Setup Jenkins with Docker on Mac

Setup Jenkins with Docker on Mac

Environment

```
OS: macOS High Sierra 10.13
```

Download Docker for Mac

Create Docker ID

You need to go to https://www.docker.com/ to create you own docker id.

Download Jenkins CI with blue ocean plugin.

Download link: https://hub.docker.com/r/jenkinsci/blueocean/ .

Follow the installation instructions to install the **jenkinsci/blueocean** Docker container. And do remember to launch the Docker.app after installation as launching the application is a necessary step to get the Docker daemon run.

Start the Jenkins CI container

```
$ docker run -u root --rm -d -p 8080:8080 -p 50000:50000 -v jenkins-data:/var/jenkins_home -v /var/run/docker.sock:/var/run/docker.sock jenkinsci/blueocean
```

Command options

--rm
is to remove the docker container after it is shut down.

-d
is to let jenkinsci container run in background and silent the output rather than print out in terminal console. **Recommend not to add this option, because usually you want to check the output from terminal console, e.g. the initial password.**

-p 8080:8080
is the to map the port number for host machine and docker container respectively.

-p 50000:50000
is for the JNLP based Jenkins master-slave configurations. JNLP-based Jenkins agents communicate with the Jenkins master through TCP port 50000 by default.

-v jenkins-data:/var/jenkins_home
is to map the /var/jenkins_home directory in the container to the Docker volume with the name jenkins-data, this option is to ask Jenkins to persist state for instance restarts. -v /var/run/docker.sock:/var/run/docker.sock, this option is to allow jenkins ci container to communicate with docker daemon. **jenkinsci/blueocean** is the jenkinsci image with blueocean plugin, specifying this option will automatically pull any updates.

Possibile errors

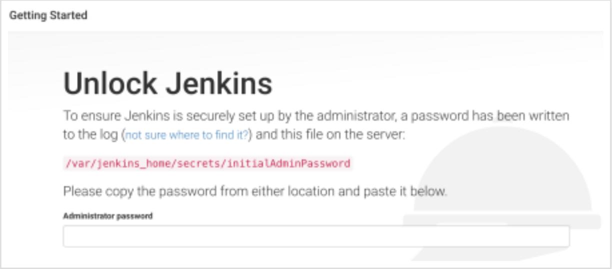
```
docker: Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?.

See 'docker run --help'.
```

This error is because the Docker daemon is not started yet. On MacOS, you need to manually start the Docker.app in order to let the docker daemon start.

Open the Jenkins landing page

Browse http://localhost:8080/ , you will see below page.



Getting Started

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:

/var/jenkins_home/secrets/initialAdminPassword

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

Administrator password

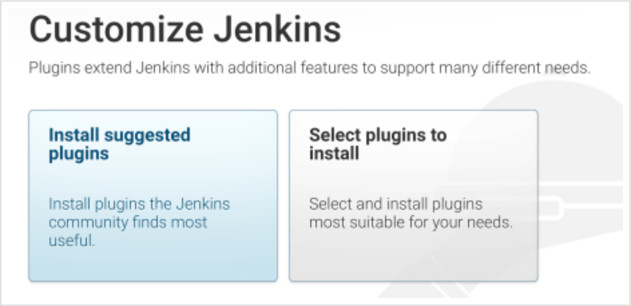
Find the password from console output and paste it above input box to continue.

If you suddenly lost your initial password from the console output. You can use below command to get it back. Also, actually this password is your default password for your Jenkins **admin** account.

```
$ docker exec cat /var/jenkins_home/secrets/initialAdminPassword
```

Initial customisation

You can ask the Jenkins to install the suggested plugins for you or choose the ones you need only. You can also install the plugins from "**Manage Plugins**" after you complete the Jenkins setup.



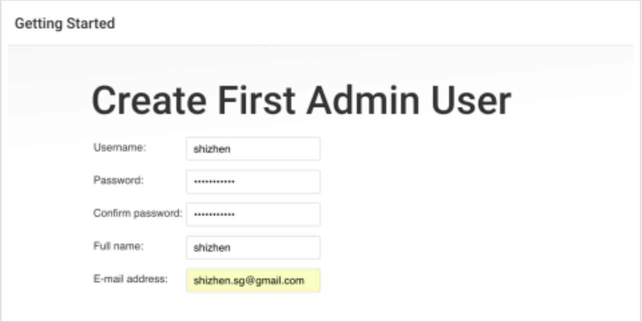
Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins
Install plugins the Jenkins community finds most useful.
Select plugins to install
Select and install plugins most suitable for your needs.

Create your first jenkins account

You can create your own jenkins account during the setup phase. Or you can also ignore this step but use the "**admin**" account with its **default password, the one output in console,** for login.



Getting Started

Create First Admin User

Username: shizhen

Password: •••••••••••

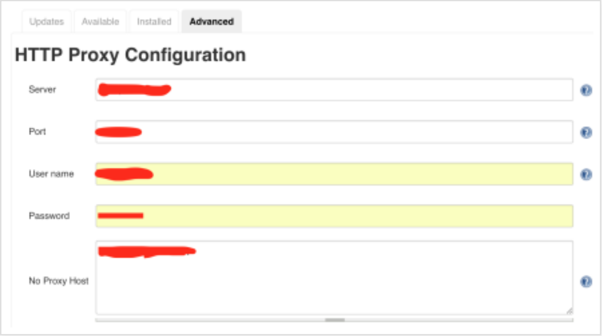
Confirm password: •••••••••••

Full name: shizhen

E-mail address: shizhen.sg@gmail.com

Proxy Setup

If your Jenkins server is behind a proxy, then you need to configure this proxy properly in order to install your Jenkins plugins. Navigate yourself to "**Manage Jenkins**" -> "**Manager Plugins**" -> "**Advanced**".



HTTP Proxy Configuration

Server [redacted]

Port [redacted]

User name [redacted]

Password [redacted]

No Proxy Host [redacted]

Some useful commands

Check container IDs

```
docker ps -a
```

Check logs for a particular container

```
docker logs <container-id>
```

By Naveen AutomationLabs - November 14, 2019

Reactions:

No comments:

Test cases for Online Shopping Application

Test cases for Online Shopping Application

General Test Cases

1. Verify that user is able to navigate through all the products across different categories
2. Verify that all the links and banners are redirecting to correct product/category pages and none of the links arr broken
3. Verify that the company logo is clearly visible

4. Verify that all the text - product, category name, price and product description are clearly visible
5. Verify that all the images - product and banner are clearly visible
6. Verify that category pages have relevant product listed specific to the category
7. Verify that correct count of total products are listed on the category pages
8. Search - Verify that on searching all the product satisfying the search criteria are visble on the search result page
9. Search - Verify the more relevant product for the search term are displayed on the top for a particular search term
10. Search - Verify that count of products is correctly displayed on the search result page for a particular search term
11. Filtering - Verify that filtering functionality correctly filters product based on the filter applied
12. Filtering - Verify that filtering works correctly on category pages
13. Filtering - Verify that filtering works correctly on the search result page
14. Filtering - Verify that correct count of total products is displayed after a filter is applied
15. Sorting - Verify that all the sort options work correctly - correctly sort the products based on the sort option chosen
16. Sorting - Verify that sorting works correctly on the category pages
17. Sorting - Verify that sorting works correctly on the search result page
18. Sorting - Verify that sorting works correctly on the pages containing filtered result, after applying filters
19. Sorting - Verify that product count remains intact irrespective of sorting option applied

Product Buy Flow - Test cases

1. Verify that on the product page, user can select the desired attribute of the product e.g. size, color etc
2. Verify that user can add to cart one or more products
3. Verify that user can add products to wishlist
4. Verify that user can buy products added to cart after signing in to the application (or as per the functionality of the website)
5. Verify that user can successfully buy more than one products that were added to his/her cart
6. Verify that user cannot add more than available inventory of the product
7. Verify that the limit to the number of products a user can by is working correctly by displaying error message and preventing user from buying more than the limit
8. Verify that the delivery can be declined of for the places where shipping is not available
9. Verify that Cash on Delivery option of payment is working fine
10. Verify that the different pre-paid methods of payments are working fine
11. Verify that product return functionality works fine

User(Buyer) Registration - Test cases

1. Verify that all the specified fields are present on the registration page
2. Verify that the required/mandatory fields are marked with * against the field
3. Verify that for better user interface dropdowns, radio buttons and checkboxes etc fields are displayed wherever possible instead of just textboxes
4. Verify the page has both submit and cancel/reset buttons at the end
5. Verify that clicking submit button after entering all the required fields, submits the data to the server
6. Verify that clicking cancel/reset button after entering all the required fields, cancels the submit request and resets all the fields
7. Verify that whenever possible validation should take place at client side
8. Verify that not filling the mandatory fields and clicking submit button will lead to validation error
9. Verify that not filling the optional fields and clicking submit button will still send data to server without any validation error
10. Check the upper limit of the textboxes
11. Check validation on date and email fields (only valid dates and valid email Ids should be allowed
12. Check validation on numeric fields by entering alphabets and special characters
13. Verify that leading and trailing spaces are trimmed
14. Verify that entering blank spaces on mandatory fields lead to validation error
15. Verify that after making a request to the server and then sending the same request again with the same unique key will lead to server side validation error

Seller - Product creation Test cases

1. Verify that authenticated sellers get access to product creation panel specific to the authorised categories
2. Verify that product creation panel is working fine for single product creation
3. Verify that product creation panel is working fine for multiple product creation
4. Verify that maximum product creation limit for seller is working fine, limiting seller to create more than the desired number of products
5. Verify panel validation for checking mandatory fields
6. Verify that duplicate product creation is restricted through panel
7. Verify that seller can update information and price of existing products
8. Verify that product created by seller get visible on the website after certain period of time
9. Verify that updation made by seller get visible on the website after certain period of time

By Naveen AutomationLabs - October 20, 2019

Reactions:

No comments: ✉

Labels: Manual Test Cases, Manual Testing

Test Cases of Date field

Test Cases of Date field

1. Verify that on clicking the datefield a calendar widget should open.
2. Verify that the default width of the calendar should be displayed as per the specification.
3. Verify that user can select a date in the calendar and after slecting the date the same gets displayed in the date field.
4. Verify that by default current month's calendar should be displayed.
5. Verify that user can move to previous and next month's calendar by choosing the left and right icon over the calendar.
6. Verify that user can check a specific year's calendar.
7. Verify that user enter date manually in the date following the date format.
8. Verify that user can edit a date set by choosing the one selected from the calendar.
9. Verify that values other than numerics should not be allowed in the date field(apart from the charcters used in the date format like '/' or '-'.
10. Verify that invalid dates are not allowed in the date field (like date value exceeding 31, month value exceeding 12 etc).

By Naveen AutomationLabs - October 20, 2019

Reactions:

5 comments: ✉

Labels: Manual Test Cases, Manual Testing

Test Cases of Calculator

Test Cases of Calculator

1. Check if the calculator is normal calculator or scientific calculator
2. Verify that all the buttons are present and text written on them is readable
3. Check the arithmetic operations are working fine- +, -, /, * etc
4. Verify that BODMAS is applied in case of complex queries and correct result is returned
5. Verify that calculator gives correct result in case of operations containing decimal numbers
6. Check if the calculator is battery operated or works on solar power
7. Verify the outer body material of the calculator
8. Verify the spacing between the two buttons, the buttons should not be too closely placed
9. Check the pressure required to press a button, the pressure required should not be too high
10. Verify the number of digits allowed to enter in the calculator for any operation
11. Verify the limit of the response value

12. Verify the functioning of memory functions
13. Check if calculator allows to navigate through previous calculations
14. Verify that hitting 'C' cancels any digits or operation added
15. Verify the working of ON-OFF button in the calculator
16. Check if keeping the calculator unused for certain period of time, turns it off automatically
17. Verify that on pressing two operators one after the other, the latest one will override the previous operator
18. Verify the state of calculator when two buttons are pressed simultaneously
19. Verify if user can delete a digits one by one using backspace key

By Naveen AutomationLabs - October 19, 2019

Reactions:

No comments: ✉

Labels: Manual Test Cases, Manual Testing

Test Cases of ATM Machine

Test Cases of ATM Machine

1. Verify the slot for ATM Card insertion is as per the specification
2. Verify that user is presented with options when card is inserted from proper side
3. Verify that no option to continue and enter credentials is displayed to user when card is inserted correctly
4. Verify that font of the text displayed in ATM screen is as per the specifications
5. Verify that touch of the ATM screen is smooth and operational
6. Verify that user is presented with option to choose language for further operations
7. Verify that user asked to enter pin number before displaying any card/bank account detail
8. Verify that there are limited number of attempts up to which user is allowed to enter pin code
9. Verify that if total number of incorrect pin attempts gets surpassed then user is not allowed to continue further- operations like blocking of card etc gets initiated
10. Verify that pin is encrypted and when entered
11. Verify that user is presented with different account type options like- saving, current etc
12. Verify that user is allowed to get account details like available balance
13. Verify that user same amount of money gets dispatched as entered by user for cash withdrawal
14. Verify that user is only allowed to enter amount in multiples of denominations as per the specifications
15. Verify that user is prompted to enter the amount again in case amount entered is not as per the specification and proper message should be displayed for the same
16. Verify that user cannot fetch more amount than the total available balance
17. Verify that user is provided the option to print the transaction/inquiry
18. Verify that user user's session timeout is maintained and is as per the specifications
19. Verify that user is not allowed to exceed one transaction limit amount
20. Verify that user is not allowed to exceed one day transaction limit amount
21. Verify that user is allowed to do only one transaction per pin request
22. Verify that user is not allowed to proceed with expired ATM card
23. Verify that in case ATM machine runs out of money, proper message is displayed to user
24. Verify that in case sudden electricity loss in between the operation, the transaction is marked as null and amount is not withdrawn from user's account

By Naveen AutomationLabs - October 19, 2019

Reactions:

No comments: ✉

Labels: Manual Test Cases, Manual Testing

Take Screenshot of WebElement in Selenium || New Selenium 4 feature - Naveen AutomationLabs

Take Screenshot of WebElement in Selenium || New Selenium 4 feature

In this blog, I have explained how to take screenshot for a web element in Selenium 4 version. This is the new feature in Selenium 4, earlier we could only take screenshot only for web page, now you can take screenshot for elements and use them with your bug report.

Watch Full video here:

Source Code:

```
package WM;

import java.io.File;
import java.io.IOException;
import java.util.concurrent.TimeUnit;

import org.apache.commons.io.FileUtils;
import org.openqa.selenium.By;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

import io.github.bonigarcia.wdm.WebDriverManager;

public class TakeElementScreenshot {

 public static void main(String[] args) throws IOException {

  WebDriverManager.chromedriver().setup();
  WebDriver driver = new ChromeDriver();
  driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

  driver.get("https://app.hubspot.com/login");

  WebElement email = driver.findElement(By.id("username"));
  WebElement password = driver.findElement(By.id("password"));
  WebElement loginButton = driver.findElement(By.id("loginBtn"));
  WebElement forgotPwd = driver.findElement(By.linkText("Forgot my password"));

  email.sendKeys("naveen@gmail.com");
  password.sendKeys("Test@123");

  takeElementScreenshot(email, "emailElement");
  takeElementScreenshot(password, "passwordElement");
  takeElementScreenshot(loginButton, "loginButtonElement");
  takeElementScreenshot(forgotPwd, "forgotPwdLink");

  driver.quit();

 }

//take screenshot using getScreenshotAs() method.

 public static void takeElementScreenshot(WebElement element, String fileName) {

  File srcFile = element.getScreenshotAs(OutputType.FILE);
  try {
   FileUtils.copyFile(srcFile, new File("./target/screenshots/" + fileName + ".png"));
  } catch (IOException e) {
   e.printStackTrace();
  }

 }

}
```

By Naveen AutomationLabs - October 16, 2019

Reactions:

No comments:

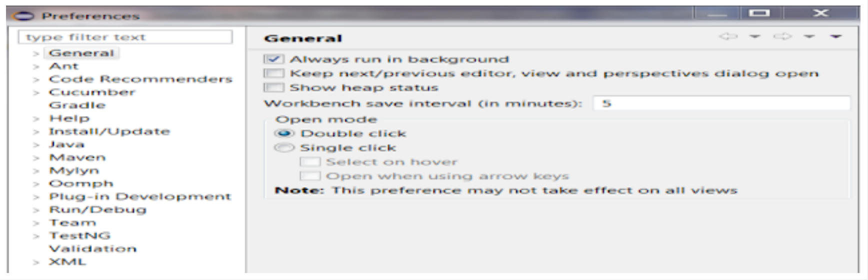
Labels: Selenium, Selenium 4

Executing Maven Project in windows from Command Prompt:

Executing Maven Project in windows from Command Prompt

Section-I:
Pre-requisites:
Go to Menu Bar in Eclipse >> Window >> Preferences.

Check if Maven is integrated with Eclipse or No.
[If you download Eclipse Neon Version, by default Maven will be integrated with Eclipse].



General

☑ Always run in background
□ Keep next/previous editor, view and perspectives dialog open
□ Show heap status

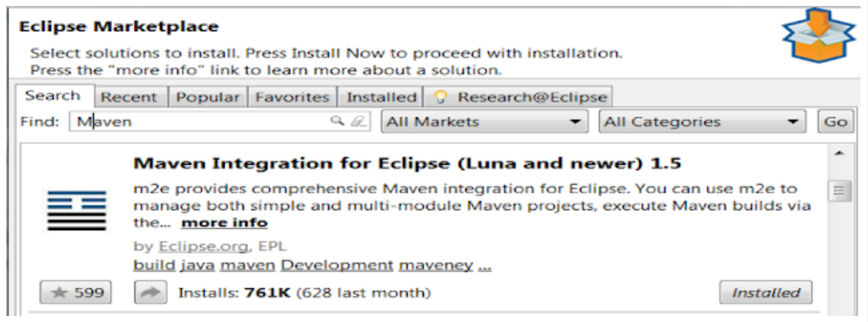
Workbench save interval (in minutes): 5

Open mode
● Double click
○ Single click
□ Select on hover
□ Open when using arrow keys

Note: This preference may not take effect on all views

If Maven is integrated with Eclipse then we are good to create Maven Project.

If not installed go to Menu Bar >> Help >> Eclipse Market Place and Search by Maven and install required Maven Plugins shown below.



Eclipse Marketplace

Select solutions to install. Press Install Now to proceed with installation.
Press the "more info" link to learn more about a solution.

Find: Maven

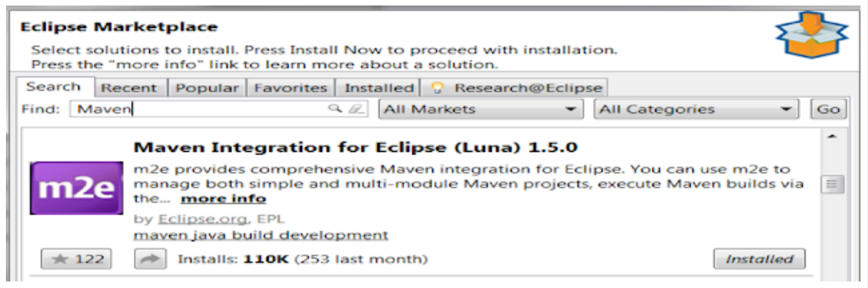
Maven Integration for Eclipse (Luna and newer) 1.5

m2e provides comprehensive Maven integration for Eclipse. You can use m2e to manage both simple and multi-module Maven projects, execute Maven builds via the... **more info**

by Eclipse.org, EPL

build java maven Development maveney ...

Installs: **761K** (628 last month)



Eclipse Marketplace

Select solutions to install. Press Install Now to proceed with installation.
Press the "more info" link to learn more about a solution.

Find: Maven

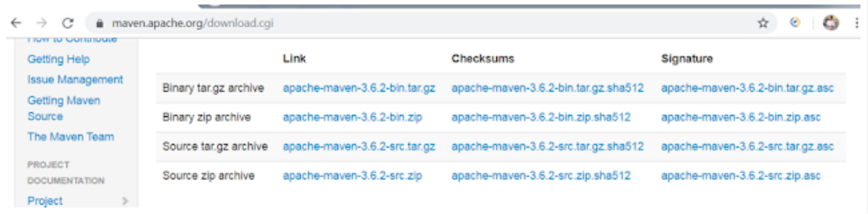
Maven Integration for Eclipse (Luna) 1.5.0

m2e provides comprehensive Maven integration for Eclipse. You can use m2e to manage both simple and multi-module Maven projects, execute Maven builds via the... **more info**

by Eclipse.org, EPL

maven java build development

Installs: **110K** (253 last month)



maven.apache.org/download.cgi

	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.6.2-bin.tar.gz	apache-maven-3.6.2-bin.tar.gz.sha512	apache-maven-3.6.2-bin.tar.gz.asc
Binary zip archive	apache-maven-3.6.2-bin.zip	apache-maven-3.6.2-bin.zip.sha512	apache-maven-3.6.2-bin.zip.asc
Source tar.gz archive	apache-maven-3.6.2-src.tar.gz	apache-maven-3.6.2-src.tar.gz.sha512	apache-maven-3.6.2-src.tar.gz.asc
Source zip archive	apache-maven-3.6.2-src.zip	apache-maven-3.6.2-src.zip.sha512	apache-maven-3.6.2-src.zip.asc

Section II:

Steps to install Maven Jar Files and Setting Path in Environment Variables: https://maven.apache.org/download.cgi

Download below highlighted Zip File.

Once download is completed, unzip it and keep it in any drive in your local system.

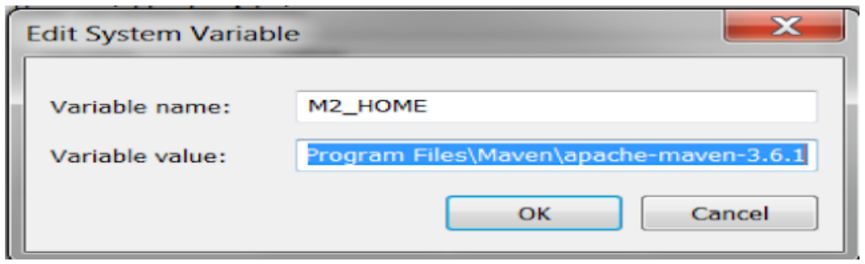
It is always good practice to keep it in C Drive.

Now we need to set path for Maven in Environment Variables:

Go to My Computer >> Right Click >> Properties >> Advanced System Settings.

Now Go to Environment Variables and Set Pat as shown below.

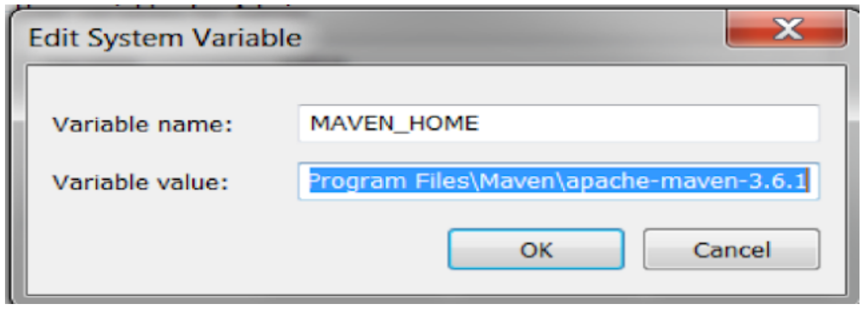
[In my case i have kept Maven Jars in C Drive]



Edit System Variable

Variable name: M2_HOME

Variable value: Program Files\Maven\apache-maven-3.6.1



Edit System Variable

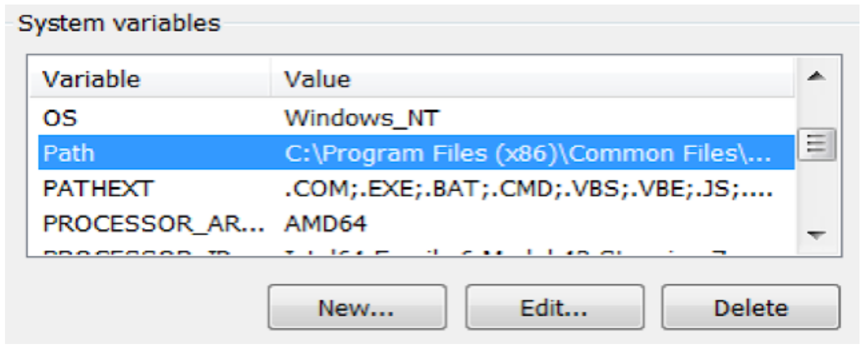
Variable name: MAVEN_HOME

Variable value: Program Files\Maven\apache-maven-3.6.1

Now Add Bin Path of Maven in below Path

highlighted

C:\Program Files\Maven\apache-maven-3.6.1\bin

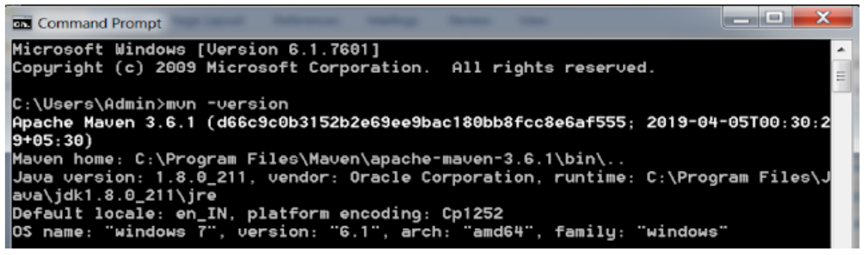


System variables

Variable	Value
OS	Windows_NT
Path	C:\Program Files (x86)\Common Files\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_AR...	AMD64

Once Path is set for Maven in Environment Variables.

Go to Command Prompt and type command as shown below to make sure that Maven is installed or not.



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

C:\Users\Admin>mvn -version
Apache Maven 3.6.1 (d66c9c0b3152b2e69ee9bac180bb8fcc8e6af555; 2019-04-05T00:30:2
9+05:30)
Maven home: C:\Program Files\Maven\apache-maven-3.6.1\bin\..
Java version: 1.8.0_211, vendor: Oracle Corporation, runtime: C:\Program Files\J
ava\jdk1.8.0_211\jre
Default locale: en_IN, platform encoding: Cp1252
OS name: "windows 7", version: "6.1", arch: "amd64", family: "windows"
```

If Maven is installed then we will see as highlighted above and now we are good to execute Maven Project.

Section III:

Now Executing Maven Project from Command Prompt.

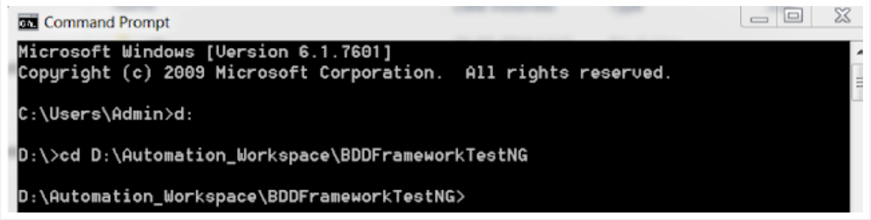
To Execute Maven Project from Command Prompt, In pom.xml

Maven Surefire Plugin must be added.

```
<build>
        <plugins>
                        <plugin>
                                    <groupId>org.apache.maven.plugins</groupId>
                                    <artifactId>maven-surefire-plugin</artifactId>
                                    <version>2.18.1</version>
                                    <configuration>
                                                <suiteXmlFiles>
                                                            <suiteXmlFile>testng_sanity.xml</suiteXmlFile>                                                                                             </su
                                    </configuration>
                        </plugin>
            </plugins>
</build>
```

Now Go to Command Promp

Go to Project Folder as shown below



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

C:\Users\Admin>d:

D:\>cd D:\Automation_Workspace\BDDFrameworkTestNG

D:\Automation_Workspace\BDDFrameworkTestNG>
```

Now Use below commands to execute Maven Project from

Command Prompt

1. mvn clean install : It will execute all test cases as well as it will generate a build. Build will be generated under target folder.

2. mvn clean test : It will execute only test cases and it won't generate build.

This blog is written by Pavan. He is an expert in Selenium Automation and currently working in Infosys as Automation QA Engineer. Also, he is helping QA community with his knowledge along with Naveen AutomationLabs!!

Cheers!

Naveen AutomationLabs