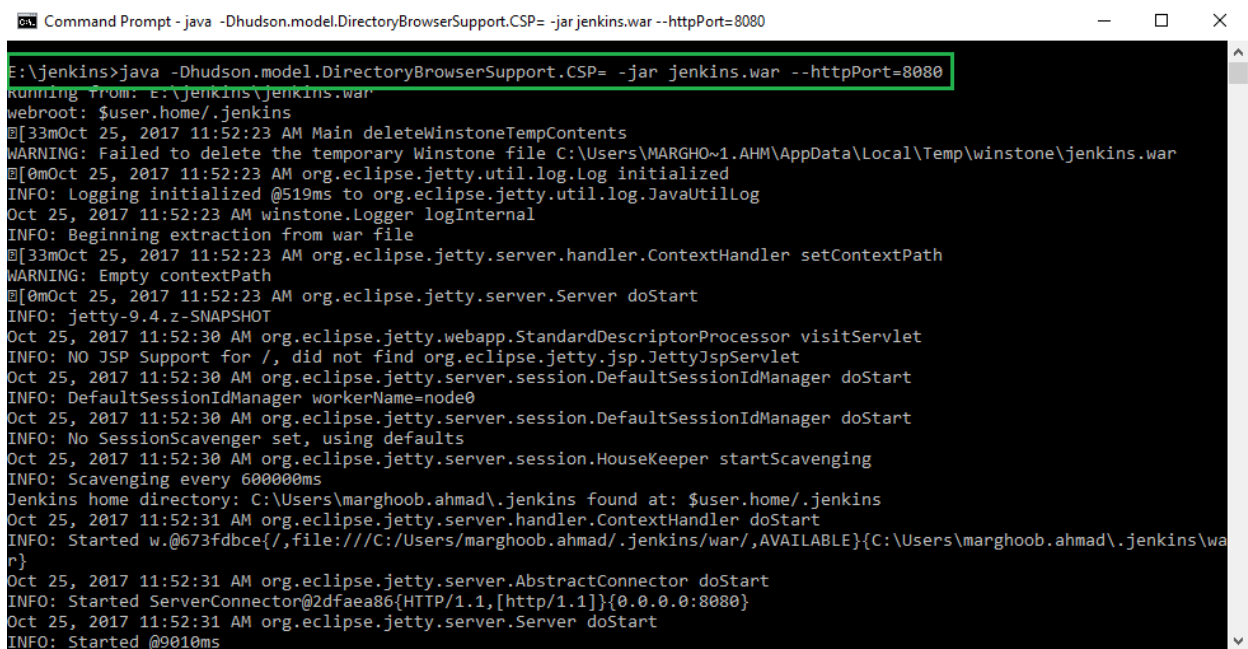
Download Jenkins from <https://jenkins.io/download/> , Download latest Jenkins for Generic java package (.war)

#### Step # 2:

Open a command prompt and type

```
java -Dhudson.model.DirectoryBrowserSupport.CSP= -jar jenkins.war --httpPort=<portno>
```

Note: Jenkins default port is 8080



```
Command Prompt - java -Dhudson.model.DirectoryBrowserSupport.CSP= -jar jenkins.war --httpPort=8080

E:\jenkins>java -Dhudson.model.DirectoryBrowserSupport.CSP= -jar jenkins.war --httpPort=8080
Running from: E:\jenkins\jenkins.war
webroot: $user.home/.jenkins
Oct 25, 2017 11:52:23 AM Main deleteWinstoneTempContents
WARNING: Failed to delete the temporary Winstone file C:\Users\MARGHO~1.AHM\AppData\Local\Temp\winstone\jenkins.war
Oct 25, 2017 11:52:23 AM org.eclipse.jetty.util.log.Log initialized
INFO: Logging initialized @519ms to org.eclipse.jetty.util.log.JavaUtilLog
Oct 25, 2017 11:52:23 AM winstone.Logger logInternal
INFO: Beginning extraction from war file
Oct 25, 2017 11:52:23 AM org.eclipse.jetty.server.handler.ContextHandler setContextPath
WARNING: Empty contextPath
Oct 25, 2017 11:52:23 AM org.eclipse.jetty.server.Server doStart
INFO: jetty-9.4.z-SNAPSHOT
Oct 25, 2017 11:52:30 AM org.eclipse.jetty.webapp.StandardDescriptorProcessor visitServlet
INFO: NO JSP Support for /, did not find org.eclipse.jetty.jsp.JettyJspServlet
Oct 25, 2017 11:52:30 AM org.eclipse.jetty.server.session.DefaultSessionIdManager doStart
INFO: DefaultSessionIdManager workerName=node0
Oct 25, 2017 11:52:30 AM org.eclipse.jetty.server.session.DefaultSessionIdManager doStart
INFO: No SessionScavenger set, using defaults
Oct 25, 2017 11:52:30 AM org.eclipse.jetty.server.session.HouseKeeper startScavenging
INFO: Scavenging every 60000ms
Jenkins home directory: C:\Users\marghoob.ahmad\.jenkins found at: $user.home/.jenkins
Oct 25, 2017 11:52:31 AM org.eclipse.jetty.server.handler.ContextHandler doStart
INFO: Started w.@673fdbce[/,file:///C:/Users/marghoob.ahmad/.jenkins/war/,AVAILABLE]{C:\Users\marghoob.ahmad\.jenkins\war}
Oct 25, 2017 11:52:31 AM org.eclipse.jetty.server.AbstractConnector doStart
INFO: Started ServerConnector@2d4faa86{HTTP/1.1,[http/1.1]}{0.0.0.0:8080}
Oct 25, 2017 11:52:31 AM org.eclipse.jetty.server.Server doStart
INFO: Started @9010ms
```

#### Step # 3:

Wait for the console output to print '*Jenkins is fully up and running*' and that's it! You have successfully started the Jenkins server!

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

#### Step # 4 :

Now go to the browser and type <http://localhost:8080> , you will be getting following screen to Unlock Jenkins. You need to enter Administrator password.

In order to get Administrator password, go to the cmd command prompt and check password, just copy password and enter.

## Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`C:\Users\m\.jenkins\secrets\initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

```
Select Command Prompt - java -jar jenkins.war
INFO: Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@3b447308: def
ining beans [filter,legacy]; root of factory hierarchy
Oct 12, 2017 4:10:50 PM jenkins.install.SetupWizard init
INFO:
*****
*****
*****

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

bf4b61f3ad364bafb382fa23bd765d58

This may also be found at: C:\Users\m\.jenkins\secrets\initialAdminPassword

*****
*****
*****

Oct 12, 2017 4:10:55 PM hudson.model.UpdateSite updateData
INFO: Obtained the latest update center data file for UpdateSource default
Oct 12, 2017 4:10:56 PM hudson.model.DownloadService$Downloadable load
INFO: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
Oct 12, 2017 4:10:56 PM hudson.model.UpdateSite updateData
INFO: Obtained the latest update center data file for UpdateSource default
Oct 12, 2017 4:10:56 PM hudson.WebAppMain$3 run
INFO: Jenkins is fully up and running
```

### Step # 5 :

Let's customize Jenkins. Therefore, need to install suggested plugins. See the below screen shot to the same.

#### Getting Started

# Getting Started

✓ Folders Plugin	OWASP Markup Formatter Plugin	Build Timeout	Credentials Binding Plugin	Folders Plugin ** bouncycastle API Plugin
Timestampers	Workspace Cleanup Plugin	Ant Plugin	Gradle Plugin	
Pipeline	GitHub Branch Source Plugin	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View Plugin	
Git plugin	Subversion Plug-in	SSH Slaves plugin	Matrix Authorization Strategy Plugin	
PAM Authentication plugin	LDAP Plugin	Email Extension Plugin	Mailer Plugin	

**Step # 6:**

Create admin user and save.

**Getting Started**

## Create First Admin User

Username:

Password:

Confirm password:

Full name:

E-mail address:

**Step # 7:**

Finally, you will be able to see Jenkins dashboard as below.



Jenkins ▶

- New Item
- People
- Build History
- Manage Jenkins
- My Views
- Credentials

## Welcome to Jenkins!

Please **create new jobs** to get started.

### Build Queue

No builds in the queue.

### Build Executor Status

- 1 Idle
- 2 Idle

## Chapter 2

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

#### Step # 1:

##### 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
- Add list of variables under environment variables, java home and maven home

##### Global properties

☒ Environment variables

List of variables

Name	<input type="text" value="JAVA_HOME"/>	
Value	<input type="text" value="C:\Program Files\Java\jdk1.8.0_112"/>	<input type="button" value="Delete"/>
Name	<input type="text" value="MAVEN_HOME"/>	<input type="button" value="Delete"/>
Value	<input type="text" value="D:\Maven"/>	<input type="button" value="Delete"/>
<input type="button" value="Add"/>		

#### Step # 2:

Click on “Create new jobs” and enter the tem name eg : “TestMavenSeleniumProject”

#### Step # 3 :


Select Maven project and click ok

Enter an item name


TestMavenSeleniumProject

» Required field


---

 **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.


---

 **Pipeline**  
Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.


---

 **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.


---

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


---

 **Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

---


 **GitHub Organization**  
Scans a GitHub organization (or user account) for all repositories matching some defined markers.

---

 **Multibranch Pipeline**  
Creates a set of Pipeline projects according to detected branches in one SCM repository.

---

If you want to create a new item from other existing, you can use this option:



#### Step # 4 :

Click on project “Configure” and enter the build value,

Root POM : need to add project pom.xml file

Goals and Options : just add the maven command to perform action

Eg : clean test -Dxml=custom -Denv=custom

You can add email notification to add Recipients email id inside Build settings.

Click on apply and save.



**Build**

Root POM:

Goals and options:

Advanced...

**Post Steps**

☐ Run only if build succeeds
 ☐ Run only if build succeeds or is unstable
 ☒ Run regardless of build result

Should the post-build steps run only for successful builds, etc.

Add post-build step

**Build Settings**

☒ E-mail Notification

Recipients:

☒ Send e-mail for every unstable build  
☐ Send separate e-mails to individuals who broke the build  
☒ Send e-mail for each failed module

Save Apply

## Step # 5 :

Go back to the project and click on Build now link

← → ↻ ⓘ localhost:8080/job/TestMavenSeleniumProject/

Apps For quick access, place your bookmarks here on the bookmarks bar. [Import bookmarks now...](#)

# Jenkins

Jenkins > TestMavenSeleniumProject >

[Back to Dashboard](#)  
[Status](#)  
[Changes](#)  
[Workspace](#)  
[Build Now](#)  
[Delete Maven project](#)  
[Configure](#)  
[Modules](#)  
[Dependency Graph](#)

## Maven project TestMavenSeleniumProject

[Workspace](#)  
[Recent Changes](#)

### Permalinks

- [Last build \(#7\). 9 days 5 hr ago](#)
- [Last stable build \(#2\). 11 days ago](#)
- [Last successful build \(#2\). 11 days ago](#)
- [Last failed build \(#7\). 9 days 5 hr ago](#)
- [Last unsuccessful build \(#7\). 9 days 5 hr ago](#)
- [Last completed build \(#7\). 9 days 5 hr ago](#)

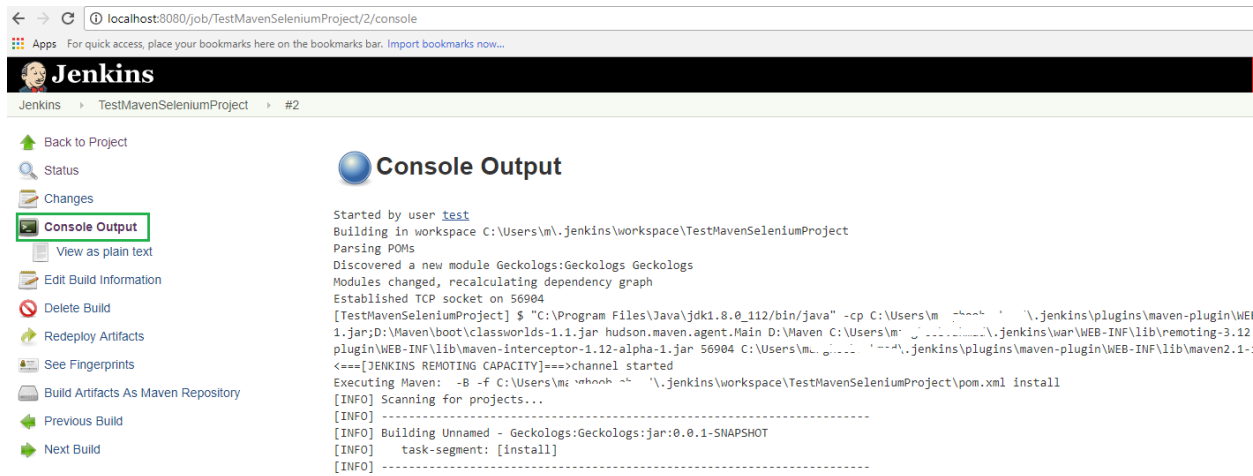
**Build History** trend

find x

#7	Oct 16, 2017 12:29 PM
#6	Oct 16, 2017 12:28 PM

## Step # 6 :

Verify console output to check the Build status.



The screenshot shows the Jenkins web interface in a browser. The address bar indicates the URL is `localhost:8080/job/TestMavenSeleniumProject/2/console`. The Jenkins logo and navigation links are visible at the top. On the left sidebar, the 'Console Output' link is highlighted with a red box. The main content area is titled 'Console Output' and displays the following text:

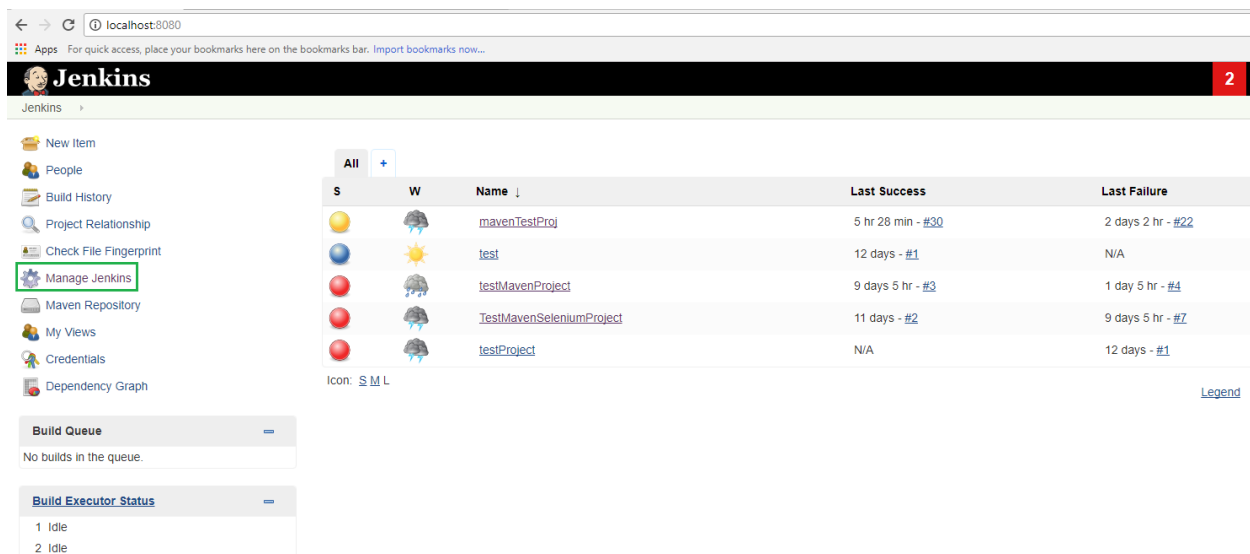
```
Started by user test
Building in workspace C:\Users\m\workspace\TestMavenSeleniumProject
Parsing POMs
Discovered a new module Geckologs:Geckologs Geckologs
Modules changed, recalculating dependency graph
Established TCP socket on 56904
[TestMavenSeleniumProject] $ "C:\Program Files\Java\jdk1.8.0_112\bin/java" -cp C:\Users\m\workspace\TestMavenSeleniumProject\pom.xml \
1.jar;D:\Maven\boot\classworlds-1.1.jar hudson.maven.agent.Main D:\Maven C:\Users\m\workspace\TestMavenSeleniumProject\pom.xml \
plugin\WEB-INF\lib\maven-interceptor-1.12-alpha-1.jar 56904 C:\Users\m\workspace\TestMavenSeleniumProject\pom.xml \
<====[JENKINS REMOTING CAPACITY]====>channel started
Executing Maven: -B -f C:\Users\m\workspace\TestMavenSeleniumProject\pom.xml install
[INFO] Scanning for projects...
[INFO] Building Unnamed - Geckologs:Geckologs:jar:0.0.1-SNAPSHOT
[INFO] task-segment: [install]
[INFO]
```

## Chapter 3

# Integration of FIRE scripts to JENKINS

Pre-requisite:

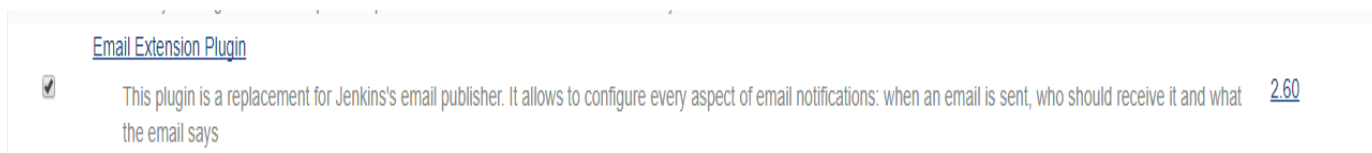
Step #1. Click on manage Jenkins in order to Manage Plugins



The screenshot shows the Jenkins web interface at localhost:8080. The left sidebar contains a menu with items like 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins' (highlighted with a green box), 'Maven Repository', 'My Views', 'Credentials', and 'Dependency Graph'. The main content area shows a table of build jobs. The table has columns for status (S), weather icon (W), name, last success, and last failure. The jobs listed are 'mavenTestProj', 'test', 'testMavenProject', 'TestMavenSeleniumProject', and 'testProject'.

S	W	Name	Last Success	Last Failure
Yellow circle	Cloud with rain	mavenTestProj	5 hr 28 min - #30	2 days 2 hr - #22
Blue circle	Sun	test	12 days - #1	N/A
Red circle	Cloud with rain	testMavenProject	9 days 5 hr - #3	1 day 5 hr - #4
Red circle	Cloud with rain	TestMavenSeleniumProject	11 days - #2	9 days 5 hr - #7
Red circle	Cloud with rain	testProject	N/A	12 days - #1

For email need to add email extension plugins

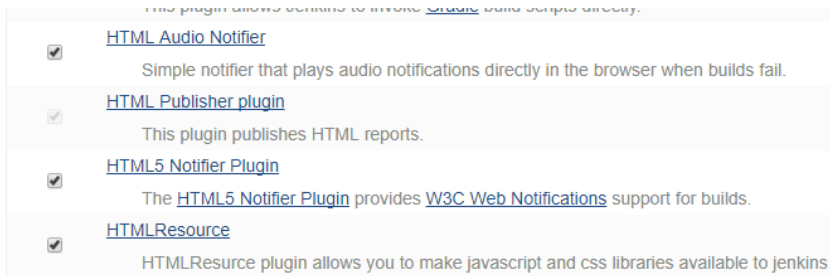


Email Extension Plugin

This plugin is a replacement for Jenkins's email publisher. It allows to configure every aspect of email notifications: when an email is sent, who should receive it and what the email says

2.60

For HTML report need to add HTML publisher plugin

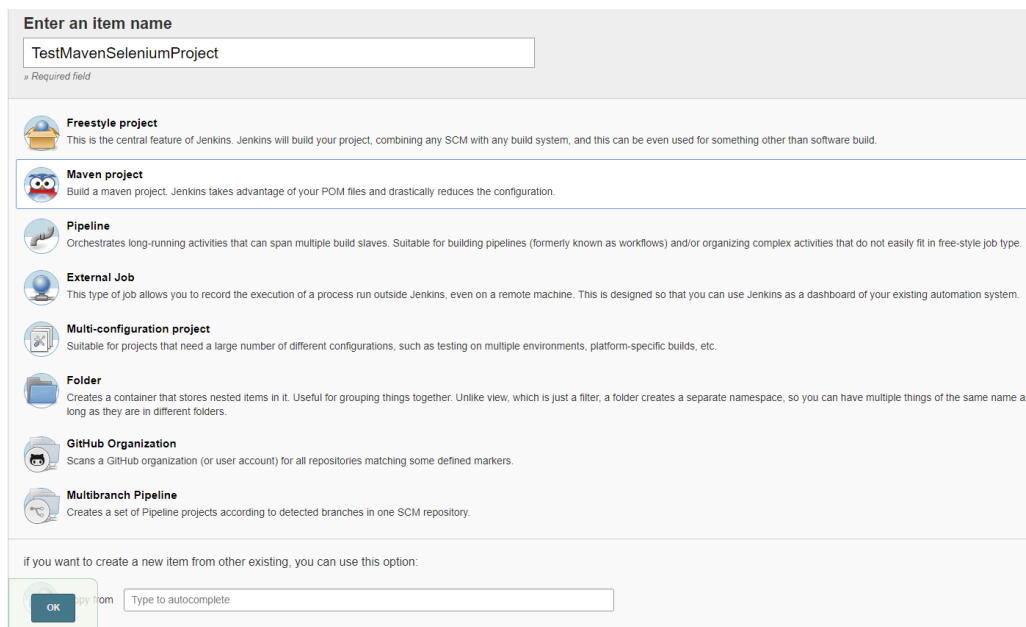


### Step # 2:

Click on **“Create new jobs”** and enter the tem name eg : **“TestMavenSeleniumProject”**

### Step # 3 :

Select Maven project and click ok



### Step # 4 :

Click on project **“Configure”** and enter the build value,

Goals and Options : just add the maven command to perform action

Eg : `clean test -Dxml=custom -Denv=custom`

### Step # 5:

You can add email notification in order to add Recipients email id inside Build settings.

### Step # 6:

We can **schedule jobs** in Jenkins using Build periodically.

If we wanted to trigger build automatically, however need to set build periodically option under build triggers section.

eg: If you want to schedule your build every

5 minutes, this will do the job : `*/5 * * * *`

1 hour, this will do the job: `H/1 * * * *`

6:00PM and 1 AM every day : `0 1,18 * * *`

build start daily at morning - `03 09 * * 1-5`

build start at midnight - `59 23 * * 1-5` OR `@midnight`

November 29, 9:30 AM: `30 9 29 11 *`

December 23, 7:30 PM : `30 19 23 12 *`

**Build Triggers**

- ☐ Build whenever a SNAPSHOT dependency is built
- ☐ Trigger builds remotely (e.g., from scripts)
- ☐ Build after other projects are built
- ☒ Build periodically

Schedule: 30 19 23 12 \*

⚠ Spread load evenly by using 'H 19 23 12 \*' rather than '30 19 23 12 \*'  
 Would last have run at Friday, December 23, 2016 7:30:42 PM IST; would next run at Saturday December 23, 2017 7:30:42 PM IST.

- ☐ GitHub hook trigger for GITScm polling
- ☐ Poll SCM

### Step # 7:

To generate HTML report, need to add post build actions

**HTML directory to archive:** add project base folder -> test-output

**Index page:** Result.html

**Report title:** HTML report

**Post-build Actions**

**Publish HTML reports**

Reports

HTML directory to archive	E:\EA-DCMT\DCMT\documents\QA\AutomationScripts\dcmt\FireMavenProject\test-output
Index page[s]	Result.html
Index page title[s] (Optional)	
Report title	HTML Report

Publishing options...

Add


Click on **Apply** and **Save**.

### Step # 8 :

Go back to the project and click on Build now link

### Step # 9 :

Go to the project and verify HTML report.


**Jenkins**


Jenkins > mavenTestProj >


[Back to Dashboard](#)  
[Status](#)  
[Changes](#)  
[Workspace](#)  
[Build Now](#)  
[Delete Maven project](#)  
[Configure](#)  
[Modules](#)  
[Dependency Graph](#)  
[HTML Report](#)


## Maven project mavenTestProj


test project

 [HTML Report](#)

 [Workspace](#)

 [Recent Changes](#)

 [Latest Test Result](#) (1 failure / ±0)

 [Latest Test Result](#) (1 failure / ±0)

**Build History** [trend](#)  
 X  

#30 Oct 25, 2017 12:20 PM

TEST EXECUTION REPORT			
SCRIPT NAME:	CUSTOM	TEST DATE:	Oct 23,2017
TEST START TIME:	15:14:25.245	TEST END TIME:	15:15:34.085
DURATION	0 Hours 1 Minutes 8 Seconds	EXECUTED BY :	TI10357/marchook... OS-Windows 10

SUMMARY REPORT			
TOTAL TEST CASES:	TEST CASES PASSED:	TEST CASES FAILED:	TEST CASES SKIPPED:
1	1	0	0

Pass (1)

DETAILED RESULTS REPORT		
FIREFOX - SHOW/HIDE		
TEST NAME	STATUS	REASON
NBCU_TC-01:Verify Content pages	Pass	Duration: 0 Hours 1 Minutes 7 Seconds

## Chapter 4

### Configure FIRE

Pre-requisite:

**To get started on the FIRE Framework, the initial setup steps are as follows**

1. Install JDK and set environment variable JAVA\_HOME
2. Download MAVEN and set environment variable MAVEN\_HOME
3. Copy project “**FireKWDemo**” to system from SVN  
  
(\*path: <http://svn.corp.tavant.com/svnproj/FunctionalAutomation/trunk/DemoProjects>)
4. Download Eclipse and create maven project
5. Run the Command “**mvn clean install**” from projectNameefolder where pom.xml file is located.
6. Now open the Testcases.xls from path “...\\FireKWDemo\\src\\test\\resources”.
7. Next, click on Setup and RUN Button from RunConfiguration tab.
8. Select appropriate test cases / browser/groups and click on Submit.