Robot Framework :-

<https://robotframework.org/>

<http://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html>



Install Robot Framework : pip install robotframework

To Install A Fresh Copy After Uninstalling : pip install –no-cache-dir robotframework

Upgrade : pip install –upgrade robotframework

Specific Version : pip install robot

All Python Libraries : pip freeze

All Python Libraries : pip list

Details Of Library : pip show robotframework

Check whether installed properly : pip check robotframework

-------------------------------------------------------------------------------------Install RED : Robot Editor Plugin from marketplace in Eclipse

<https://github.com/nokia/RED>

Similarly, If You Are Working On PyCharm. Install The Robot Framework Plugin.

Link To Install Plugin : <http://master.dl.sourceforge.net/project/red-robot-editor/repository>

Now, For Robot Framework To Work : We Must Have Python & Robot Framework Library Installed On Our Machine.

Then Install ‘RED’ eclipse plugin

Eclipse : preferences > robot framework > installed frameworks

Eclipse : perspective > open perspective > others > robot framework

So, Now We Will Open The Robot Perspective. Then Create a New Robot Project In Eclipse.

Robot Project Will Have Standard Robot Libraries & red.xml File

Create A Folder And Inside It Create A Test Suite.

\*\*\* Test Cases \*\*\* Is Header For The Test Case :-

\*\*\* Test Cases \*\*\*

**MyFirstTest**

**Log** Hello World

Run > output.xml, log.xml, report.html Will Be Generated.

Executing Via CMD : robot tests.robot

rebot output.xml

Now To Work With Selenium We Need To Install Selenium Libraries.

pip install --upgrade robotframework-seleniumlibrary (Remove –upgrade when installing for the 1st Time)

On Robot Framework Website, We Can Find A Number Of External Libraries Under External Library Menu.

<https://github.com/robotframework/SeleniumLibrary>

<http://robotframework.org/SeleniumLibrary/SeleniumLibrary.html> : Docs

Let’s Create A New Project :

Write Test Case:-

First Line will Be \*\*\* settings \*\*\*, Here We Will Import Required Library Into Out Test Case.

Note : In Robot Framework Min 2 Space Is Required To Distinct Between 2 Different Words.

\*\*\* Settings \*\*\*

Library SeleniumLibrary

If SeleniumLibrary Is Showing Red DownLine Color : Do CTRL+1. It Will Take Us Library Configuration. It Will Automatically Configure Else We Will Select Our Library From Python Library Folder.

C:\User\...\AppData\local\...\Python\lib\site packages\Selenium Library\\_\_init\_\_.py

Now, Write Test Cases:-

\*\*\* Settings \*\*\*

Library SeleniumLibrary

\*\*\* Test Cases \*\*\*

**MyFirstTestCase**

**Log** This Is My First Test Case level=INFO

**My Second Test Case**

**Log** This Is My Second Test Case Which Is First Selenium Test Case.

**Open Browser**

Note : To Run Our Selenium Test We Need Browser Drivers. We Will Need To Put Our Drivers Inside C:\User\..\AppData\Local\..\python\Scripts\BrowserDriver.exe

First TC :

*\*\*\* Settings \*\*\**

*Library SeleniumLibrary*

*\*\*\* Variables \*\*\**

*${URL} http://google.co.in/*

*${BROWSER} Chrome*

*\*\*\* Test Cases \*\*\**

***MyFirstTestCase***

***Log*** *This Is My First Test Case level=INFO*

***My Second Test Case***

***Log*** *This Is My Second Test Case Which Is First Selenium Test Case.*

***Open Browser*** *${URL} browser=${BROWSER}*

***Set Browser Implicit Wait*** *10*

***Maximize Browser Window***

***Go To*** *http://google.co.in*

***Input Text*** *name=q Krishna Poultry Agency*

***Sleep*** *2*

*# Press Keys name=q ENTER*

***Click Button*** *xpath://div[@class='FPdoLc VlcLAe']//input[@name='btnK']*

***Click Element*** *xpath://a[contains(text(),'THANKS')]*

Such styling adds more readability to the test case. Variables can be scalar (with the $ prefix), lists (with the @ prefix, dictionaries (with the & prefix) and environment (with the % prefix).

TC 002 :

*\*\*\* Settings \*\*\**

*Library SeleniumLibrary*

*Documentation Here We Will Try To Login In a Website*

*\*\*\* Variables \*\*\**

*${URL}* [*https://opensource-demo.orangehrmlive.com/*](https://opensource-demo.orangehrmlive.com/)

*${BROWSER} Chrome*

*\*\*\* Test Cases \*\*\**

***Sample Login Test***

*[Documentation] Test Case For Login*

***Open Browser*** *https://opensource-demo.orangehrmlive.com/ chrome*

***Maximize Browser Window***

***Set Browser Implicit Wait*** *5*

***Clear Element Text*** *id=txtUsername*

***Input Text*** *id=txtUsername Admin*

***Clear Element Text*** *txtPassword*

***Input Password*** *id=txtPassword admin123*

***Click Button*** *id=btnLogin*

***Click Element*** *id=welcome*

***Click Element*** *xpath://a[text()='Logout']*

***Close Browser***

***-------------------------------------------------------------------------------------***

**Variables :-**

**Scalar : ${url}** [**http://google.co.in**](http://google.co.in)

**Use As ${url}**

**List : @{Credentials} admin password@123**

**Use As @{Credentials}[index]**

**Dictionary : &{LoginData} username=admin password=password@123**

**Use As %{LoginData}[key]**

**Environment : %username%, %os%**

**Use As ‘This TC Has Been Executed By %username% On %os%**

**Built In :** [**http://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html**](http://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html)

**Using Custom Keyword :-**

**TC 003 :-**

*\*\*\* Settings \*\*\**

*Library SeleniumLibrary*

*Documentation Here We Will Try To Login In a Website*

*\*\*\* Variables \*\*\**

*${URL} https://opensource-demo.orangehrmlive.com/*

*${BROWSER} Chrome*

*@{Credentials} Admin admin123*

*&{LoginData} username=Admin password=admin123*

*\*\*\* Keywords \*\*\**

***Submit Login Details***

***Clear Element Text*** *id=txtUsername*

***Input Text*** *id=txtUsername @{Credentials}[0]*

***Clear Element Text*** *txtPassword*

***Input Password*** *id=txtPassword @{Credentials}[1]*

***Click Button*** *id=btnLogin*

*\*\*\* Test Cases \*\*\**

***Sample Login Test***

*[Documentation] Test Case For Login*

***Open Browser*** *${URL} ${BROWSER}*

***Maximize Browser Window***

***Set Browser Implicit Wait*** *5*

*# Using The Custom Keyword*

***Submit Login Details***

***Click Element*** *id=welcome*

***Click Element*** *xpath://a[text()='Logout']*

***Log*** *User Was Logged In As Username : &{LoginData}[username] and Password &{LoginData}[password]*

***Close Browser***

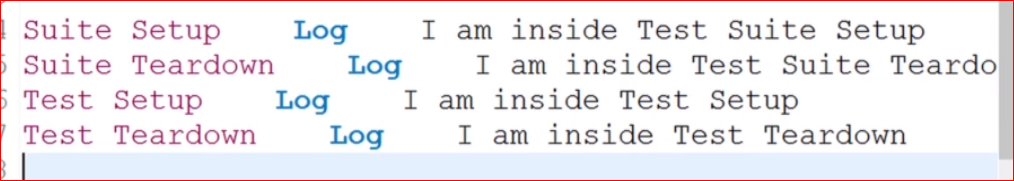
Some Hooks Used To Run Before Or After The Test Case Inside \*\*\* Settings \*\*\*

Suite Setup : Runs Before Suite

Test Setup : Runs Before Test

Suite Teardown : Runs After Suite

Test Teardown : Runs After Test



**Using Tags:**

We Can Use Tags Like Smoke, Sanity, Regression By Using [Tags] Above The Test Case.

We Can Also Create Dynamic Tags. By Keyword : ***Set Tags*** *Regression*

Also We Can Add Variables Inside Test Cases : ${username}= Set Variable admin

TC 004 :

*\*\*\* Settings \*\*\**

*Library SeleniumLibrary*

*Documentation Here We Will Try To Login In a Website*

*Test Setup* ***Log*** *Starting The Test Case*

*Test Teardown* ***Log*** *Completing The Test Case*

*Suite Setup* ***Log*** *Running The Test Suite*

*Suite Teardown* ***Log*** *Completing The Test Suite*

*Test Timeout 50000*

*Default Tags Normal*

*\*\*\* Variables \*\*\**

*${URL} https://opensource-demo.orangehrmlive.com/*

*${BROWSER} Chrome*

*@{Credentials} Admin admin123*

*&{LoginData} username=Admin password=admin123*

*\*\*\* Keywords \*\*\**

***Submit Login Details***

***Clear Element Text*** *id=txtUsername*

***Input Text*** *id=txtUsername @{Credentials}[0]*

***Clear Element Text*** *txtPassword*

***Input Password*** *id=txtPassword @{Credentials}[1]*

***Click Button*** *id=btnLogin*

*\*\*\* Test Cases \*\*\**

***MyFirstTestCase***

*[Tags] Sanity*

***Log*** *This Is My First Test Case level=INFO*

***My Second Test Case***

*[Tags] Sanity Smoke*

***Log*** *This Is My Second Test Case Which Is First Selenium Test Case.*

***Open Browser*** *${URL} browser=${BROWSER}*

***Set Browser Implicit Wait*** *10*

***Maximize Browser Window***

***Go To*** *http://google.co.in*

***Input Text*** *name=q Krishna Poultry Agency*

***Sleep*** *2*

*# Press Keys name=q ENTER*

***Click Button*** *xpath://div[@class='FPdoLc VlcLAe']//input[@name='btnK']*

***Click Element*** *xpath://a[contains(text(),'THANKS')]*

***Close Browser***

***Third Test Login Test***

*[Documentation] Test Case For Logins*

*[Tags] Sanity Smoke Regression*

***Open Browser*** *${URL} ${BROWSER}*

***Maximize Browser Window***

***Set Browser Implicit Wait*** *5*

*# Using The Custom Keyword*

***Submit Login Details***

***Click Element*** *id=welcome*

***Click Element*** *xpath://a[text()='Logout']*

***Log*** *User Was Logged In As Username &{LoginData}[username] and Password &{LoginData}[password]*

***Close Browser***

***Fourth Test***

**Open Browser** ${url} chrome

${keyword} **Set Variable** Hello World

***Set Tags*** *Regression2*

**Input Text** name=q ${keyword}

**Press Keys** xpath://body TAB

**Click Button** xpath://div[@class='FPdoLc VlcLAe']//input[@name='btnK']

**Execute Javascript** window.scrollTo(0,1000)

**Sleep** 5

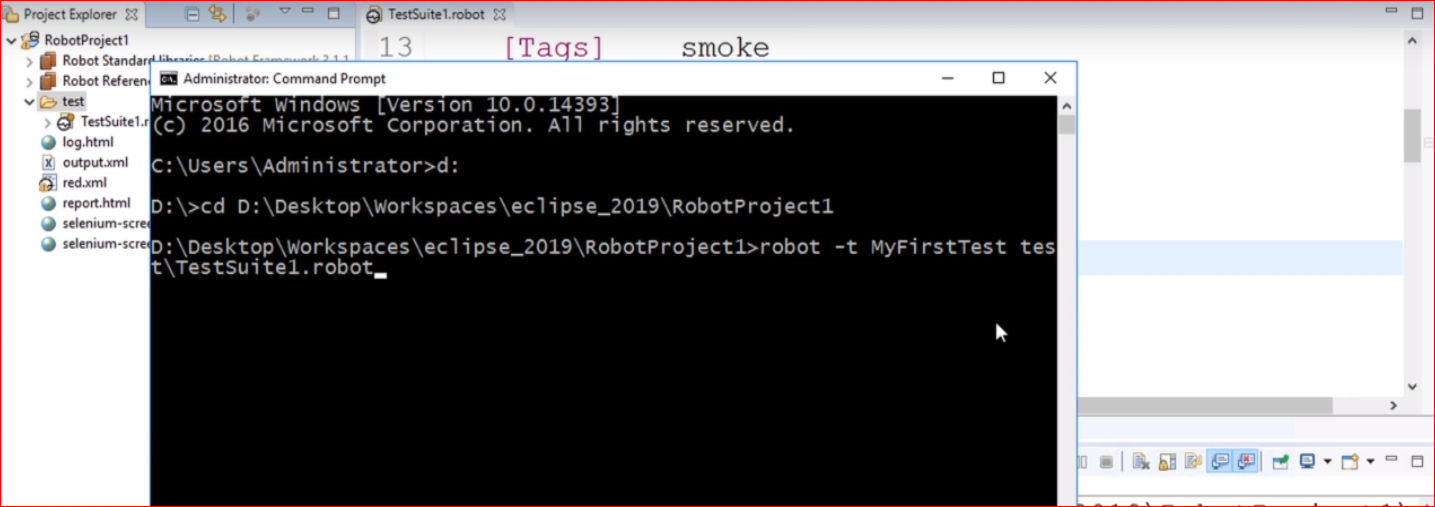
${title} **Get Title**

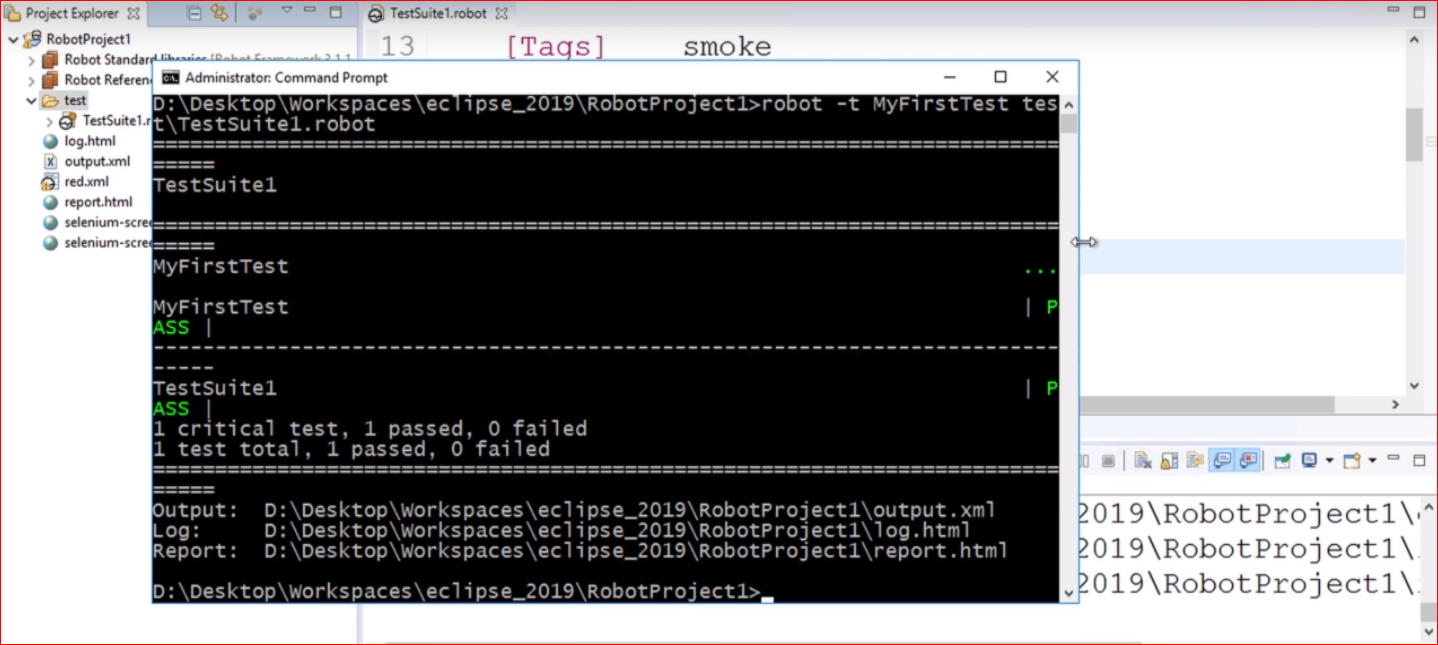
**Log** ${title}

**Log To Console** ${title}

**Close Browser**

Command Line Test Run





Setting Tag In CMD :

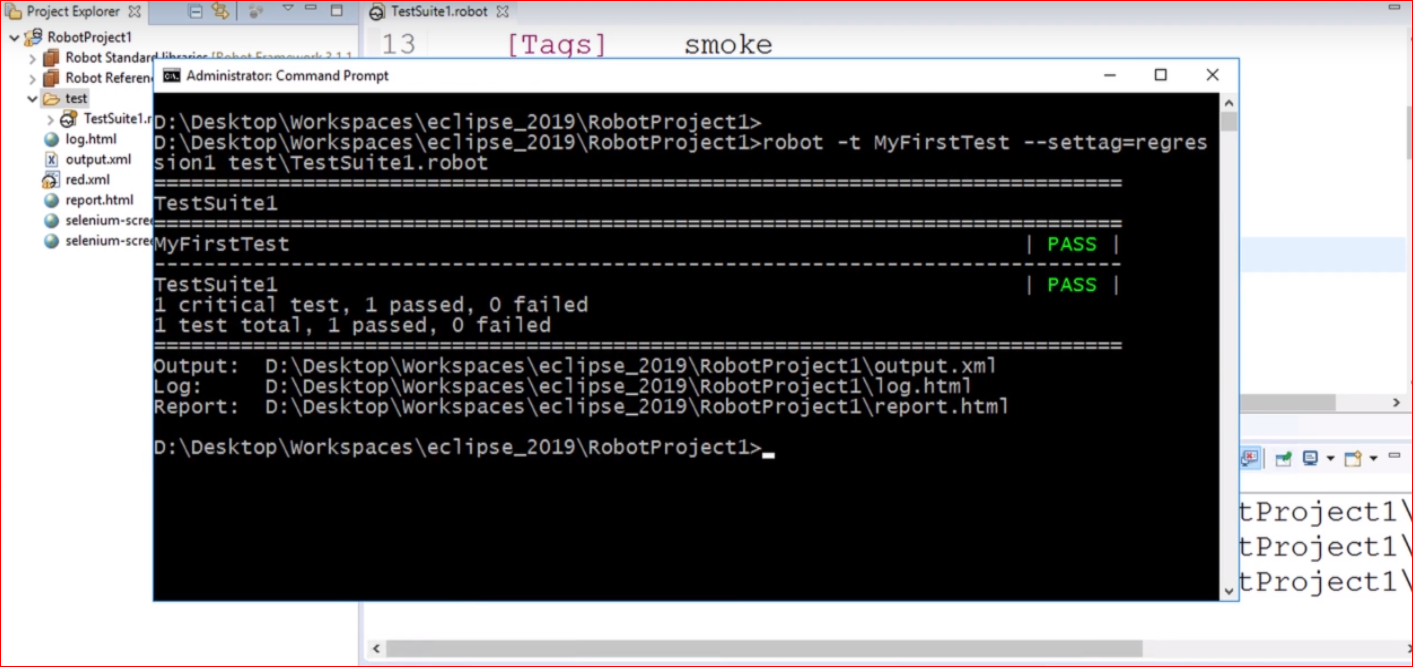
Running Complete Suite



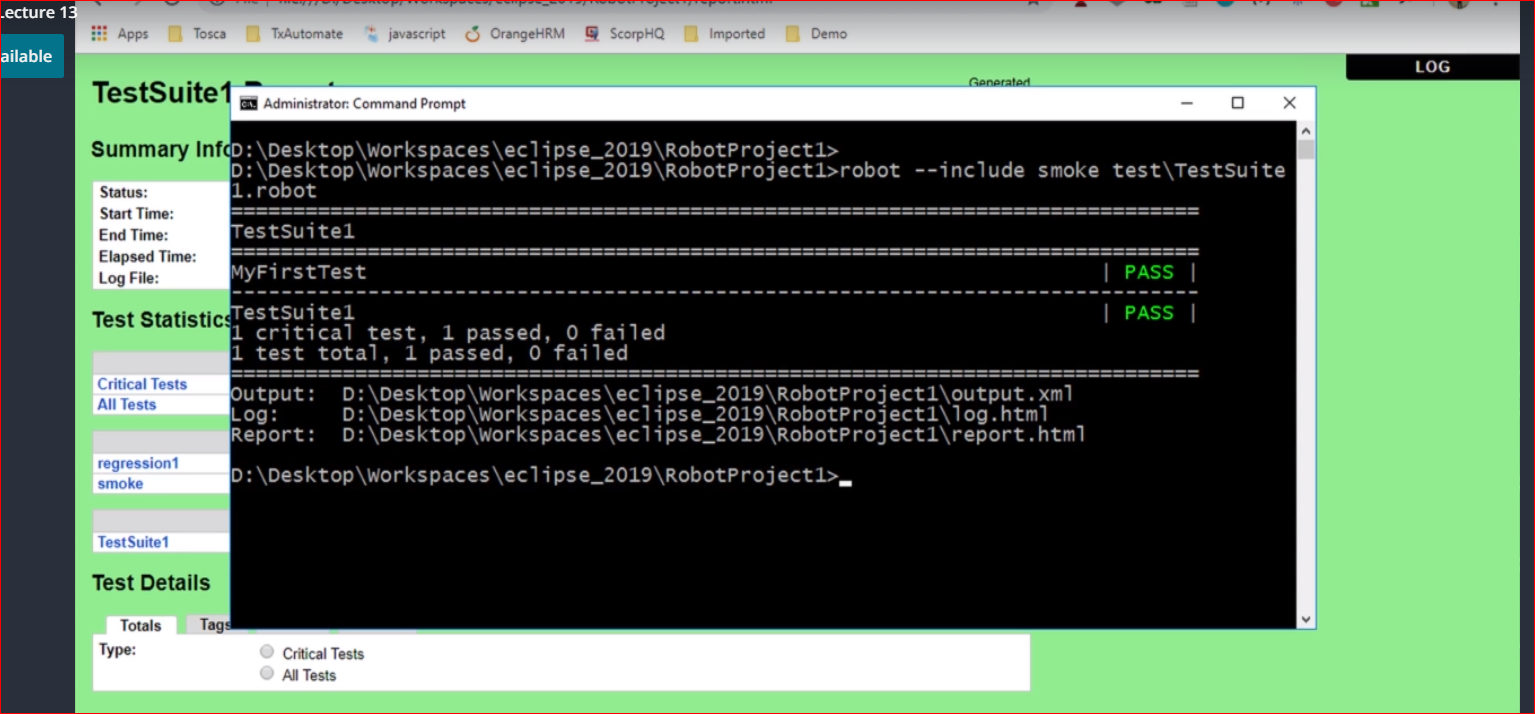
Running Some Test Cases Of Suite

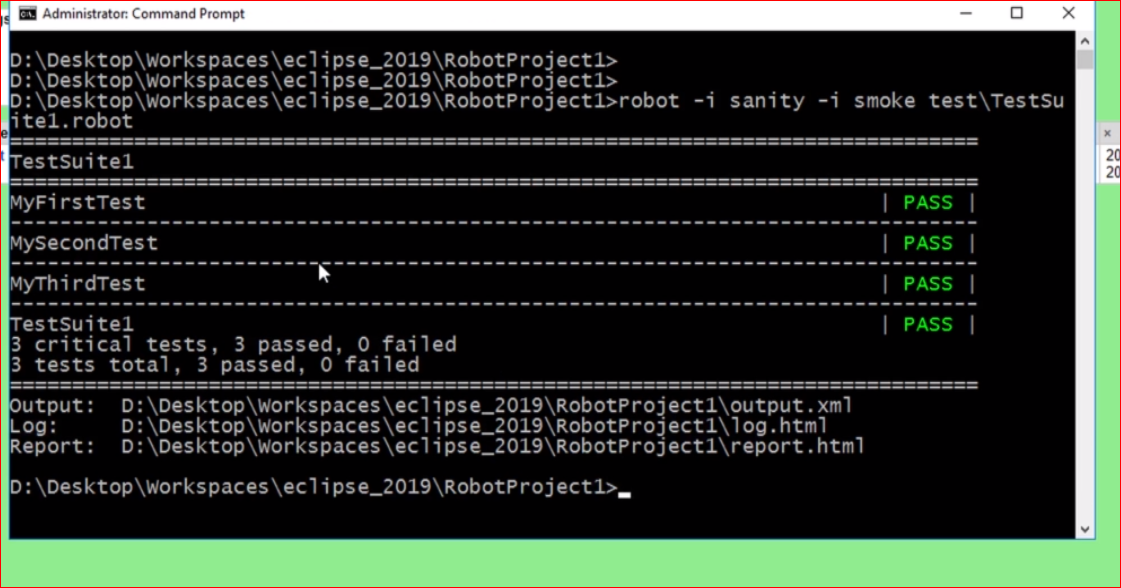


Setting Tag With Run

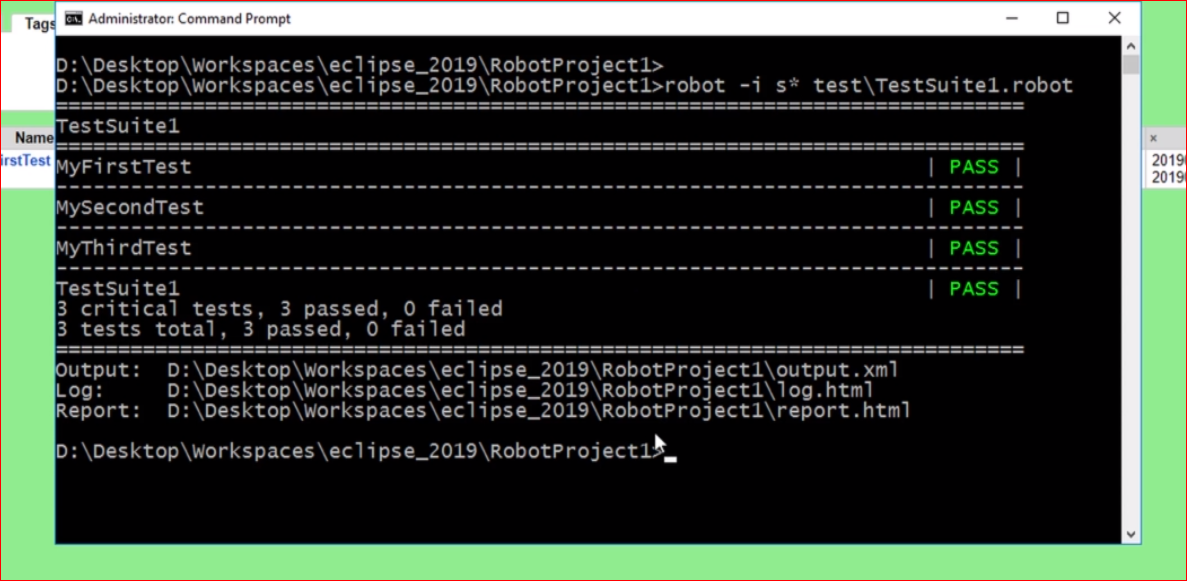


Running Only Desired Test With Tag.

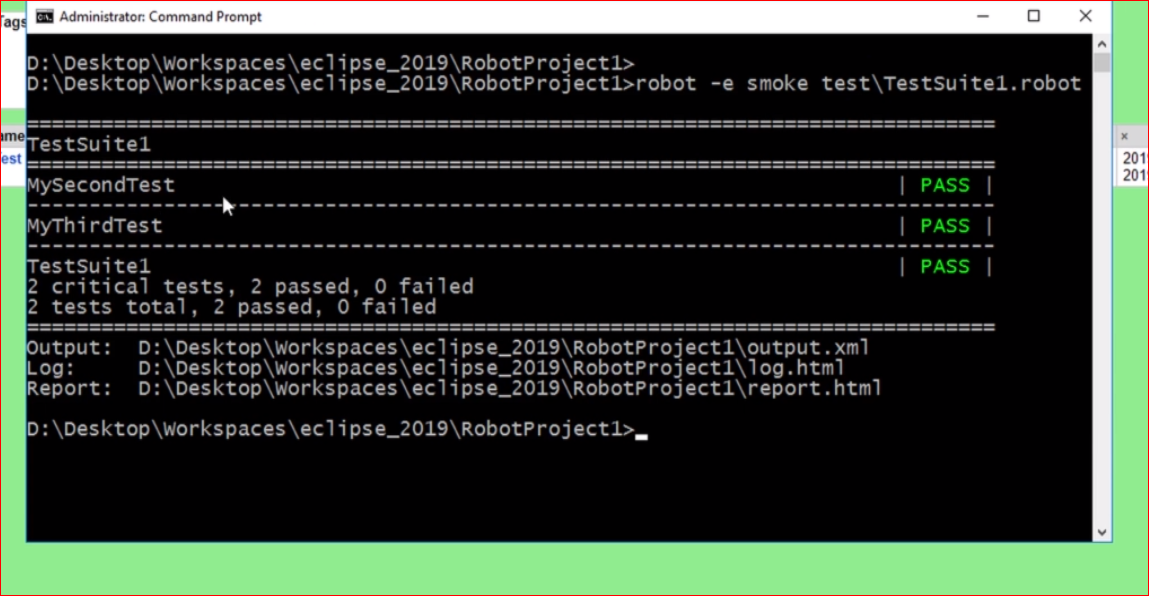




**Running All Tags That Starts With ‘S’**



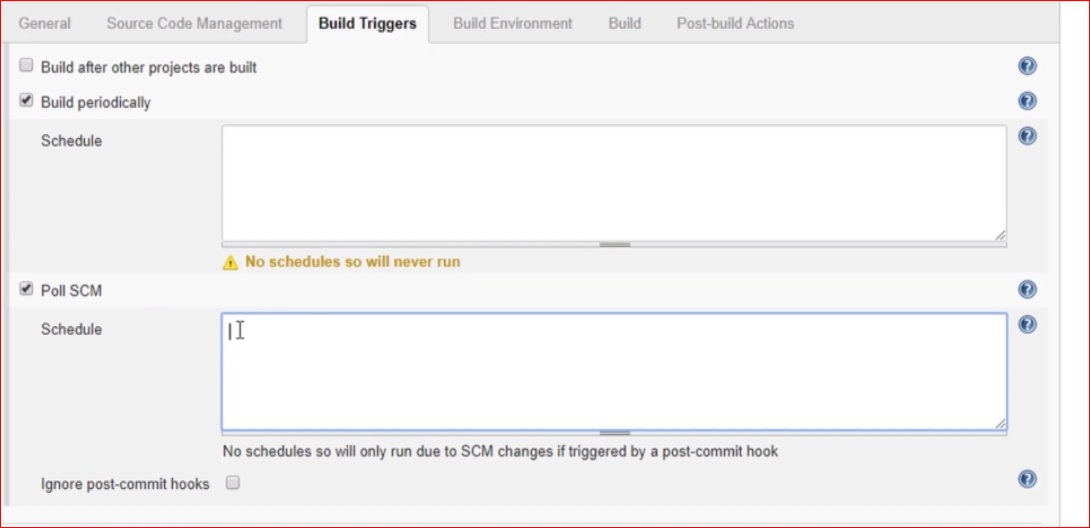
**Excluding Particular Tags From Test**



**Jenkins :-**

**Setting Interval Of Build Run.**

**Or Set Poll SCM Which Checks The Git Commit. If There Is Any New Commit. It Will Run The Build.**



**Post Build actions**

