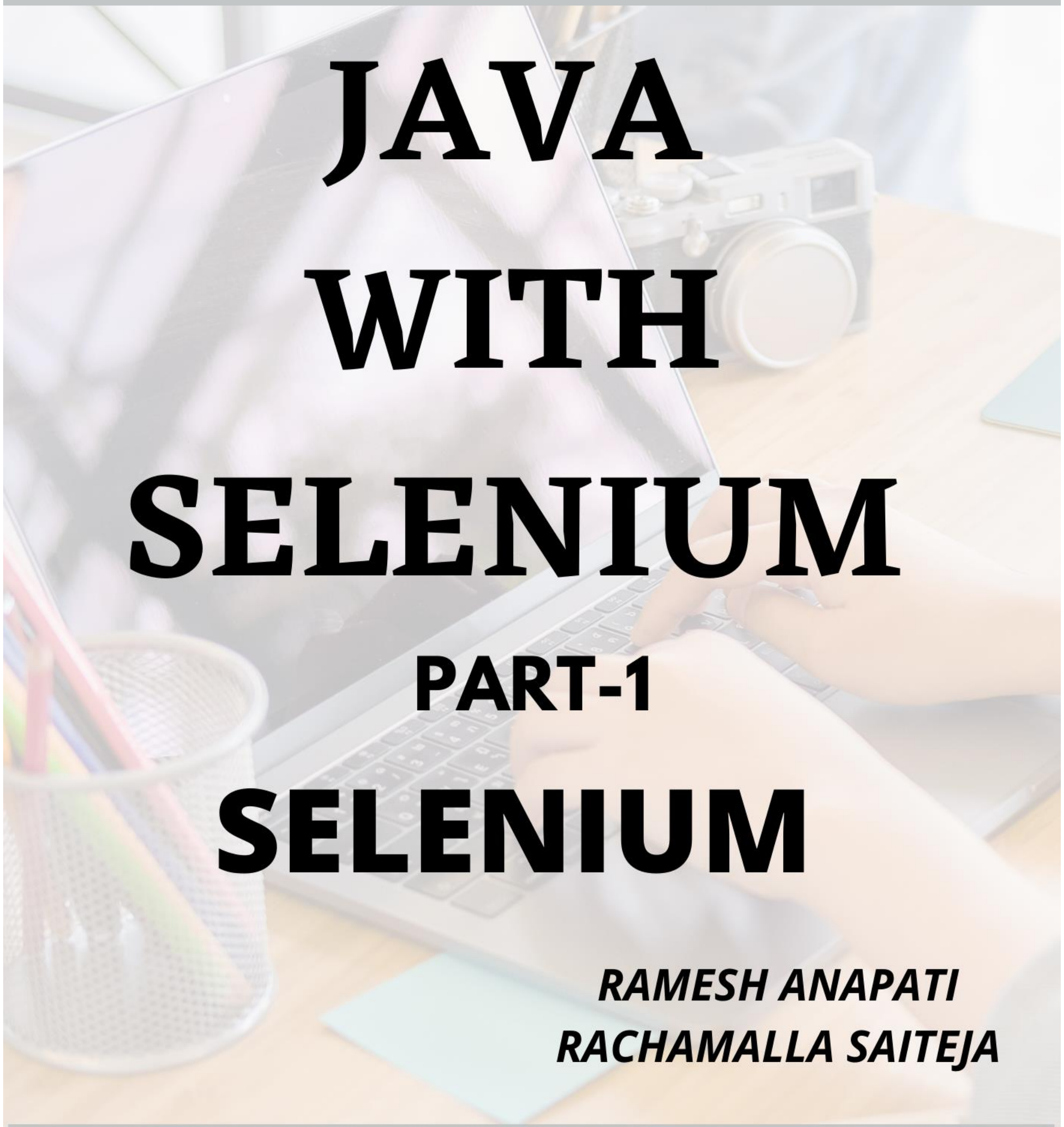
RAMESHSOFT

SOFTWARE TRAINING INSTITUTE JOB CONSULTANCY



JAVA WITH SELENIUM

PART-1

SELENIUM

RAMESH ANAPATI
RACHAMALLA SAITEJA

SR
edu publications
Publisher & Distributors

ISBN 978-93-92941-09-2

JAVA WITH SELENIUM

IN REAL TIME

SELENIUM

PART-2

RAMESH ANAPATI

RACHAMALLA SAITEJA

SR edu publications
Publisher & Distributors

Title of the Book: JAVA WITH SELENIUM IN REALTIME – SELENIUM(PART-2)

Edition: First- 2022

ISBN: 978-93-92941-09-2

ISBN 978-93-92941-09-2

9 789392 941092

SR edu publications

Publisher and Distributor

12-51/d/13/a, Kalyan Nagar

Kalwakurthy, Telangana-509324

Phone: +919553211467

Email: sredupublications@gmail.com

Website: www.Sredupublications.com

JAVA WITH SELENIUM

In Real time

INDIA'S NO1 REAL TIME TRAINING INSTITUTE
Rameshsoft
9177791456
SOFTWARE TRAINING INSTITUTE

Contents:

- SELENIUM WEBDRIVER
- Actions
- TestNg
- Maven
- GIT
- Jenkins

SELENIUM

Automation Testing: -

Automation testing means write once and execute any no. of times.

Java with selenium:-

By using selenium, we can do web application testing.

- Selenium supports different languages like
  - Java
  - Python
  - c#
  - ruby
  - Java Script

- Different Kinds of Modules:
  1. Selenium RC (Outdated)
  2. Selenium IDE (Outdated)
  3. selenium WebDriver (Present trending)
  4. Selenium grid

In order to identify the elements / to find the elements

- We have a predefined interface called Search context.
- Search context is an interface which is specially designed to identifying the elements.
- To identify the elements in web page we have element method called findelement.
  [webelement findElements(By by);]
- **find element** is a method which will identify the elements and which return in the form of element.
- To identify the group of elements we have a method called **findelements.**
  [List<webElements. findelements (By by);]
- webDriver is an interface which internally extends the search context.

```
interface webDriver extends searchcontext
{

}
```

- webDriver is an interface which is specially designed to perform the actions whatever we do in browser page.

```
Interface searchcontext
{
webElement findElement (By by);
List<webElements> findElements (By by);
}
Interface webDriver extends searchcontext
{

}
```

- webDriver is interface which is child class of search context.

In webDriver we have n no. of methods

```
Void get(String url);               //To enter URL
String getTitle( );                 //To get title of Page
String getwindowhandle( );          //To get window name
String getPageSource( );            //To get page source
Set<String> getwindowHandles( );   //To get all windows
Void close( );                      //To close current focus window
Void quit( );                       // To quit
String getcurrentUrl( );            // To get the URL
TargetLocator Switchto( );          // Switching mechanism
Options manage( );
Navigation navigate(  );
```

Implementations:-

There are three implementation classes we have.

```
Class RemoteWebDriver implements webDriver
{
;;;;;;;;;;
}
```

```
Class HtmlUnitDriver implements WebDriver
{
;;;;;;;;;;;
}
```

```
Class EventFiringWebDriver implements webDriver
{
;;;;;;;;;;
}
```

- remote webdriver is specially designed for remote level
- Html Unit Driver – If you want to perform headless browser testing without UI and browser to execute test cases.
- Using HtmlUnitdriver and Phantom JS driver we can perform headless browser testing.

```
{



}
```

To work with browser like chrome

```
Class chromeDriver extends RemoteWebDriver
{

}
```

For Other Browsers

```
Class FirefoxDriver extends RemoteWebDriver
{

}


Class internetExplorerDriver extends remoteWebDriver
{

}


Class safariDriver extends RemoteWebDriver
{

}


Class operaDriver extends RemoteWebDriver
{

}
```

Chrome Browser

- Chrome driver is a class which is child class of remote web driver.
- Chrome browser is specially designed to work with chrome.
- In chrome browser we have some constructors.

Default constructor

```
Public chromedriver( )
{
        //pickups chrome binary and launches chrome browser
}


Public chromeDriver(chrome options cos)
{
}
```

-To perform head less browser testing

```
Public chromedriver ( DesiredCapabilities dc)
{
}
```

-To set as Desired capabilities

Firefox Browser

```
Class FirefoxDriver extends RemotewebDriver
{
Public firefox driver( )
{
        //pickups firefox binary and launches firefox browser
}
;;;;;;;;
```

```
        }
```

Internet Explorer

```
        Class Internetexplorer extends RemotewebDriver
        {
        Public Internetexplorerdriver( )
        {
                //pickups IE binary and launches IE browser
        }
        ;;;;;;;;;
        }
```

Same for other browsers also

Edge browser works in only windows os.

A webpage contains webElements.

- Each webelement contains some attributes like name, id, class, etc
- webElement is an interface which is specially designed to find webelement

```
                interface webElement
                {
                Void clear( );
                Void sendkeys(char sequence……  cs);
                Void click( );
                boolean isDisplayed( );
                boolean isEnabled( );
                boolean isSelected( );
                String gettext( );
                String getTagName( );
                String getAtrribute(String attname);
                ;;;;;;;;;
                }
                Class RemoteWebElement implements webElement
                {
                }
```

Automate the following testcase:

1.Open the chrome Browser

2.Enter URL as www.gmail.com

3.Enter username as javaselenium@gmail.com

4.click on next button

5.close the browser

System:

System is a predefined class which is coming from java.lang package.

```
        Class system
        {
        Public static void setProperty(string key, String value);
        Public static void getProperty(string dir);
        Public static void exit(int status);
        ;;;;;;;;;;
        }
```

In this system class we have all methods static to access static through classname.

Standard Keys

webDriver.chrome.driver

webdriver.ie.driver

webdriver.gecko.driver (firefox)

<u>To Launch Browser</u>

```
Public class ChromeTest
{
Public static void main(string[ ] args)
{
System property("webdriver.chrome.driver", "Location");
```

1st approach // chromeDriver driver = new ChromeDriver( );

2nd approach //RemoteWebDriver driver = new ChromeDriver( );

3rd approach //Searchcontext sc = new ChromeDriver( );

4th approach //WebDriver driver = new Chrome Driver( );

5th Approach Using WebDriver Manager

```
driver.manage().windows().maximize();
driver.manage().deleteAllCookies();
driver.get(https://gmail.com);
Thread.sleep(5000);
driver.close();
driver.close();
```

To wait the browser for few seconds we have Thread class (predefined)

```
Class Thread
{
Public static void sleep(int ms);
Public static void sleep(int ms, int ms);
;;;;;;;;
}
```

To identify the elements, we use find elements

```
webElement username=driver.findelement(By.id(“identifierId”));
```

based on the attributes of web element we are going to identify the elements.

1.Identify the element

```
webElement username=driver.findelement(By.id(“identifierId”));
```

2.Perform the action

```
Username.clear();
Username.senkeys(“javaselenium@gmail.com”);
```

To click on next

1.Identify the element

```
Webelement nxt= driver.findelement(By.classname(“ ”));
```

2.Perform Action

```
nxt.click();
```

Locator Mechanism:

To identify the elements, we have locator mechanism using locator mechanism we can identify elements.

There are 8 types of locator mechanisms.

- Id
- Name
- Classname
- Xpath
- Css selector
- Tagname
- Linktext
- Partial link text

All 8-locator mechanism are implemented as a part of predefined class By.

-By is a class which is specially designed to work with locator mechanism.

All these 8 methods are public static

```
Public abstract class By

{

Public static By id(String id)

Public static By  name(String name)

Public static By  classname(String classname)

Public static By  xpath(String xpath)

Public static By  css Selector(String css)

Public static By   tagName(String tagname)

Public static By  linkText(String txt)

Public static By  partiallinkText(String txt)
```

We can access above methods by using classname

Classname.methodname

Id: -

In most of the cases id will be unique.

Whenever id contains numeric id name changes

Id=demo

Name: -

We can identify the element with name.

But there may be a chance of existing duplicate name.

Classname: -

We can identify element using classname also but here also there is a chance of existing duplicates.

Linktext: -

We can identify element using link text for that we have few guidelines.

Guidelines:

1.Element should start with <a> (anchor tag)

2.Element should have text

3.pass full text to linktext(-) method

```
webElement element = driver.findElement(By.linkText("Hardwork"));
        <a  id = demo>Hardwork</a>
```

Partial Link Text:

We can identify element using Partial link text for that we have few guidelines.

Guidelines:

1.Element should start with <a> (anchor tag)

2.Element should have text

3.pass full/partial text to partial linktext(-) method

Few examples for partial linktext

webElement element = driver.findElement(By.partiallinkText("Hardwork")); Valid

webElement element = driver.findElement(By.partiallinkText("ardwork")); valid

webElement element = driver.findElement(By.partiallinkText("Hard")); valid

webElement element = driver.findElement(By.partiallinkText("Hardork")); invalid

Sequence should be there properly

I18N:-

A Single application providing support for multiple language is called internationalisation

Language to language attributes are never going to change.

Whenever application using I18N don't identify the elements using linktext and partial link text.

Xpath:-

Xpath is a mechanism to identify the elements.

There are two kinds of xpath

1.Relative Xpath

2.Absolute Xpath

Relative xpath:

- Single attribute xpath

  Syntax :
  //tagname[@attname='attvalue']
  Or
  //*[@attname='attvalue']

Browser console: - we can validate

$x(String xpath)

- Multi attribute xpath

  Syntax:
  //input[@id='identifierId'][@name='identifier']

  Every multi attribute xpath behaves as and operator

--text()

Text function is to identify the elements

Syntax:

//tagname[text()='text']

//*[text()='text']

//input[text(-)='PRACTISE']

//input[text()='PRACTISE']

--contains

--ends-with

--starts-with

$\}$ **these are the 3 functions used to identify the elements.**

Contains: *It is a function which is applicable at two levels*

1.attribute level

2.text level

Syntax:

//tagname |*[contains(@attname,'attvalue')]

//tagname |*[contains(text(),'textvalue')]

Example: javawithselenium contains java

Javawithselenium contains selenium

Ends-With:

Syntax:

//tagname |*[ends-with(@attname,’attvalue’)]

//tagname |*[ends-with(text(),’textvalue’)]

Example: javawithselenium ends-with selenium

Javawithselenium ends-with um

Starts-with:

Syntax:

//tagname |*[starts-with(@attname,’attvalue’)]

//tagname |*[starts-with(text(),’textvalue’)]

Example: javawithselenium starts-with java

Javawithselenium starts-with jav

For every function sequence is mandatory.

- If the element does not have any attribute, we can identify by checking the nearest element which is having strongest attribute.
- If the elements are at same level those elements are called siblings.

```
<input  id = demo>  →  having 5 siblings 5 following 0 preceding
<div>
    <a id = job>
        <div>  ↲ child  2 siblings 1 preceding 1following
            <ifra ne>
        <a>
<iframe> →  have 5siblings 3 following 2 preceding
<div>  →  1 direct child 2 indirect child
        <input> directchild
          <div id = name> in directchild
               <a>  →  having 1 direct child
                   <input name = software>
<iframe>  →  0 child's
<iframe>.
```

| → These lines having 7 following and 6 preceding

Identify find the elements

1st <a> ADP </a>

<a> ADP</a>

<a> ADP</a>

<a> ADP</a>

//a[text()='ADP'] – identifying all but we need only 3

(//a[text()='ADP'])[3]

2nd <a> ADP </a>

<a id=java> ADP </a>

<a> ADP</a>

<a> ADP</a> **-----identify**

(//a[text()='ADP'])[4]

But if application is I18N text is going to changes from language to language.

Search for another attribute other than text but it does not having .

Checking the strongest near attribute

<a id=java>ADP</a> having strong attribute

//a[@id='java'] /following-sibling::[2]

CSS SELECTOR: -

It is also a locator mechanism to identify the elements.

In order to identify

Single attribute:

*[attname = 'attvalue']

tagname[attname='attvalue']

Multi attribute:

*[attname = 'attvalue'] [attname = 'attvalue']

tagname[attname='attvalue'] [attname='attvalue']

validation in browser console

$$(String css)

If you want to specify "or" operator in css selector use ","(coma)

Special operators:

, = or operator

][= and operator

$ = ends with

^ = stats with

** = contains*

+ = adjacent siblings

~ = preceding siblings

> = direct child

Space = indirect child

.(dot) = class name

# = id

- If you want to identify specially with id

  tagname #id

  #id

  Ex: Q) input[id='java']

  A) Input #java
  #java
- In same way we can identify classname
  tagname.classname
  .classname (.[dot] Represents classname)
- In css selector we need to cover space with(.)
  <input id=java class=selenium>
  Input.java.selenium

Example: //input[@id='java']/iframe (xpath)

Input[id='java']>iframe (css)

Input#java>iframe

#java>iframe

Ex:(1) write the below xpath in to css selector

Xpath= //input[@id='java']/iframe[1]

Css=input[id='java'>iframe:nth-of-type(1)

Ex(2): write the below xpath in to css

Xpath=//input[@id='java']/iframe[1]//div[@class='java']

css
- input[id='java']>iframe:nth-of-type(1) div[class='java']
- Input#java>iframe:nth-of-type(1)div.java
- #java>iframe: nth-of-type(1).java

- After / if we are writing any function, it will be function
- After / if we write any tagname it will be child.
- In css we don't have text mechanisms.

<u>$ ends with:-</u>

*|tagname [attname$='attvalue']

Ex:- <input id=java class=javaselenium>

Input[id$='java'] valid

Input[id$='va'] valid

Input[id$='jav'] invalid

^ Starts with:-

*|tagname[attname^='attvalue']

Input[class^= 'java selenium]

Input[class^= 'java se'] valid

Input[class^= '   selenium'] invalid

***contains:-**

|tagname[attname='attvalue']

Input[class*='java selenium']

Input[class*='selenium]

Input[class*='java']

XPATH TO CSS

Ex1: //input[@class= ‘java selenium]/div//iframe/a[@id= ‘java’]

Input[class= ‘java selenium’]>div iframe>a[id=’java’]

Input.java selenium>div iframe>a#’java’

.java selenium>div iframe>#java

EX2://input[@class=’java’]/preceeding-sibling::div/iframe[2]/a[3]

Input[class=’java’]+div>iframe:nth-of-type(2)>a:nth-of-type(3)

Input.java+div>iframe:nth-of-type(2)>a:nth-of-type(3)

<u>**WebElement**</u>

webElement is an interface

- Before checking an element first check whether it is Displaying or not by .isDisplayed() and then check element is enabled or not by .isEnabled()

Q) How do you validate the checkbox or radiobutton is selected or not?

A) To do that we have a method called .isSelected()

- If webElement is displayed it returns true else false
- If webElement is enable it returns true else false
- If webelement is selected it returns true else false

isSelected() is a method which is applicable for radio buttons and check boxes.

<u>**Dropdowns**</u>

Contains a list of values or group of values

1.single select dropdown

2.multi select dropdown

3.it is a drop down but it is not dropdown

Every dropdown must and should starts with **select** tagname.

Whenever the web element is starting with select tagname such kind of element are called web element.

- ➢ It seems like a dropdown in UI level but it does not start with select tagname.

In order to handle dropdowns, we have predefined class **select**.

The prerequisite to use select class is compulsory the dropdown should start with **select** tagname.

```
Class select
{
Private webElement element;
Public select(webelement element)
{
Stringname=element.getTagname();
If(tagname.equalIgnorecase("select")
{
//you are allowed to access select class
}
else
{
//it will throw exception
}
}
```

Different Methods: -

Public Boolean isMultiple()   //if it is multi select returns true else false

In order to select the values from dropdown.

Public void selectByindex(int Index)

Public void selectByValue(String Value)

Public void SelectByvisisbleText(String text)

To deselect (for only multi select dropdown)

Public void deselectByindex(int Index)

Public void deselectByvalue(String value)

Public void deselectByVisibleText(String text)

Public void deselectAll()

If we selected multiple option to know which first selected

Public webElement getFirstSelectedOption()

Public List<webElement> getAllselectedoptions()

Public List<webElement> getOptions() //return all in the form of list

equalsIgnoreCase (): -

it is a method which compare two strings if the strings are equal this method returns Boolean true or else false.

Actions: -

Actions is a class in that actions class we have three constructors. whenever we execute constructors we can perform web actions, keyboards, mouse operations etc...

```
Public  class Actions

{

Private webDriver driver;

Public Actions(webDriver driver)

{

;;;;;;;;;     //all webdriver keyboard mouse operations

}

Public  Actions (Keyboard board, Mouse mouse)

{

;;;;;;;;;;

}

Public Actions(Mouse mouse)

{

;;;;;;;;;

}

Public Actions click();

Public Actions click(webElement ele);

Public Actions doubleClick();

Public Actions doubleClick(webElement ele);


Public Actions sendkeys(Keys to send keys);   //to enter some keys

Public Actions sendkeys(webElement ele,Keys to send keys);
Public Actions contextClick ();  //right click
Public Actions contextClick(webElement ele);

Public Actions moveToElement(webElement ele);  //mouse hover actions
Public Actions movetoElement(webElement ele, int x offset, int y offset);

Public Actions release();
Public Actions ClickAndHold();
Public Actions keysUp(Keys keys);
Public Actions keysUp(webelement ele, Keys keys);
Public Actions  keysDown(Keys keys);
Public Actions keysDown(webElement ele, Keys keys);

Public Actions pause(longtime);
Public Actions release(webElement element);

Public Action build();     }  Except these two methods all methods have
Public Action perform();   }          return type actions
}
```

Action:

Action is an interface which is specially designed to perform Action

Different ways to enter URL:

(1) driver.get("URL");

(2) Navigation navigation = driver.navigate();
Navigation.to("URL");

(3) URL url = new URL("url");
navigation.to(url);

(4) Using JavaScript executor

Different ways to perform click options (Actions)

- webElement nxt = driver.finElement(By.id("identifierNext"));
//nxt.click();
- actions.Click(nxt).build().perform();
- actions.doubleClick(nxt).build().perform();
- actions.sendKeys(nxt,keys.Enter).build().perform();
- Using javaScript executor also

Enum Keys: -

Keys is an Enum class. In this we all have all keyboard their as part of Enum class.

By default, it is public static final

We can access through only classname

Keys.F1/F2/ENTER/etc...

Public enum keys

{

F1, //public static final

F2,

F3,

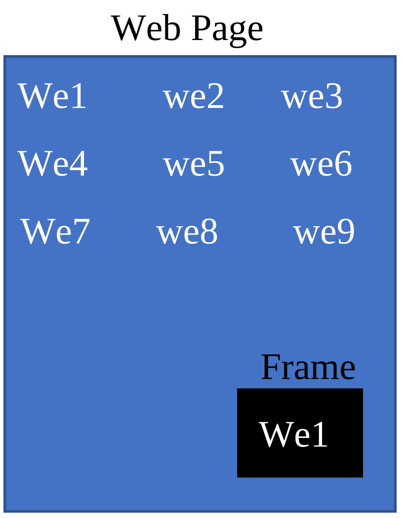
F4,

```
;;;;;
ENTER
NUMPAD
;;;;;;;;;;
}
```

Mouse Hover actions: -

1.using Actions class

2.using JavaScript executor



- whenever elements are their part of webpage, we can perform operation on these webpages directly.

- But the elements which is there in the part of frame we cannot identify the elements directly.

- In that case first we need to identify the element and then on that we should perform action generally.

- But whenever element there as part of frame Directly we cannot identify elements.

- First, we need to identify the frame once you identified the frame elements then you need to identify the web element.

- so that we use switch to identify elements.

How can we switch to the frame?

In order to switch to frame we have 3 ways.

frame(int index)

frame(String frame Id/name)

frame(webElement element)

➢ **To find total number of frames**

driver.findElements(By.tagname("iframe"));

List<webElements> frames = driver.findElements(By.tagname("iframe"));

System.out.println("Total no of frame"+frames.size());

[size() is a method to find the total number of elements in the list]

Switching:

Switch to frame

driver.switchTo().frame(0);

switch to main window (frame to main window)

driver.switchTo().defaultcontent();

- If there are two windows by default selenium will perform action on windows1 to shift the focus of window1 to wndow2 we use switch mechanism.

They implemented as a part of predefined interface called **Target Locator**

Target Locator is an interface which is specially designed to implement switching mechanism.

In this target locator interface, we have n number of methods.

```
Interface TargetLocator
{
webDriver frame(int index);                       ┐
webDriver frame(String frame Id or name);         ├ switch to frame
webDriver frame(webElement element);              ┘
webDriver window(String name);   //Switch to window
WebDriver parentFrame();         // Switch to parent Frame
Alert alert();
webElement active Element();
webDriver default content();     //Switch to Frame to main window
}
```

- Remote TargetLocator is implementation class of TargetLocator.

```
Class RemoteTargetlocator implements TargetLocator
{
;;;;;;;;;;;
}
```

Generally, we create object to access methods but we are not required to create object.

In webDriver Interface

- We have method called switchTo method to relocate the Targetlocator.

```
TargetLocator switchTo();
//driver/switchTo.defaultcontent();
TargetLocator targetLocator = driver.switchTo();
targetLocator.defaultContent();
```

<u>**Synchronisation:-**</u>

It is a process of making our tool and our application to wait each other to get appropriate results is called synchronisation.

By default, selenium wait time is 30 milli seconds

Different categories to implement synchronisation

1.implicit wait

2.explicit wait

3.fluent wait (polling based)

4.sleep mechanism

5.pageload timeout

6.set script timeout

Implicit wait:-

Once the browser is open initialize the implicitly wait.

driver.manage().timeouts.implicitlywait(45, Timeout.seconds);

$$\begin{tikzcd} \text{Timeout.seconds} \arrow[d] \\ \text{enums class} \end{tikzcd}$$

applicable for each and every element

Assume that the element is found in 5 seconds hen the remaining 40 sec going to skipped out.

Pageload timeOut:-

When click operation takes some time it takes sometime to load called pageload.

driver.manage().timeouts().pageLoadtimeout(2,Timeunit.MINUTES);

options: *options is an interface specially designed for cookies management system.*

In this options interface we have one more method timeOuts().

To get timeouts object

```
Options options   – driver,manage();
Timeouts timeouts = options.timeouts();
timeouts.implicitlywait(40,TimeUnit.SECONDS);
```

instead of writing three line we write one line(implicitly wait)

timeouts: in these timeouts interface we have three methods

```
timeouts.implicitlywait()
timeouts.pageloadtimeout()
timeouts.Setscripttimeout()
```

timeouts is an interface which is specially designed for synchronization.

Sleep Mechanism: In order to implement the sleep mechanism, we have a predefined class Thread.

```
Class Thread
{
Public static void sleep(int ms)
Public static void sleep(int ms, int ns)
;;;;;;;
}
```

As it is a static, we can access through classname

```
    Thread.sleep(5000);
```

Sleep is a method which is going to pause the application.

Explicitly wait: -

Whenever you want to handle the Ajax kind of dynamically called then we go for explicitly wait.

In order to implements the explicitly wait.

There are two classes.

```
    Class webDriverWait
    {
    Public webDriverWait(int ms second, webDriver driver)
    {
    ;;;;;;;;
    }
    Public void until(-);
    }


    Class Expected condition
    {
    Public static void visibilityOf(-)
    ;;;;;;;;;
    }
```

Explicit wait is applicable for only one element.

Object creation

```
webDriverWait  wait = new webDriverWait(driver,45);
wait.until(Expected conditions.visibilityOf());
webElement nxt = driver.findElement(By.id(“identifiernxt”));
nxt.click();
```

If it’s not visible in 45 seconds exception occur saying that invisibility of exception

In Expected condition all methods are static.

In xpath (//*) --- we get total no of elements

In css (*) --- we get total no of elements

Q) Find the total number of words Adp words count in Adp.com

```
Public class Adpcount
{
Public static void main(String[] args)
{
System.setProperty(“webDriver.chrome.driver”, “location of driver”);
webDriver driver = new Chromedriver();
driver.manage().window().maximize();
driver.manage().timeouts().implicitlywait(45,TimeUnit: SECONDS);
driver.manage().deleteAllCookies();
driver.get(“www.adp.com”);
int adpcount=0;
List<webElements> listelements = driver.findElements(By.xpath(//*));
for(webElement webelement: Listelements)
{
String txt = webElements.getText();
If(txt contains(“ADP”) ||txt.contains(“adp”))
{
adpcount++
}
}
System.out.println(“Total”+adpcount);
}
}
```

By default selenium is going to focus single window. To get current window name we have method called

```
String getWindowHandle ();
```

To get total no. of window.

```
Set<String> getwindowHandles();
```

TestNg

Till now we are running our testcases in java application.

In Realtime there is no concept of main method

In Realtime we are going to use TestNg only.

- Whenever you run testcases in TestNg we have n number of advantages.
- By running testcases through TestNg it is going to generate reports.
- Using TestNg we can execute the group of testcases
- Using TestNg we can enable testcases and disable
- Using TestNg we can re run the testcases
- Using TestNg we can check the performance of test cases
- Using TestNg we can execute single testcase multiple times.

TestNg having bunch(group) of annotations.

- This annotation will tell you the flow of execution in which order testcases should execute.
- Using TestNg we can re-run the failed testcases

TestNg annotations

@BeforeSuite

@BeforeTest

@BeforeClass

@BeforeMethod

@Test

@AfterMethod

@AfterClass

@AfterTest

@AfterSuite

@DataProvider

@Optional

@Factory

@Parameters

- In a single testcases if you have more than one testcase or @test it will execute by Alphabetical order.
- To give **priority** one testcase which case should be executed first for that we have an attribute **(priority=0)**
- If we don’t want to executed test case method then
  (enabled=false) not going to execute
  (enabled=True) going to execute
- If you want to execute the test case/method multiple times then we use **(invocationcount=5)** here 5 means no of times to be executed as per requirement.
  5times before method and 5 times after method
- Invocation count is used to check performance.
  (invocationcount=5, invocationTimeout=25000)
  Should execute 5 times in 25 seconds.

To run Multiple test cases

Using testNg.xml file we can run all testcases at a time.

SuiteMechanism

Generally, we save this file as testng.xml file. Using this testng.xml /suitexml we can run multiple testcases at a time.

Select the package > testNg > convert to testNg.

A xml file will be generated.

In a single class if we have multiple methods their execution happens based on alphabetical order.

Create xml file Manually

Select the project > right Click > new > other > look for xml > click next > give filename.

If in a class we have 20 testcases assume we want to execute only 5 testcases.

- We cannot write (enabled=false) at method because whenever you do modifications in java file, we need to do compilation. don’t change any modification

- We should add
```
<class name= -----------   class>
   <methods>
       <include name= “methodname”>
       </include>
    </methods>
</class>
```

To exclude testcases instead of include statement use the below one.

```
<exclude name= “method name”>

</exclude>
```

➡ We can execute our testcases at three levels

- Suite level with package
- Suite level with classes
- Suite level with package and methods

➡How do you re run the failed test cases?

Under the test-out folder file will be created testng-failed.xml (of failed test cases). under Suite also we can see

1.using TestNg class

2. using IRetryAnalyzer Listener

➡Assume that we have suite of test cases

Regression suite ---------200 test cases

Sanity Suite -------------- 40 test cases

End to end suite----------250 test cases

This we can implement with help of grouping mechanism in xml file

```
<suite name=”Suite”>

<groups>

<run>

    <include name= “e2e”> </include>

</run>
```

```
</groups>
```

To get that at method we should specify

(group = {“e2e”, “rt”, “st”}) (whatever you required you should indicate)

➡ Assume that depends on groups also we can specify

Ex: @Test(groups={“e2e”}, depends on groups = {“e2e”};

Is depending on groups which group this test method is depending first that should be executed.

@Parameters

```
Public class ParametersDemo
{
@Test
Public void hello(String un, String pwd)
{
System.out.println(un+ “   “+pwd);
}
```

If you execute this directly it will show exception that method hello requires 2 parameters but 0 were supplied in the @Test annotation.

- ➢ How do I supply the parameters to the method for that we have annotation
  @Parameters ({“ , “})
  in braces supply parameters.
- ➢ From xml file also we should supply parameters
  <suite name name= “un”>
  <parameter name = “un” value= “Ramesh”></parameter>
  <parameter name = “pwd” value= “Hardwork”> </parameter>

If you are not supplying any parameters then include default values.

Public void hello(@optional(“ “)String un,(“ ”)String pwd)

If you are not supplying any parameter in xml file then default values should mention.

@DataProvider

I want to execute same test case n no. of times with different set of data then we go for @DataProvider.

- Using data provider, we can execute single test case with different sets of data.

Syntax:

```
Public Object[ ] [ ] dataprovidername( )
{
Object[ ] [ ]  varname = new Object [row] [col];
;;;;;;;;;;
return varname;
}
```

Here rows represent how many times we want to execute the test case

Columns represents number of inputs.

```
Public  class  dataProviderDemo
{
@Test(dataprovider = “helloData”)
Public void hello(string un, Sting pwd)
{
System.out.println(un+ “ ”+pwd);
}
@dataprovider
Public object[][] helloData()
{
Object[][]  obj = new Object[2][2];
Obj[0][0]  = “java”;                    1st iteration
Obj[0][1]  = “selenium” ;
Obj[1][0]  = “hardwork” ;   ⎫  2nd iteration
Obj[1][1]  = “job”;         ⎭
return obj;
}
```

Here is the list of annotations that TestNG supports

-@BeforeSuite: The annotated method will be run only once before all tests in this suite have run.

-@AfterSuite: The annotated method will be run only once after all tests in this suite have run.

-@BeforeClass: The annotated method will be run only once before the first test method in the current class is invoked.

-@AfterClass: The annotated method will be run only once after all the test methods in the current class have run.

-@BeforeTest: The annotated method will be run before any test method belonging to the classes inside the tag is run.

-@AfterTest: The annotated method will be run after all the test methods belonging to the classes inside the tag have run.

-@BeforeGroups: The list of groups that this configuration method will run before. This method is guaranteed to run shortly before the first test method that belongs to any of these groups is invoked.

-@AfterGroups: The list of groups that this configuration method will run after. This method is guaranteed to run shortly after the last test method that belongs to any of these groups is invoked.

-@BeforeMethod: The annotated method will be run before each test method.

-@AfterMethod: The annotated method will be run after each test method.

-@DataProvider: Marks a method as supplying data for a test method. The annotated method must return an Object[ ][ ], where each Object[ ] can be assigned the parameter list of the test method. The @Test method that wants to receive data from this DataProvider needs to use a dataProvider name equals to the name of this annotation.

-@Factory: Marks a method as a factory that returns objects that will be used by TestNG as Test classes. The method must return Object[ ].

-@Listeners: Defines listeners on a test class.

-@Parameters: Describes how to pass parameters to a @Test method.

-@Test: Marks a class or a method as a part of the test

Top 7 Selenium Commands with Details

---

1) get() Methods

• driver.get("https://google.com");

• • driver.getClass();

• • driver.getCurrentUrl();

• • driver.getPageSource();

• • driver.getTitle();

• • driver.getText();

• driver.findElement(By.id("findID")). getAttribute("value");

• • driver.getWindowHandle();

2) Locating links by linkText() and partialLinkText()

• driver.findElement(By.linkText(“Google”)).click();

• driver.findElement(By.partialLinkText(“abode”)).click();

3) Selecting multiple items in a drop dropdown

• // select the multiple values from a dropdown

• Select selectByValue = new Select(driver.findElement(By.id("SelectID_One")));

• selectByValue.selectByValue("greenvalue"); - By Value

• selectByValue.selectByVisibleText("Red"); - By Visible Text

• selectByValue.selectByIndex(2); - By Index

4) Submitting a form

// submit the form

• driver.findElement(By.*id*("submit")).submit();

5) Handling iframes

• Select iframe by id driver.switchTo().frame(“ID of the frame“);

• • Locating iframe using tagName

• driver.switchTo().frame(driver.findElements(By.tagName(“iframe”).get(0));

• • Locating iframe using the index:

• a) frame(index) driver.switchTo().frame(0);

• b) frame(Name of Frame) driver.switchTo().frame(“name of the frame”);

• c) frame(WebElement element) Select Parent Window driver.switchTo().defaultContent();

6) close() and quit() methods

• driver.close(); - closes only a single window that is being accessed by the WebDriver instance currently

• • driver.quit();- closes all the windows that were opened by the WebDriver instance

7) Exception Handling

• WebElement saveButton = driver.findElement(By.id("Save"));

```
try{
 if(saveButton.isDisplayed()){
 saveButton.click();
 }
 }
catch(NoSuchElementException e)
{
e.printStackTrace();
```

}

- you can use Javascript Executer to input text into a text box without using sendKeys()

  method: // Initialize JS object JavascriptExecutor JS = (JavascriptExecutor)webdriver; // Enter username JS.executeScript("document.getElementById('User').value='Abha_Rathour'"); // Enter password JS.executeScript("document.getElementById('Password').value='password123'");
- select a value from a dropdown in Selenium WebDriver.

  Select select = new Select(driver.findElement(By.id("abcd"))); select.selectByVisibleText()/deselectByVisibleText(); - selects/deselects an option by its displayed text
  select.selectByValue()/deselectByValue(); - selects/deselects an option by the value of its "value" attribute
  select.selectByIndex()/deselectByIndex(); - selects/deselects an option by its index select.isMultiple(); - returns TRUE if the drop-down element allows multiple selection at a time; FALSE if otherwise
  select.deselectAll(); - deselects all previously selected options
- What is the difference between single and double slash in Xpath?

  A double slash " // " means any descendant node of the current node in the HTML tree which matches the locator. // ->Selects nodes in the document from the current node that match the selection no matter where they are A single slash " / " means a node which is a direct child of the current. / -> Selects from the root node
- How do you find broken links in Selenium WebDriver?

  Collect all the links in the web page based on tag.
  Send HTTP request for the link and read HTTP response code.
  Find out whether the link is valid or broken based on HTTP response code.
  Repeat this for all the links captured.

GIT

In Real Time we move the code into central interface called GIT.

GIT vs GITHUB

- GitHub is a place where we are going to host our projects.
- By using GIT, we can communicate with GitHub through commands.
- Through communicate GitHub we have a channel called GIT.

Git is also called as project management tool.

In general, we use Git through

- Commands
- GitHub Desktop

By using commands, we can push code from local to remote level.

- Pull code from remote to local machine
- We can maintain all our data using commands.

Install the git in local machine.

Goto google > search download Git > open the site and download the git.

And install the file.

- To check GIT is there in system we sue command git in cmd
- To know the version open cmd and enter as git –version

Repository:

Repository is a place where we are going to maintain our projects, so in that repository only we have the projects we can push/move the code to repository.

Generally, in eclipse we are going to write our code in project.

In GitHub we are going to maintain our projects in one area called Repository.

GitHub

Create a GitHub account in GitHub website which acts as a remote. After creating account in GitHub successfully remember the credentials for future use.

HOW TO CREATE THE REPOSITORY

Go to GitHub location click new enter the repository name and apply.

Then you can see a repository URL

WHAT IS REPOSITORY URL

This URL contains our github.com URL along with username or owner name slash our repository name . (dot) git

Using https any one can clone, pull, access.

Whereas SSH we should maintain or should purchase some keys and licences have to provide some access code. It's not open source.

<u>PROCEDURE</u>

CONFIGURE PROJECT TO GIT

- git config --global user.name "GitHub user id"
- git config --global user.email "email"

TO INITIALISE GIT   (at project level only one time)

- git init

TO CHECK STATUS

- git status

ONCE YOU INTIALIZE YOU HAVE TO DO STAGING

GIT STAGING: - WE CAN ADD/ REMOVE FILE OR FILES

- git add<filename>   //for single file
- git add<filename1><filename2>   //for multiple files
- git add * (or) git add .   //to add all files at a time

IF REQUIRED CHECK STATUS AGAIN TO SEE WEATHER FILES ADDED OR NOT

- git status

NEXT COMMIT THE CODE

- git commit -m “message”

PUSH THE CODE

There are two types

1.1st time pushing

2. 2nd time pushing

1ST TIME PUSH:-

- git remote add origin <repoURL> or <branchname>
- git push <repoURL>

2ND TIME PUSH:-

- git push <repoURL>

GIT PROVIDE A COMMAND CALLED CLONE

- git clone <repoURL> or <branchname>

whenever you execute clone command whatever the data available in repository level those projects will come/ copied to local machine

- git pull <repoURL>

pull command to pull the code from remote to local machine.

DIFFERENCE BETWEEN CLONE AND PULL

Whenever you execute clone command you get entire project in to local machine.

Whenever you use pull command you get only updated code.

In Realtime before pushing the code pull the code to get updated code then you can change the code.

Whenever you pull code, you get some conflicts

Git conflicts head tail alphanumeric.

Explain Git workflow.

Step 1: Set up a GitHub Organization. ...

Step 2: Fork Organization Repository to Your Personal GitHub. ...

Step 3: Clone the Repository to Your Local Machine. ...

Step 4: Create a Branch for your Working Files. ...

Step 5: Set Remote Repository to the GitHub Organization. ...

Step 6: Get Coding!

Step 7: Pull the Most Recent Files from the Organization Repo

Step 8: Merge the Master Branch into the Feature Branch

Step 9: Push Your Code to your GitHub Repo

Step 10: Make a Pull Request to the Organization Repo

MAVEN

Maven is a build automation tool used primarily for Java projects. Maven can also be used to build and manage projects written in C#, Ruby, Scala, and other languages. The Maven project is hosted by the Apache Software Foundation, where it was formerly part of the Jakarta Project

Maven is a group of commands. BY using maven commands our job is getting easy.

Download the maven by the following link:

https://maven.apache.org/download.cgi

https://*dlcdn*.apache.org/maven/maven-3/3.8.5/binaries/apache-maven-3.8.5-bin.zip

Apache Maven

What is it?

Maven is a software project management and comprehension tool. Based on the concept of a Project Object Model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

Documentation

The most up-to-date documentation can be found at https://maven.apache.org/.

Release Notes

The full list of changes can be found at https://maven.apache.org/docs/history.html.

System Requirements

JDK:

1.7 or above (this is to execute Maven - it still allows you to build against 1.3 and prior JDK's).

Memory:

No minimum requirement.

Disk:

Approximately 10MB is required for the Maven installation itself. In addition to that, additional disk space will be used for your local Maven repository. The size of your local repository will vary depending on usage but expect at least 500MB.

Operating System:

Windows:

Windows 2000 or above.

Unix based systems (Linux, Solaris and Mac OS X) and others:

No minimum requirement.

Installing Maven

1) Unpack the archive where you would like to store the binaries, e.g.:

Unix-based operating systems (Linux, Solaris and Mac OS X)

```
tar zxvf apache-maven-3.x.y.tar.gz
```

Windows

```
unzip apache-maven-3.x.y.zip
```

2) A directory called "apache-maven-3.x.y" will be created.

3) Add the bin directory to your PATH, e.g.:

Unix-based operating systems (Linux, Solaris and Mac OS X)

```
export PATH=/usr/local/apache-maven-3.x.y/bin:$PATH
```

Windows

```
set PATH="c:\program files\apache-maven-3.x.y\bin";%PATH%
```

4) Make sure JAVA_HOME is set to the location of your JDK

5) Run "mvn --version" to verify that it is correctly installed.

For complete documentation, see https://maven.apache.org/download.html#Installation

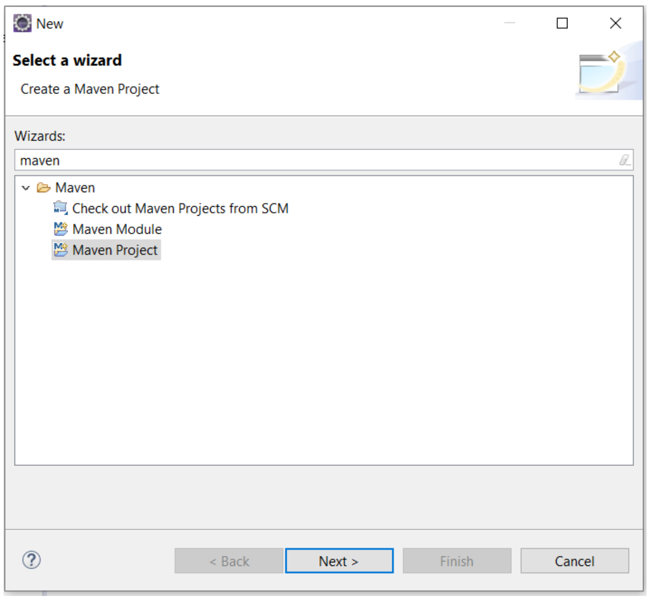
Maven URLS

Home Page:	https://maven.apache.org/
Downloads:	https://maven.apache.org/download.html
Release Notes:	https://maven.apache.org/docs/history.html

Mailing Lists:	https://maven.apache.org/mailing-lists.html
Source Code:	https://gitbox.apache.org/repos/asf/maven.git
Issue Tracking:	https://issues.apache.org/jira/browse/MNG
Wiki:	https://cwiki.apache.org/confluence/display/MAVEN/
Available Plugins:	https://maven.apache.org/plugins/

After successfully setting up the environmental variables

Go to eclipse > Right click on project explorer > select new > select other > in the input text field type Maven > then a wizard will be shown as below


New

Select a wizard

Create a Maven Project

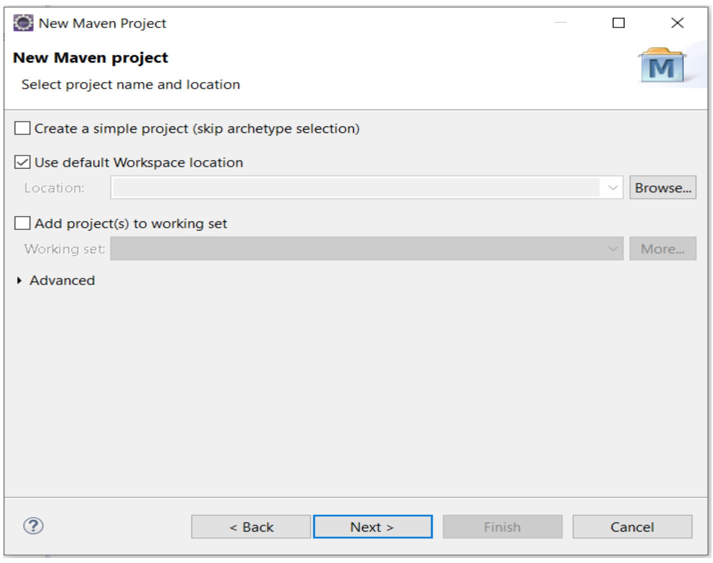
Wizards:

maven

Maven
Check out Maven Projects from SCM
Maven Module
Maven Project

Select the maven project and click on next.

By clicking on next button another wizard will be shown as below.


New Maven Project

New Maven project

Select project name and location

☐ Create a simple project (skip archetype selection)

☑ Use default Workspace location

Location:

Browse...

☐ Add project(s) to working set

Working set:

More...

Advanced

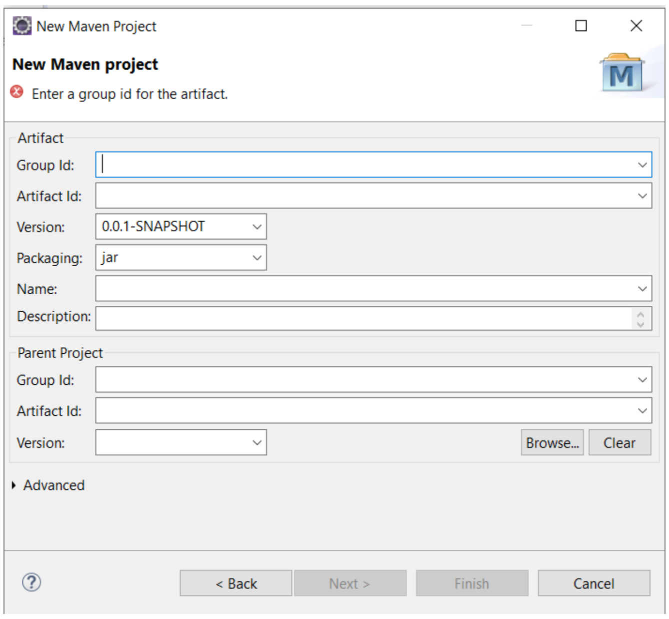
< Back | Next > | Finish | Cancel

There in upper image you can see a check box Create a simple project (skip archetype selection)

By clicking on checkbox yes, a simple project will be created. Skipping the archetype

Archetype indicates what type of project you want.

By clicking the checkbox and clicking on next button another wizard will be opened as shown.


New Maven Project

New Maven project

Enter a group id for the artifact.

Artifact

Group Id:	
Artifact Id:	
Version:	0.0.1-SNAPSHOT
Packaging:	jar
Name:	
Description:	

Parent Project

Group Id:	
Artifact Id:	
Version:	

Browse... | Clear

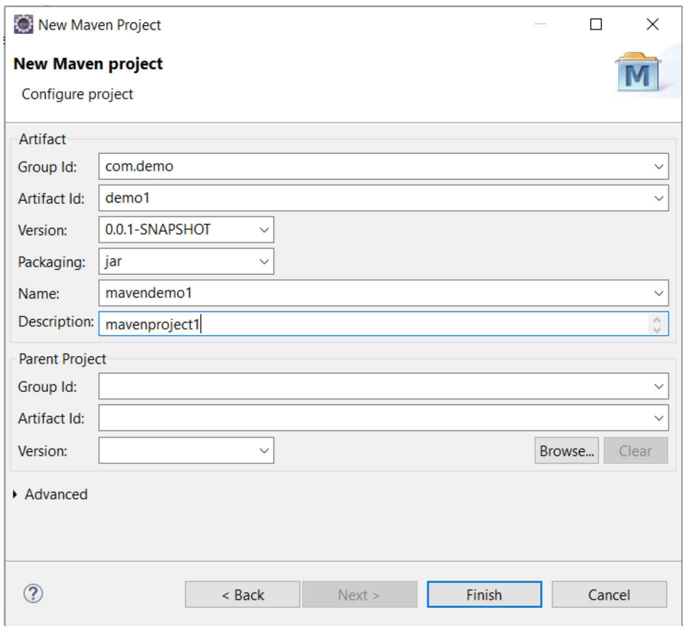
Advanced

< Back | Next > | Finish | Cancel

Here   group Id means package name

Artifact Id means project name

Enter the details as shown



Artifact
Group Id: com.demo
Artifact Id: demo1
Version: 0.0.1-SNAPSHOT
Packaging: jar
Name: mavendemo1
Description: mavenproject1

Parent Project
Group Id:
Artifact Id:
Version:

Click Finish.

In the same way u can create the maven project without checking the checkbox of archetype

We can convert normal java project in to maven project. steps to convert java project to maven project :

>rightclick on java project
>select configure option.
>Click on maven project
>click on finish.
>refresh the project.

In maven pom.xml file indicates Project Object Model.

By default, pom.xml files have dependencies. dependencies mean jar file

MAVEN contains mainly two repositories

1.Local Repository

2.Global Repository

Local repository means the repository which will be having in your local machine.

Global Repository means you need to download the jar files from the following link: https://mvnrepository.com/

By clicking on above link a site will be opened there in search bar you need to search the selenium and testng, extent reports,freemarker,apache poi api jar files and should be added to pom.xml file and by saving that file all jar files will get added.

Then you need to add maven compiler plugin and maven surefire plugin

By using maven compiler plugin, we can compile the source files.

By using maven surefire plugin, it executes the class files.

Jenkins

Jenkins is an open-source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. It is a server-based system that runs in servlet containers such as Apache Tomcat.

Jenkins is an Open-source continuous integration tool. We can download we can see the code and we can modify the code.

Generally, it is written in java programming language designed to test and report an isolated changes in larger code base in real time. It enables developers to find and solve the defects in a code base rapidly and automate the testing.

- Continuous Integration means it's a process in which all development work will integrated at a predefined time and resulting to the work it automatically tested at build.
- The Basic functionality of Jenkins is to execute a predefined the list of steps base at certain trigger of condition.

Jenkins can be started by a command line or we can run on a web application server as well.

To work on Jenkins, we need to download Jenkins from google or by clicking on below link we can download

For windows

https://www.jenkins.io/download/thank-you-downloading-windows-installer-stable

For mac

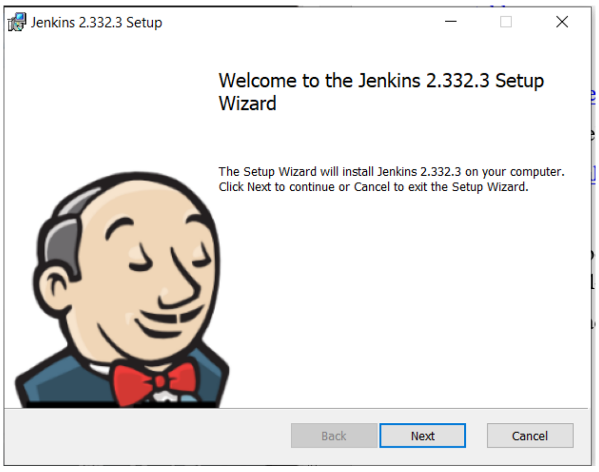
https://www.jenkins.io/download/lts/macos

For ubuntu/Debian

https://pkg.jenkins.io/debian-stable/

By clicking above link as per your operating system you can download. Jenkins.msi file will be downloaded.

After downloading Jenkins.msi file double click on the application a wizard will be shown.

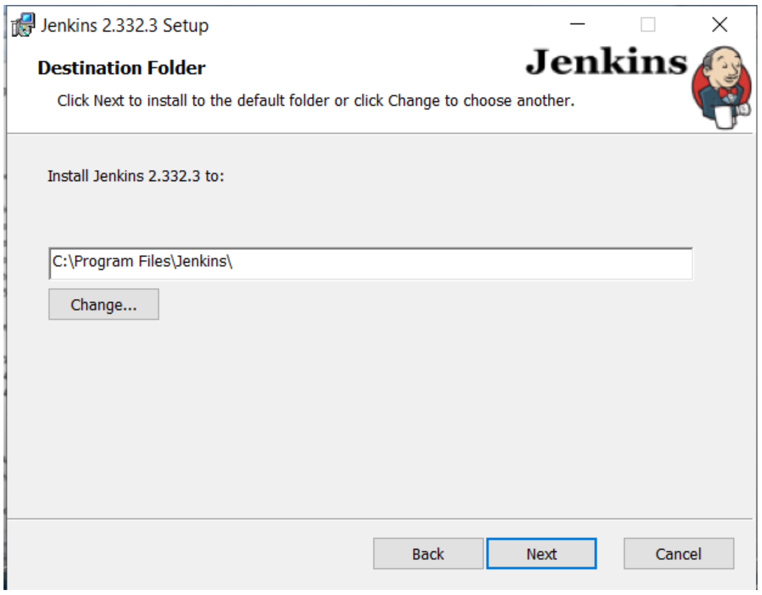


Jenkins 2.332.3 Setup

Welcome to the Jenkins 2.332.3 Setup Wizard

The Setup Wizard will install Jenkins 2.332.3 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Click next


Jenkins 2.332.3 Setup

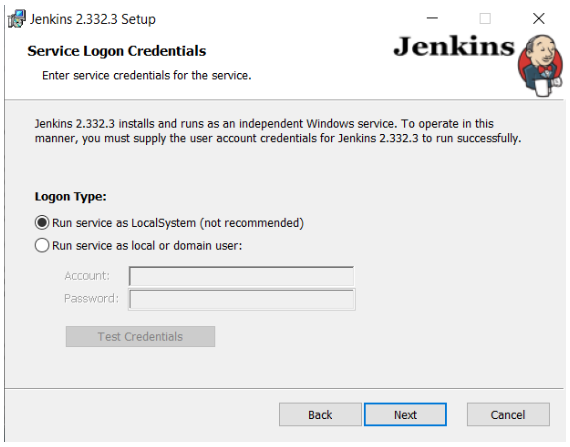
Destination Folder

Click Next to install to the default folder or click Change to choose another.

Install Jenkins 2.332.3 to:

C:\Program Files\Jenkins\

Click on Next


Jenkins 2.332.3 Setup

Service Logon Credentials

Enter service credentials for the service.

Jenkins 2.332.3 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.332.3 to run successfully.

Logon Type:

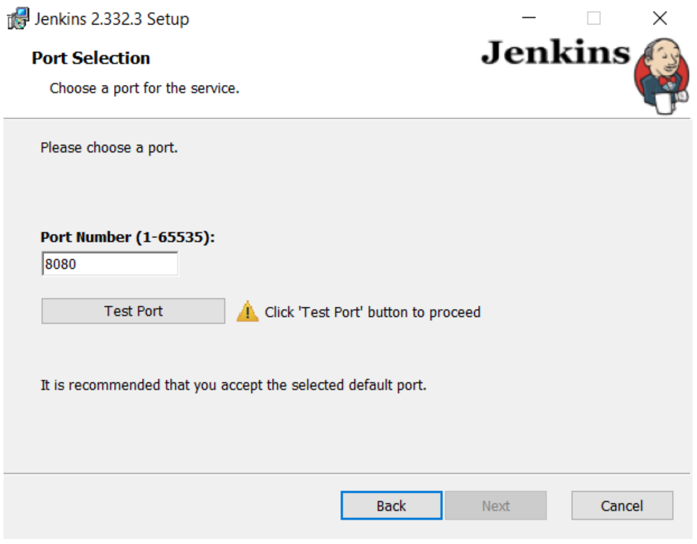
[✓ Run service as LocalSystem (not recommended)]

Run service as local or domain user:

Account:

Password:

On the above wizard select Run Service as LocalSystem and click on next


Jenkins 2.332.3 Setup

Port Selection

Choose a port for the service.

Please choose a port.

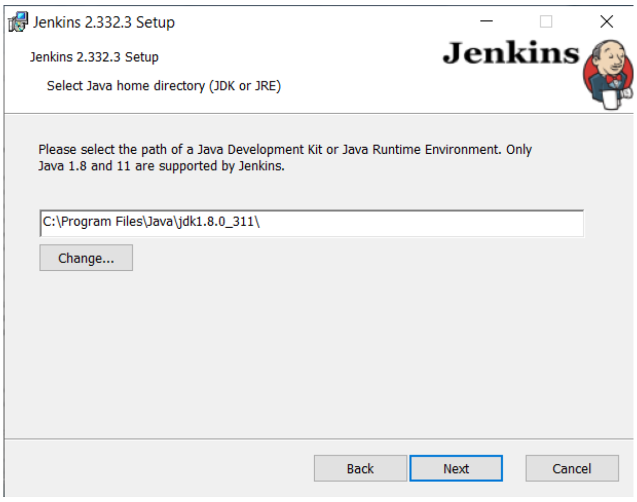
Port Number (1-65535):

8080

Click 'Test Port' button to proceed

It is recommended that you accept the selected default port.

On above wizard click on Test Port and click on Next


Jenkins 2.332.3 Setup

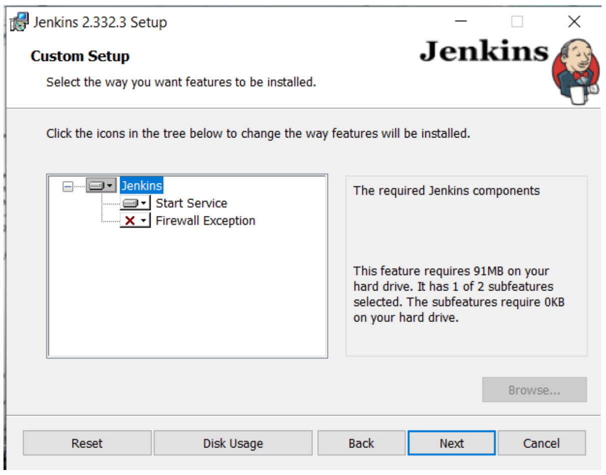
Jenkins 2.332.3 Setup

Select Java home directory (JDK or JRE)

Please select the path of a Java Development Kit or Java Runtime Environment. Only Java 1.8 and 11 are supported by Jenkins.

C:\Program Files\Java\jdk1.8.0_311\

Click Next


Jenkins 2.332.3 Setup

Custom Setup

Select the way you want features to be installed.

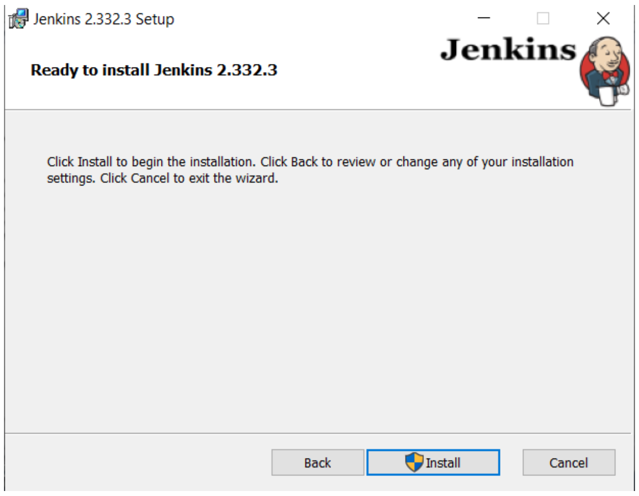
Click the icons in the tree below to change the way features will be installed.

- Jenkins
  - Start Service
  - Firewall Exception

The required Jenkins components

This feature requires 91MB on your hard drive. It has 1 of 2 subfeatures selected. The subfeatures require 0KB on your hard drive.

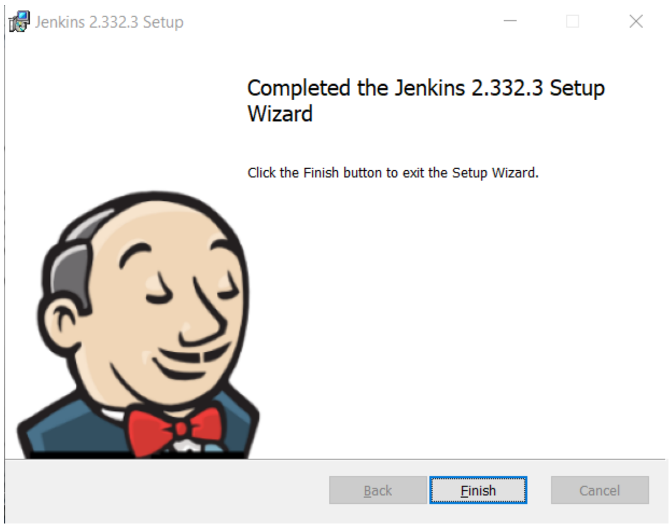
Click next


Jenkins 2.332.3 Setup

Ready to install Jenkins 2.332.3

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

Click on install option Jenkins will be installed a wizard shown like this



Jenkins 2.332.3 Setup

Completed the Jenkins 2.332.3 Setup Wizard

Click the Finish button to exit the Setup Wizard.

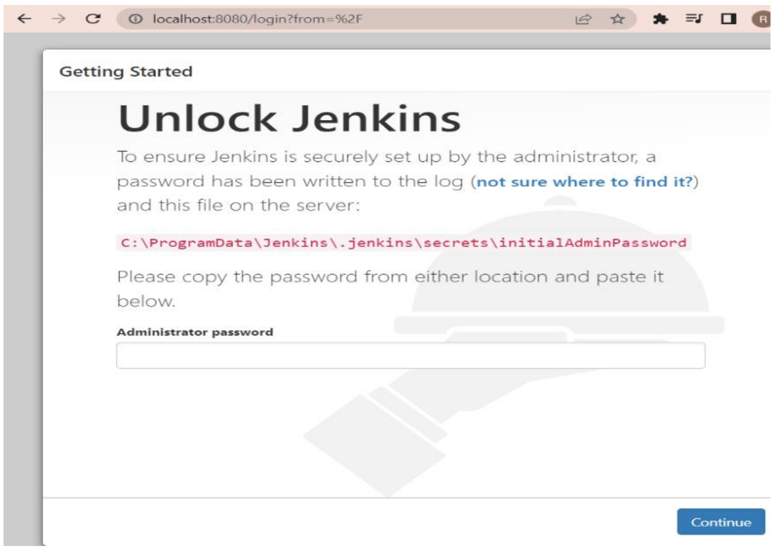
Click on FINISH

After clicking on finish to check weather Jenkins is installed or not go the following path in your system

C:\Program Files\Jenkins

Now you need to set the Jenkins path in environmental variables.

After setting the environmental variables now u need to stat Jenkins >Go to Google > Enter as it is in search bar as **localhost:8080** >Click on enter a window will be open as unlock Jenkins or use the link http://localhost:8080/login?from=%2F



localhost:8080/login?from=%2F

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:

```
C:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword
```

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

Administrator password

Unlock Jenkins

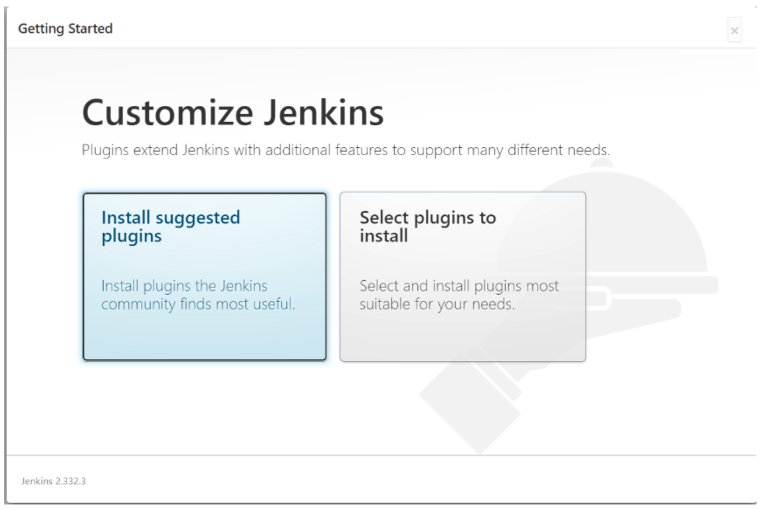
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`C:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword`

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

After copying the above file location and paste it in file manager path a list of files will be shown there you can see initial Admin password file open the file with notepad .Copy the text/password and enter in the localhost site. Click on Continue.

After entering password and clicking continue a page will be opened



Getting Started

Customize Jenkins

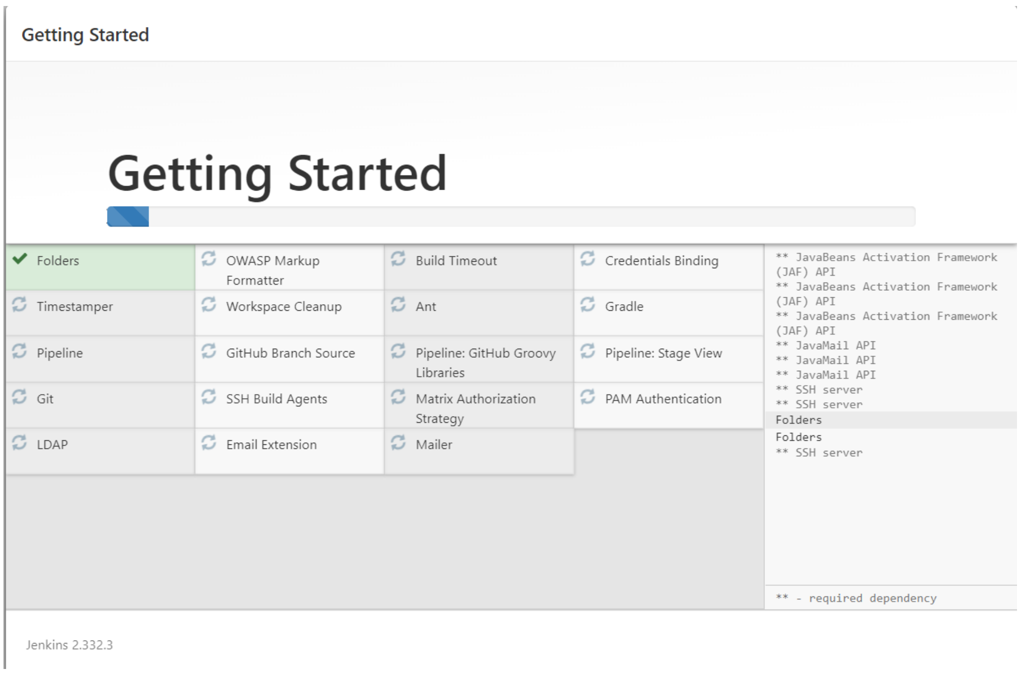
Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins
Install plugins the Jenkins community finds most useful.

Select plugins to install
Select and install plugins most suitable for your needs.

Jenkins 2.332.3

Now click on Install suggested Plugins


Getting Started

Getting Started

✓ Folders	OWASP Markup Formatter	Build Timeout	Credentials Binding
Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
LDAP	Email Extension	Mailer	

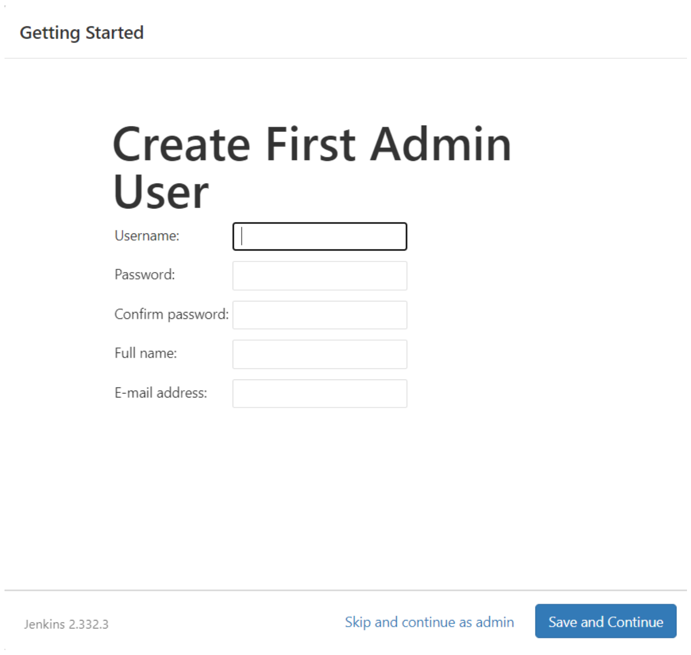
```
** JavaBeans Activation Framework
(JAF) API
** JavaBeans Activation Framework
(JAF) API
** JavaBeans Activation Framework
(JAF) API
** JavaMail API
** JavaMail API
** JavaMail API
** SSH server
** SSH server
Folders
Folders
** SSH server

** - required dependency
```

Jenkins 2.332.3

It will take some time to install all plugins.

After installing all suggested plugins next it will be re directed to account creation page

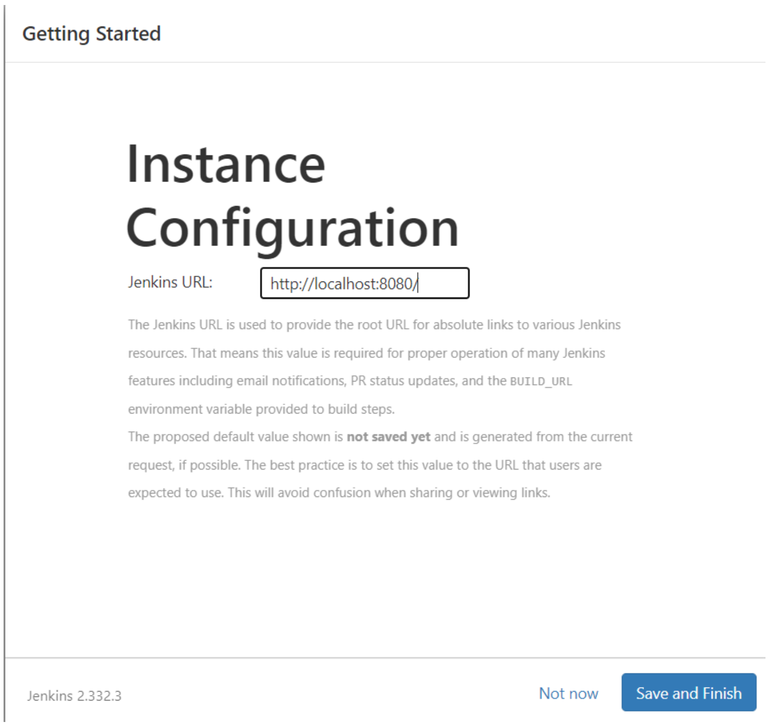

Getting Started

Create First Admin User

Username:	
Password:	
Confirm password:	
Full name:	
E-mail address:	

Jenkins 2.332.3

Provide the required details and create a account


Getting Started

Instance Configuration

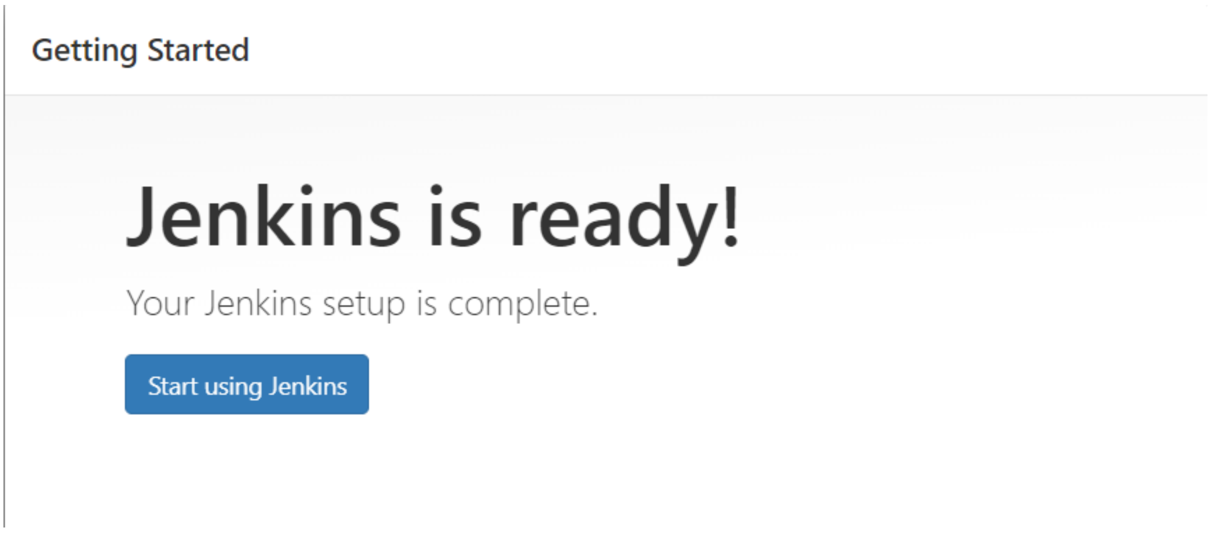
Jenkins URL: http://localhost:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.332.3

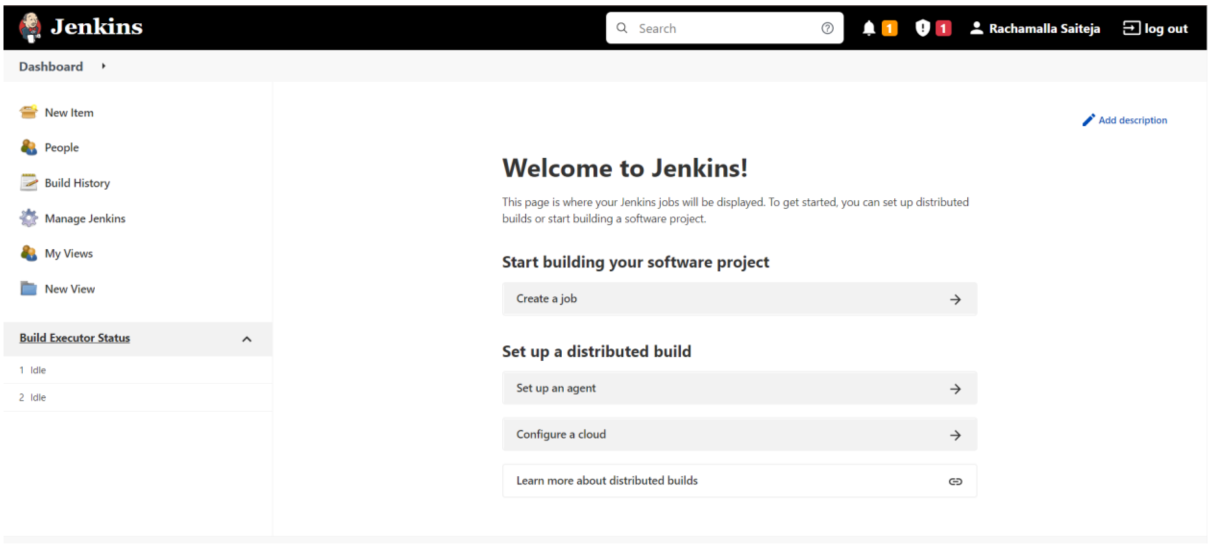
Click on save and Finish.


Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

Click on start using Jenkins your Jenkins dashboard will be displayed.



Jenkins

Search

Rachamalla Saiteja

log out

Dashboard

New Item

People

Build History

Manage Jenkins

My Views

New View

Build Executor Status

1 Idle

2 Idle

Add description

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job

Set up a distributed build

Set up an agent

Configure a cloud

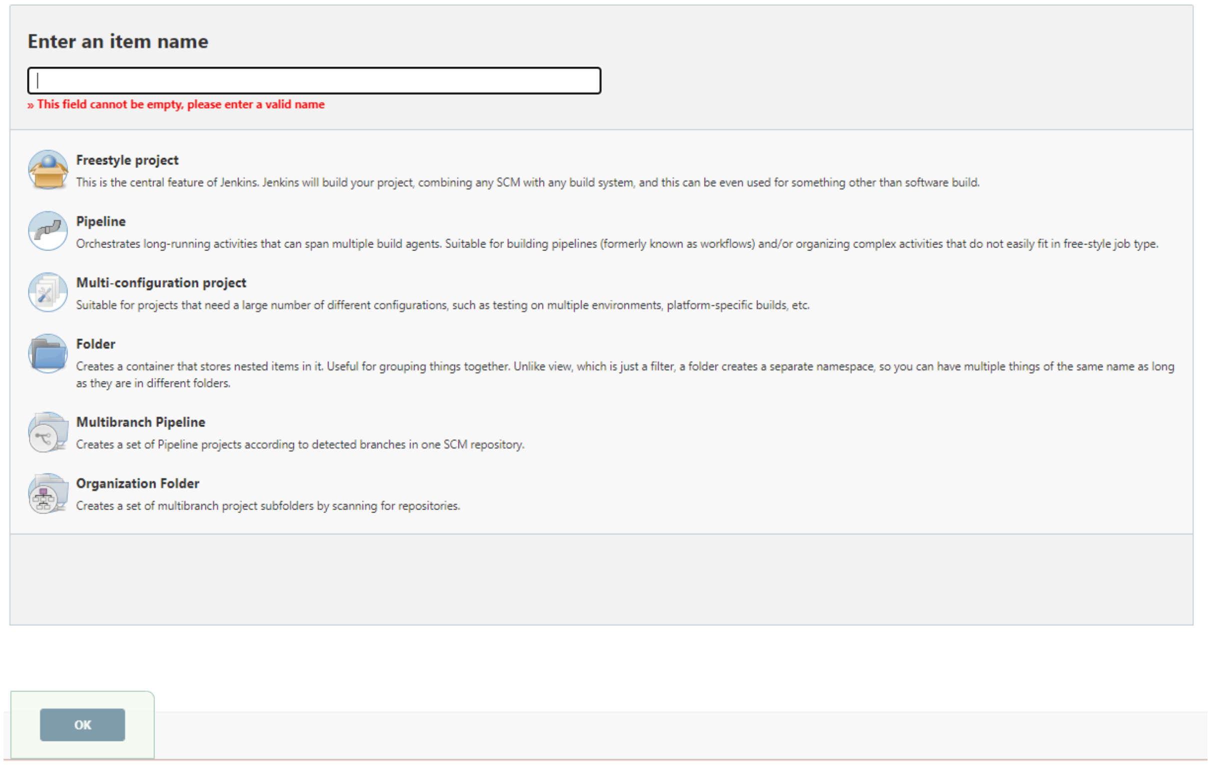
Learn more about distributed builds

How to create a job in Jenkins?

- Why we need to create a job because we have to execute a local project, I don’t want to execute a local project in local machine.
- I want to execute my local project in Jenkins for that we create a job.

By clicking on new item, you can create a job directly.

After clicking on new item another page will shown as Enter an item name.



Enter an item name

» This field cannot be empty, please enter a valid name

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.

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

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.

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

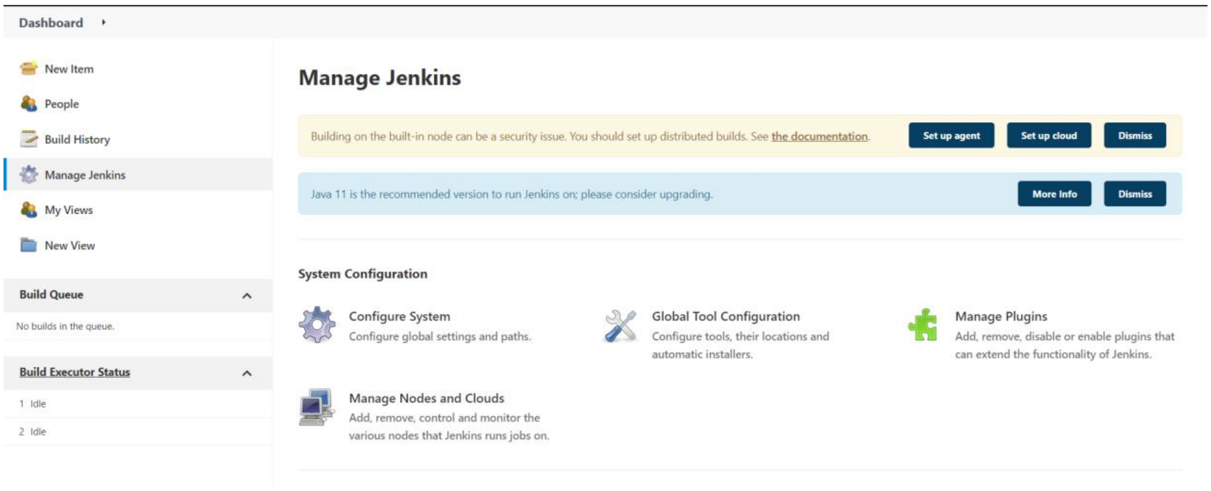
Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

OK

- In the text field you can enter a job name
- If you want to use the freestyle project you can click on free style project option.
- To use pipelines, select pipeline option
- You can select as you required for your project from the shown options.
- But we don’t have maven project option because we need to install maven plugin.

Now I need to create maven project so I will install maven plugin in dashboard.

Go to dashboard click on manage jenkins



Manage Jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation.

Java 11 is the recommended version to run Jenkins on; please consider upgrading.

No builds in the queue.

1 Idle

2 Idle

System Configuration

Configure System
Configure global settings and paths.

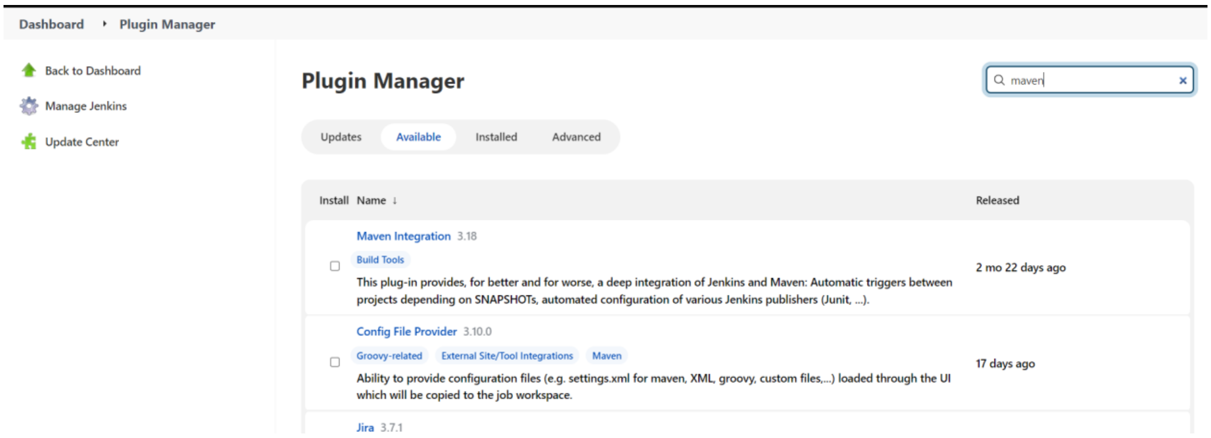
Global Tool Configuration
Configure tools, their locations and automatic installers.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Manage Nodes and Clouds
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

In the manage Jenkins we can see a option called manage plugins.

Click on manage plugins and go to available tab

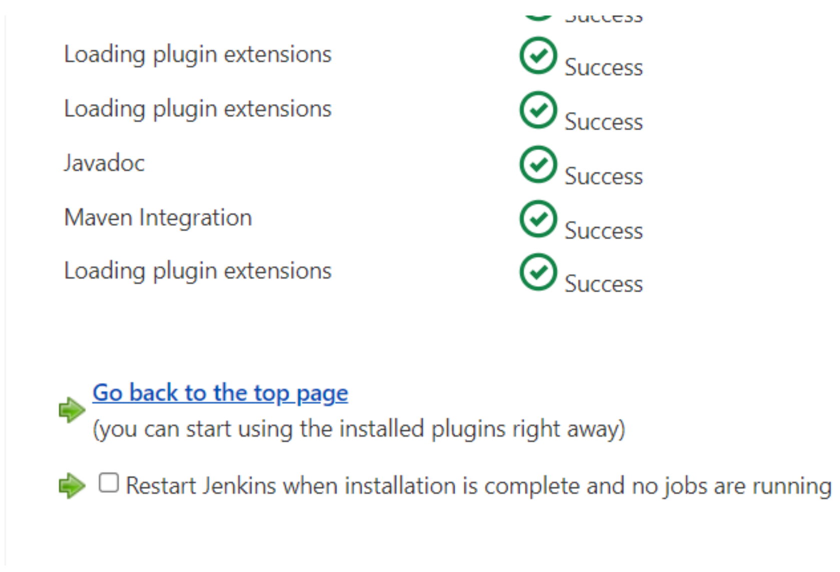


Plugin Manager

Install	Name ↓	Released
☐	Maven Integration 3.18 Build Tools This plug-in provides, for better and for worse, a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTs, automated configuration of various Jenkins publishers (Junit, ...).	2 mo 22 days ago
☐	Config File Provider 3.10.0 Groovy-related · External Site/Tool Integrations · Maven Ability to provide configuration files (e.g. settings.xml for maven, XML, groovy, custom files,...) loaded through the UI which will be copied to the job workspace.	17 days ago
	Jira 3.7.1	

Search maven integration in search bar and click on check box and then click on install with restart button.

An another page will be opened and scroll down you can see maven integration is installed


Loading plugin extensions ✓ Success

Loading plugin extensions ✓ Success

Javadoc ✓ Success

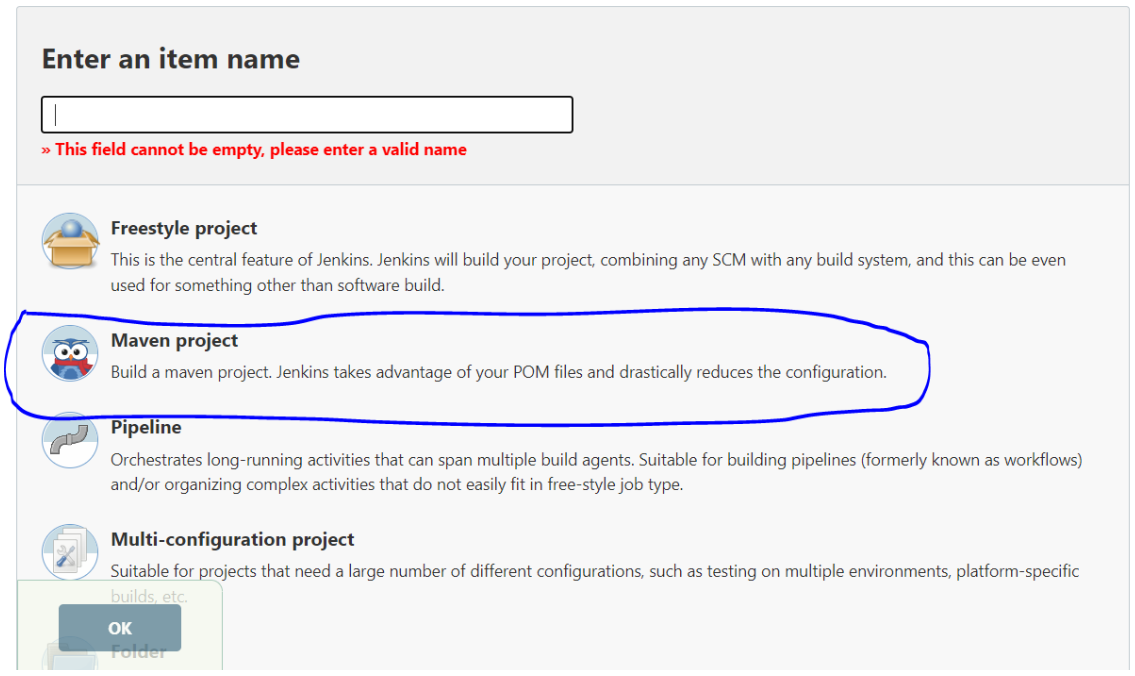
Maven Integration ✓ Success

Loading plugin extensions ✓ Success

Go back to the top page
(you can start using the installed plugins right away)

☐ Restart Jenkins when installation is complete and no jobs are running

After installing successfully go back to the dashboard and click on new item to create a job.


Enter an item name

» This field cannot be empty, please enter a valid name

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 agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

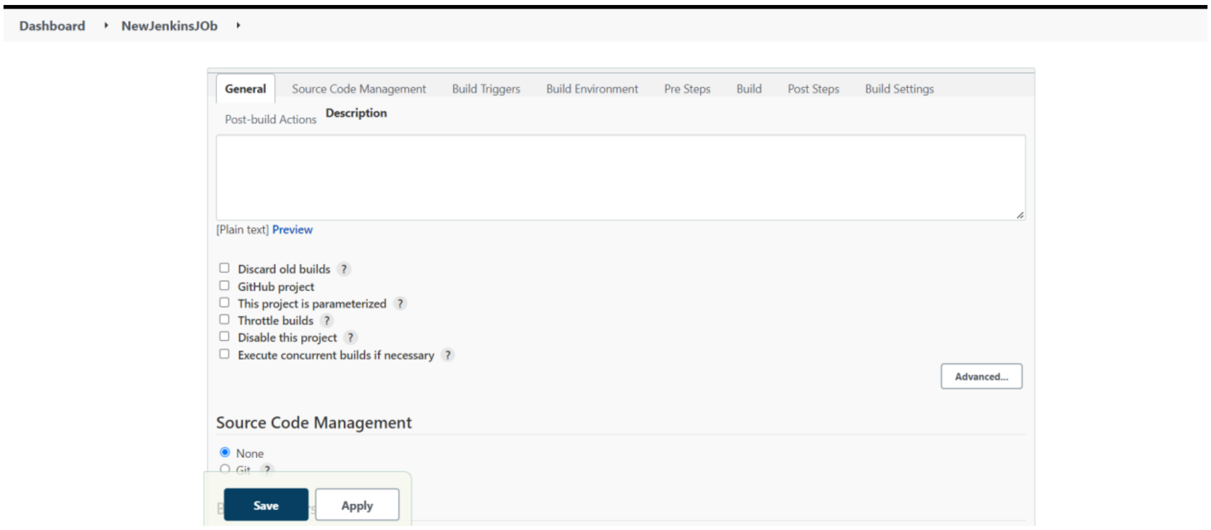
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Now you can see the maven project in a new item page, it means maven plugin is installed.

Now give a job name in text field (Do not give spaces or special characters)

After giving job name select a maven project and click on ok

Then it will be redirected to the configuration window.

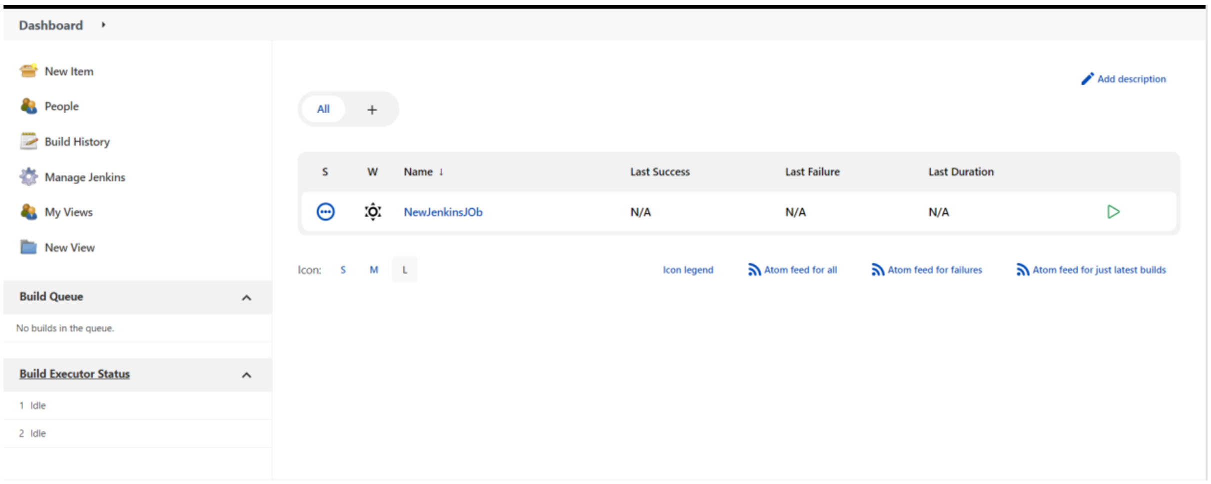


- [ ] Discard old builds
- [ ] GitHub project
- [ ] This project is parameterized
- [ ] Throttle builds
- [ ] Disable this project
- [ ] Execute concurrent builds if necessary

Source Code Management

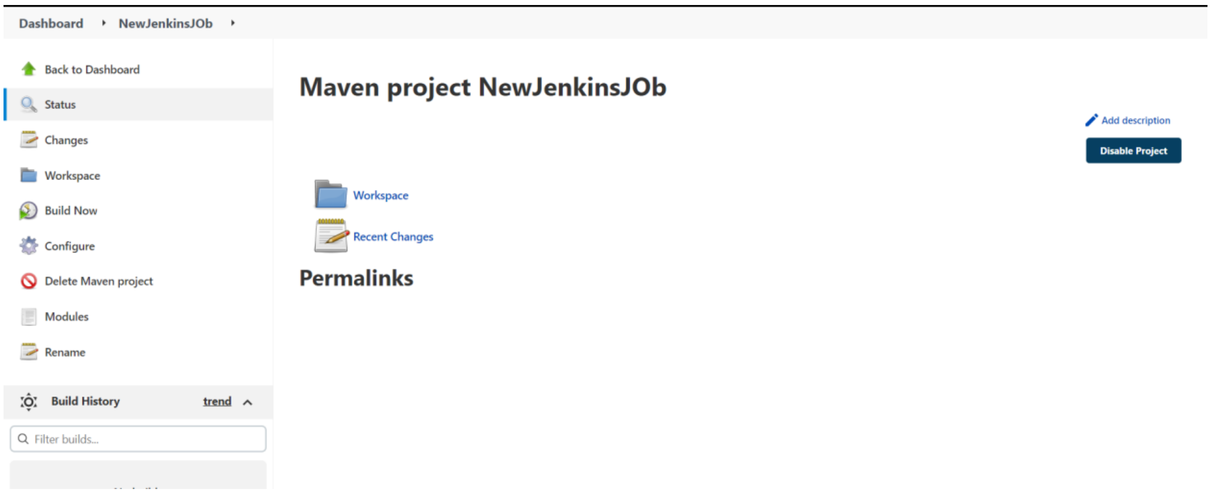
[✓ None] Git

Now you can go back to dashboard. You can see you job in dashboard.



S	W	Name ↓	Last Success	Last Failure	Last Duration
		NewJenkinsJOb	N/A	N/A	N/A

If you click on your job name you can see number of options as

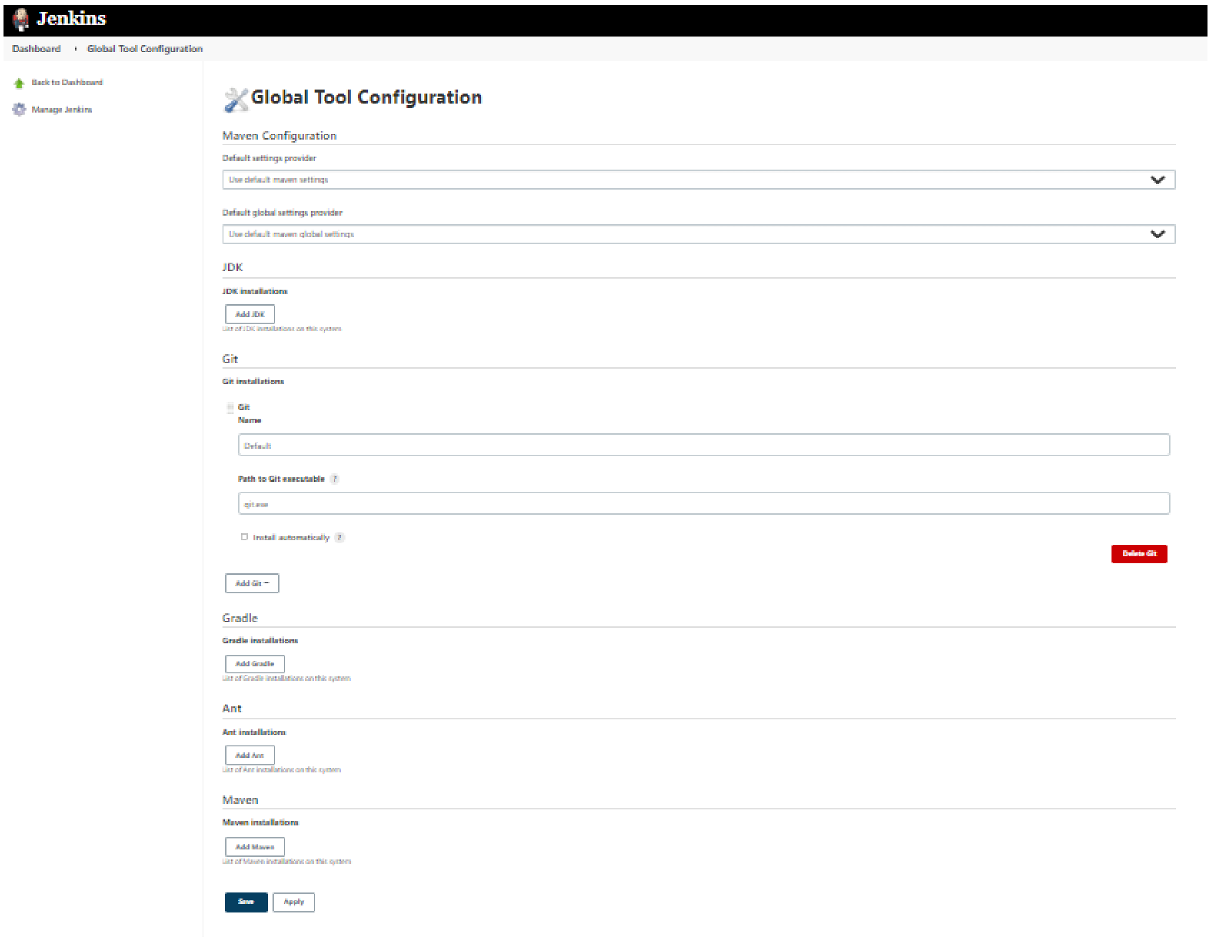


In the options go to configure window. in this window we need to configure everything.

Before configuring we need to set the jdk and maven path in Jenkins.

To set path of jdk and maven in Jenkins we need to go to **manage Jenkins** under manage Jenkins select **Global Configuration option**.

By clicking global configuration



Global Tool Configuration

Maven Configuration

Default settings provider: Use default maven settings

Default global settings provider: Use default maven global settings

JDK

JDK installations

List of JDK installations on this system

Git

Git installations

Git

Name: Default

Path to Git executable: git.exe

☐ Install automatically

Gradle

Gradle installations

List of Gradle installations on this system

Ant

Ant installations

List of Ant installations on this system

Maven

Maven installations

List of Maven installations on this system

You can see number option in the page and also you can see Add jdk and Add Maven

- ➢ Click on Add Jdk

Give name and give jdk path and uncheck install automatically

Jdk path should be up to jdk only not up to bin folder.

Similarly

- ➢ Add Maven by giving maven name and path and uncheck install automatically

Maven path also should not be copied up to bin.

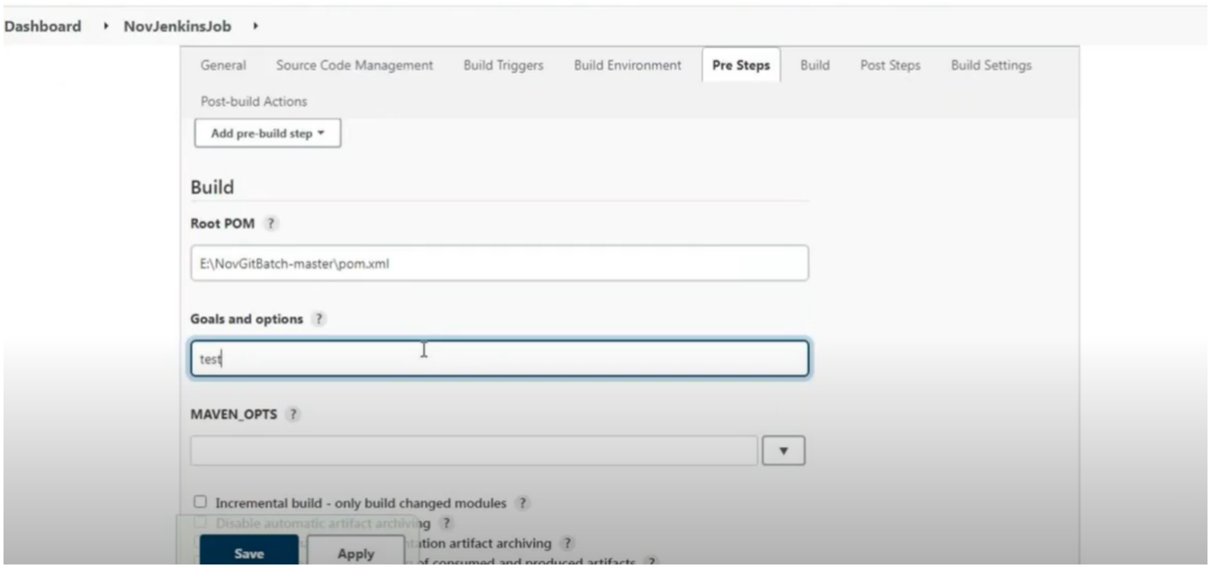
Click on Apply and Save.

Now Go to the dashboard and click on your job and click on configure.

After adding jdk and maven path successfully

Copy the path or location of pom.xml file of your project.

Under configure u can see an option called build



Build

Root POM: E:\NovGitBatch-master\pom.xml

Goals and options: test

MAVEN_OPTS

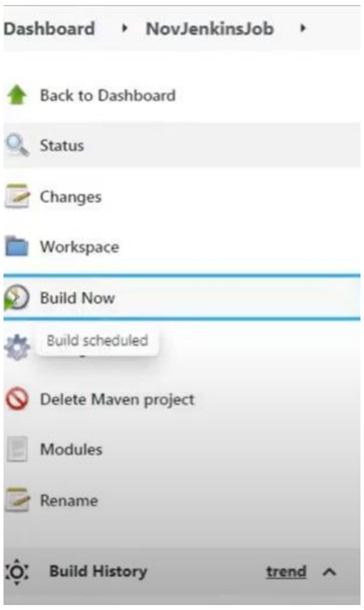
Incremental build - only build changed modules

Disable automatic artifact archiving

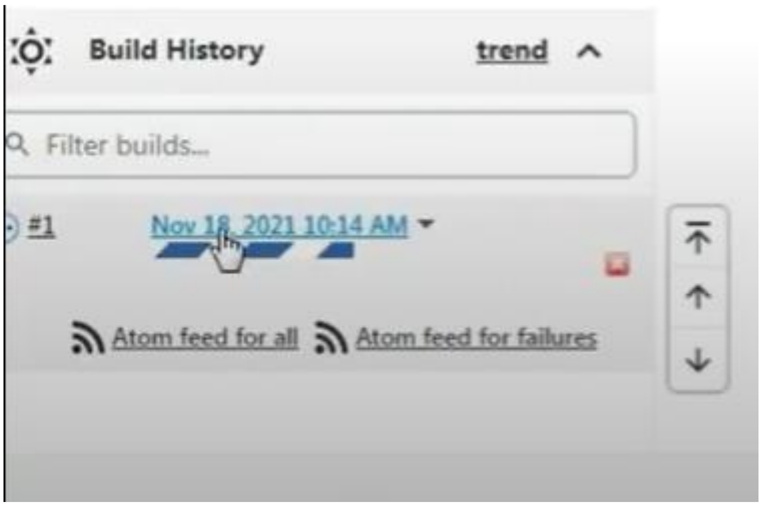
After specifying the location of pom.xml file in Build click on save and apply.

Now we need to execute pom.xml file in Jenkins

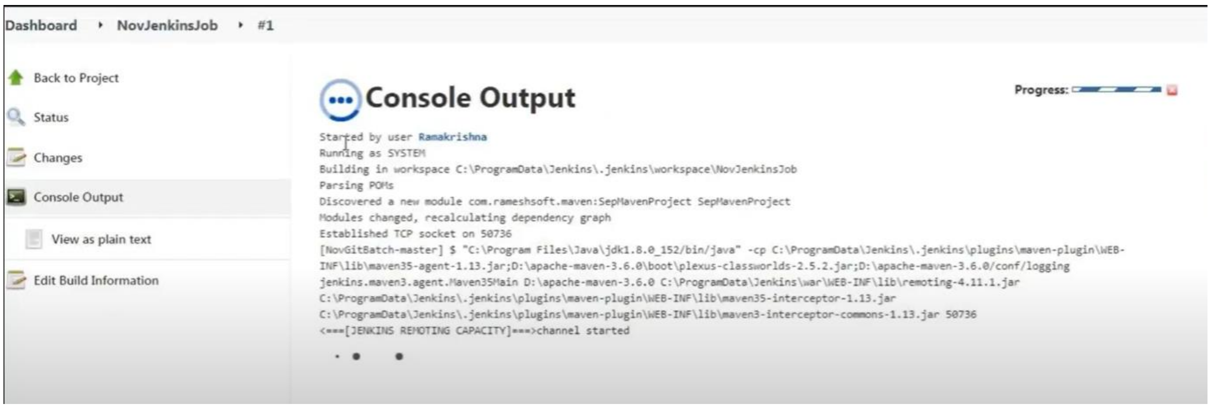
Click on build Now option on dashboard



After click on build now u can see the file is getting loading in dashboard click on that



After clicking on file, a new page will be opened and their u can see many options then click on console Output option

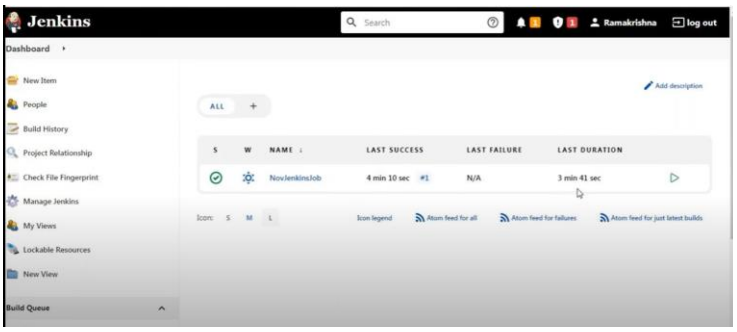


```
Started by user Ramakrishna
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\NovJenkinsJob
Parsing POMs
Discovered a new module com.rameshsoft.maven:SepMavenProject SepMavenProject
Modules changed, recalculating dependency graph
Established TCP socket on 50736
[NovGitBatch-master] $ "C:\Program Files\Java\jdk1.8.0_152/bin/java" -cp C:\ProgramData\Jenkins\.jenkins\plugins\maven-plugin\WEB-
INF\lib\maven35-agent-1.13.jar;D:\apache-maven-3.6.0\boot\plexus-classworlds-2.5.2.jar;D:\apache-maven-3.6.0/conf/logging
jenkins.maven3.agent.Maven35Main D:\apache-maven-3.6.0 C:\ProgramData\Jenkins\war\WEB-INF\lib\remoting-4.11.1.jar
C:\ProgramData\Jenkins\.jenkins\plugins\maven-plugin\WEB-INF\lib\maven35-interceptor-1.13.jar
C:\ProgramData\Jenkins\.jenkins\plugins\maven-plugin\WEB-INF\lib\maven3-interceptor-commons-1.13.jar 50736
<===[JENKINS REMOTING CAPACITY]===>channel started
```

There you can see its going to installing the all dependencies.

After completion of loading all dependencies, you can see build success message now click on file you will navigate to the build history.

Next click on dashboard there you can see the last success and duration .



S	W	NAME ↓	LAST SUCCESS	LAST FAILURE	LAST DURATION
✓	☼	NovJenkinsJob	4 min 10 sec #1	N/A	3 min 41 sec