1. Through TestNG.xml

Just right click the project and say RunAs 🡪 testNG.suite

1. Through pom.xml

To run the test cases using pom.xml, we should add the following plugins under build tag

1. Maven-compiler (<https://maven.apache.org/plugins/maven-compiler-plugin/examples/set-compiler-source-and-target.html>)
2. Maven-surefire like (<https://maven.apache.org/surefire/maven-surefire-plugin/usage.html>)

<build>

<pluginManagement>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.8.1</version>

<configuration>

<suiteXmlFiles>

<suiteXmlFile>testng.xml</suiteXmlFile>

</suiteXmlFiles>

</configuration>

</plugin>

</plugins>

</pluginManagement>

</build>

1. Then right click on pom.xml and say RunAs🡪 MavenTest
2. To run the test cases through command prompt
3. Install the maven software and set the env variable.
4. Navigate to the project location and open the command prompt form there.
5. Type “mvn clean install” (without quotes) and press enter.
6. To run as a run.bat
7. Add “b” and “c” from point 3 in a notepad and save it as run.bat file and double click it.

Note: point b should be mentioned with cd at the beginning [root directory of project]

1. To run via Jenkins
2. Download the Jenkins war file and extract and run it using command prompt with java –jar jenikins.war
3. Open browser and say localhost:8080 and press enter to open Jenkins. Since by default jekins will take the local host
4. Complete the initial one time process like credentials creations and login (admin / Rumble@12) 🡪 Note: If you forgot your password use the following link “<https://www.devglan.com/online-tools/bcrypt-hash-generator>” and use the hashed password from C:\Users\M1030042\.jenkins\users\admin\_5989254161458711088\Config file under “<passwordHash>#” tag. Then type your password and check they are matching.

**For Jk tech – user name is – raghavdce and password is usual one**

**For admin the pwd is -** d234ab3fc0c547eabb9dbec617dcbc93

1. Once logged into Jenkins, click “New Item” and in the new item go to build dropdown and select using window bat file
2. Add

cd <project location>

run.bat

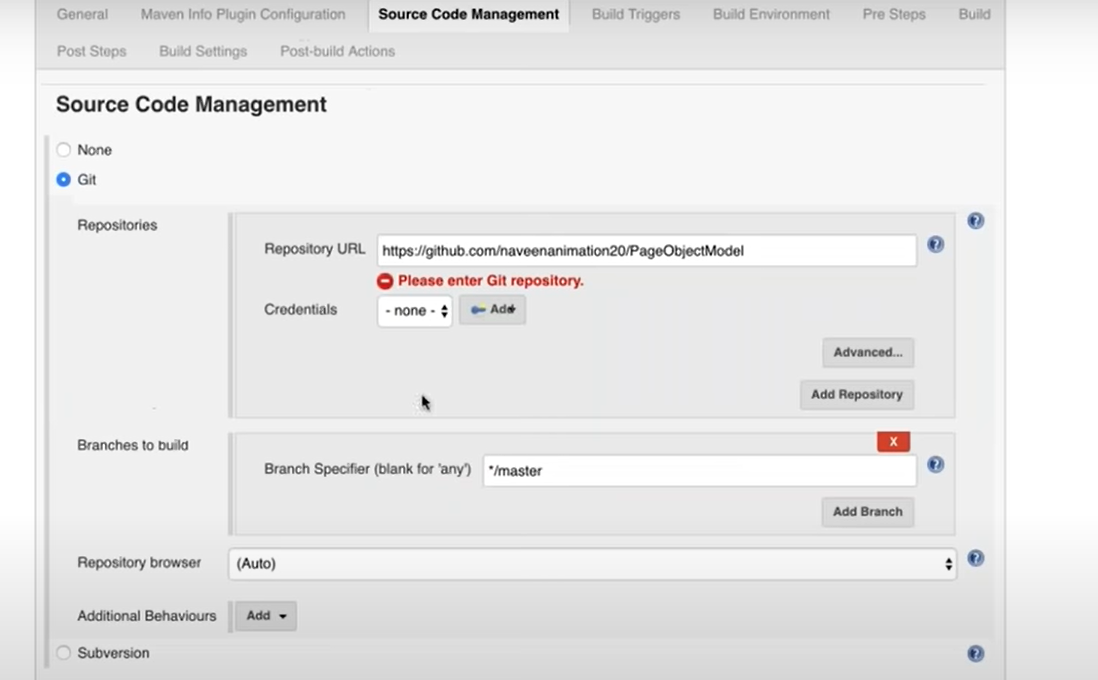
and save

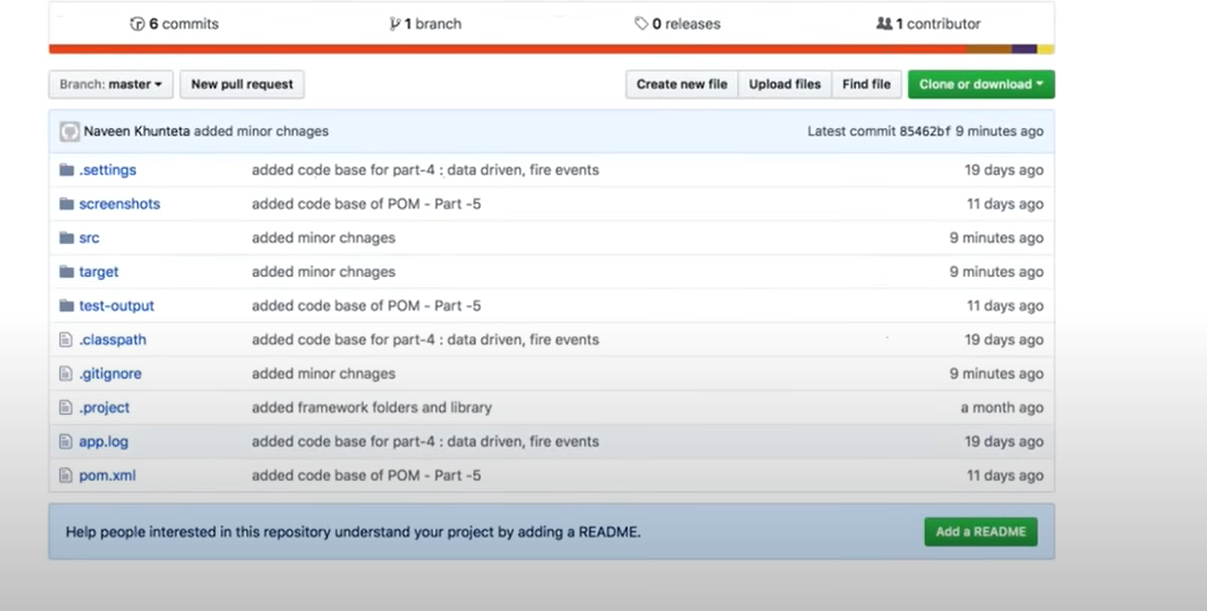
1. Click build now to see the status
2. To configure the Jenkins to run at a particular time, use the “Source Code Management” tab 🡪 then “Build Triggers” sub-tab 🡪 “Build Periodcially” using 5 starts and each star position denotes the below
3. MINUTE (0-59), HOUR (0-23), DAY (1-31), MONTH (1-12), DAY OF THE WEEK (0-6)

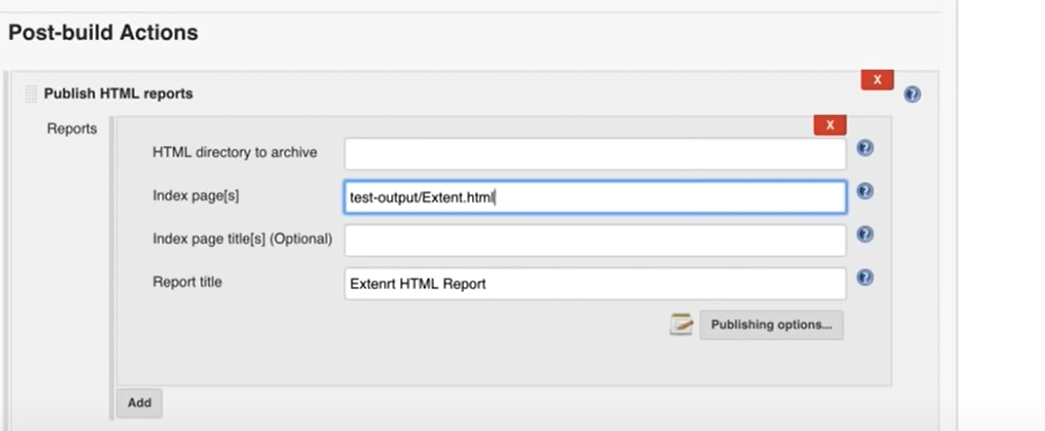
**How to connect the code from GIT to Jenkins:**

**=========================================**

1. After starting the Jenkins and launched the Jenkins in local host port, create a maven style project
2. Go to source code management tab and select “GIT” radio button and pass the GIT repo url



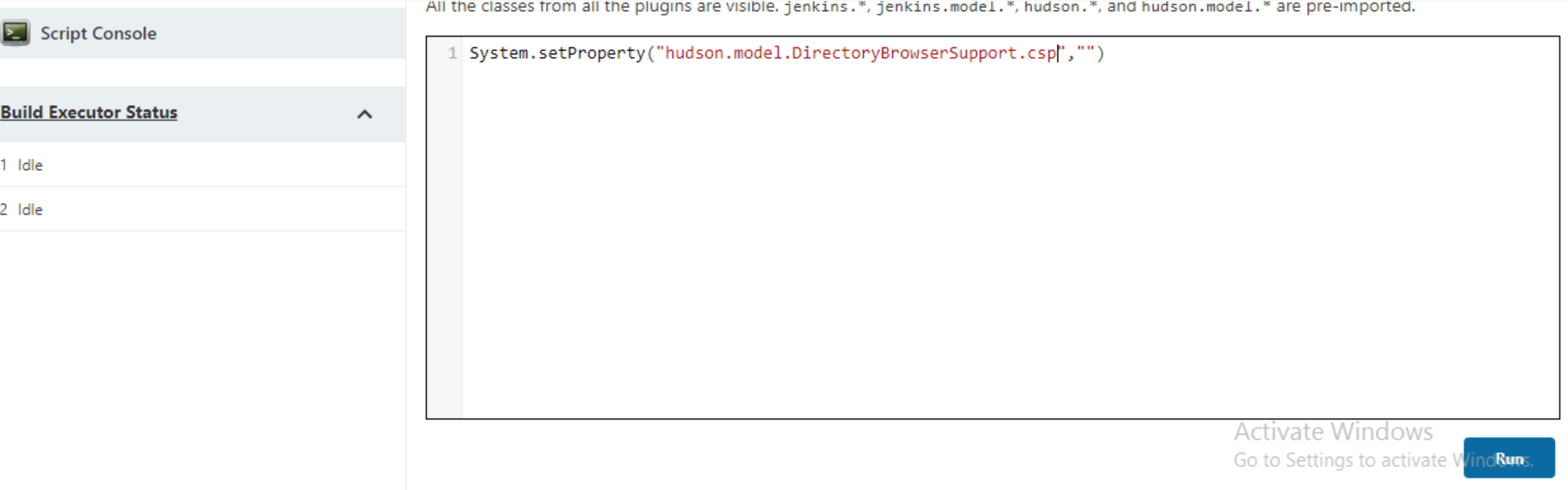
1. Rest of the options are same as above
2. Note: Build section where we mention the POM.xml path will have the value as pom.xml since we have given the root directory of the project from GIT and the pom.xml is in root folder
3. 
4. 
5. In post build section we can select Extent HTML report and do the following config



1. Also when the report is generated, starting from Jenkins 1.6, it is not showing the report in readable format and to overcome that we need to add a system.setproperty in

Jenkins icon on top left🡪 manage Jenkins 🡪 Manage Nodes & Clouds 🡪 Setting (gear icon) script console on the left pane and add the below on the right hand side window

**System.setProperty(“hudson.model.DirectoryBrowserSupport.csp”,””)**🡪 Run



To add tags in Jenkins

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

To see the reports, we need to update the goals as

**test verify -Dcucumber.options=”—tags @smoke”**

Also, the word mvn is removed bcoz we explicitly mentioned that we want to run as maven targets in Jenkins.

Instead of hardcoding the tag name in maven command, let us try to make it dynamic. To achieve this, select “This project is parameterized” option and select choice parameter.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

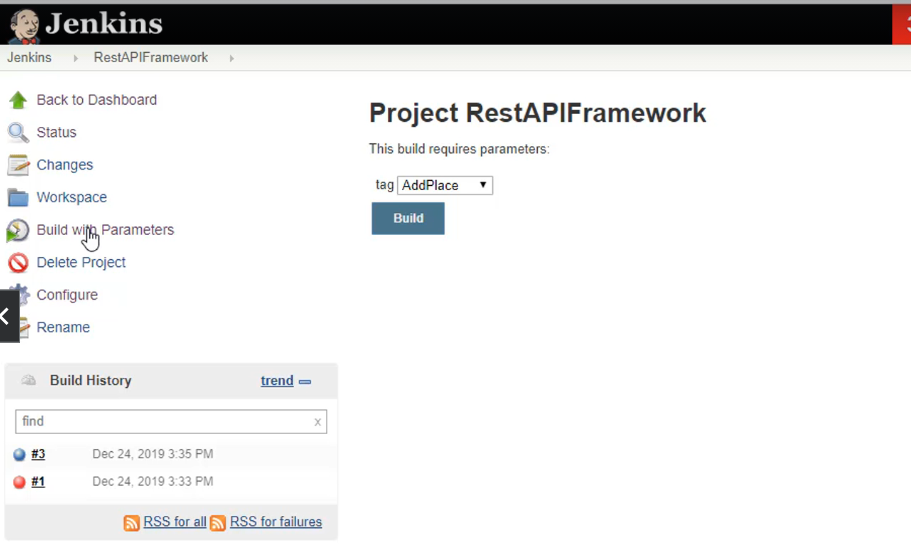
The choices text box should have all tag name like smoke, regression, functional etc as mentioned in feature file in cucumber and click save

Now, let’s modify the mvn comment in the build trigger section to make the tag name dynamic.

Test verify -Dcucumber.options=”—tags @”$**tag**””

**tag** next to $ symbol is nothing but the choice parameter name and user can give any name of their choice

If you refresh the home screen, we can start seeing the drop down with tag name as values



Select the value from the drop down and click build button to run the project.

To see the report click back to project 🡪 workspace 🡪 target 🡪 cucumber-html-report 🡪 overview-features.html

**To send email notification from Jenkins**

**Source -** <https://www.youtube.com/watch?v=g-Blt_cYFrI>

Dashboard 🡪 manage Jenkins 🡪 plugins 🡪 available plugins 🡪 email extension plugin and email extension template plugin 🡪 select the plugins and click install

Back to manage Jenkins 🡪 system configuration 🡪 system 🡪 scroll to the bottom of the page and look for Email Notification.

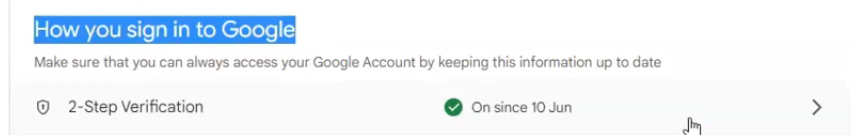
SMTP server – for gmail – smptp.gmail.com

Click advanced



Enter username as email id and password (it should not be gmail password but we need to create google app password)

To create password 🡪 login to gmail 🡪 click profile icon on the right 🡪 manage your google account 🡪 security 🡪 under how you sign in to google section 🡪 2-step verification 🡪 enable it



Enable it by following the simple step like entering email id, phone number and OTP.

After we enable it, click on it 🡪 manage settings 🡪 scroll down to the app password section

A screenshot of a computer

Description automatically generated

Click on it 🡪

A screenshot of a computer

Description automatically generated

Select other 🡪 enter Jenkins 🡪 click generate and it will generate the password

A screenshot of a computer

Description automatically generated

A screenshot of a login box

Description automatically generated

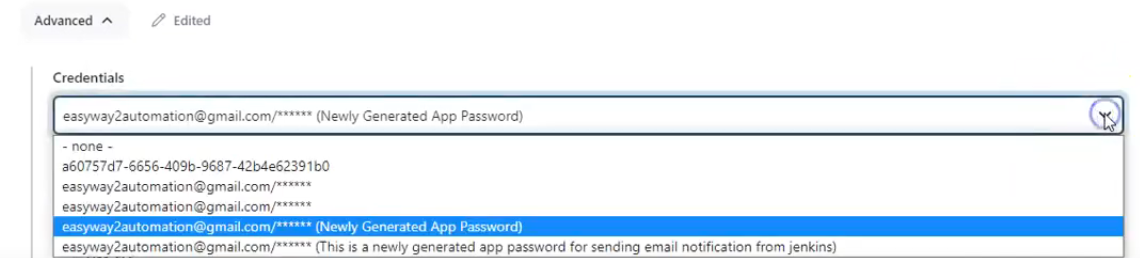
Switch back to Jenkins and update the password under advanced setting in email notification section.

A white and black line

Description automatically generated with medium confidence

Enter an gmail address and click test config and check whether we are getting success message

After this, go to extended email notification section, do the same like updating the smtp server, port, advanced setting 🡪 set add dropdown value to Jenkins 🡪 enter the username and password 🡪 description = some text about credentials and purpose 🡪 click add 🡪 then select the added credentials in the credentials dropdown🡪 select default content type = HTML(text/html) 🡪 add default recipients



Update the default subject for mail and default content as

A screenshot of a computer

Description automatically generated

May be we can write in the default content as the last line as “This is automated email, please do not reply”.

Set default triggers – always, success, failure-any and then click on apply and save buttons accordingly.