A screenshot of a cell phone

Description automatically generated

**Introduction**

**Selenium** is an automation tool for Functional Testing of the web-based application. Selenium supports different language like java, ruby, python C#, etc.

<https://www.seleniumhq.org/docs/>

**Cucumber** is a testing approach which supports Behavior Driven Development (BDD). It explains the behavior of the application in a simple English text using Gherkin language.

A cucumber is a tool based on **Behavior Driven Development (BDD) framework** which is used to write acceptance tests for the web application. It allows automation of functional validation in easily readable and understandable format (like plain English) to Business Analysts, Developers, Testers, etc.

<https://cucumber.io/docs/cucumber/>

**Page Object Model** is a design pattern to create **Object Repository** for web UI elements. Under this model, for each web page in the application, there should be corresponding page class. This Page class will find the WebElements of that web page and also contains Page methods which perform operations on those WebElements.

**SerenityBDD** is an open source library used for Behaviour Driven Development and Automated Acceptance Testing. Serenity BDD is commonly used for both Automated Acceptance Tests and Regression Tests.

<http://thucydides.info/docs/serenity-staging/>

Table of Contents

[Setup and Configurations 3](#_Toc20298412)

[**Installing Java (JDK and JRE)** 3](#_Toc20298413)

[Install Eclipse IDE 4](#_Toc20298414)

[Steps to Install Maven in Eclipse IDE 4](#_Toc20298415)

[Installing BDD plugin 7](#_Toc20298416)

[Adding Dependencies 8](#_Toc20298417)

[Adding Cucumber dependencies in pom.xml 8](#_Toc20298418)

[Page Object Model – Design Pattern 12](#_Toc20298419)

[Code Flow 13](#_Toc20298420)

# **Setup and Configurations**

### **Installing Java (JDK and JRE)**

There are mainly four steps needed to follow to install Java on Windows and Mac machine.

1. Download Java
2. Install Java
3. Set Java Environment Path
4. Verify Java Installation

Visit the hyperlink for [Java Installation](https://www.toolsqa.com/selenium-webdriver/install-java/).

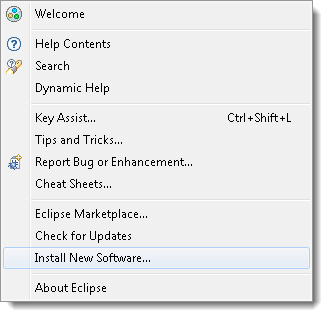
### **Install Eclipse IDE**

Download Eclipse for Java Developers, extract and save it in any drive. It is totally free. You can run ‘eclipse.exe’ directly so you do not need to install Eclipse in your system.

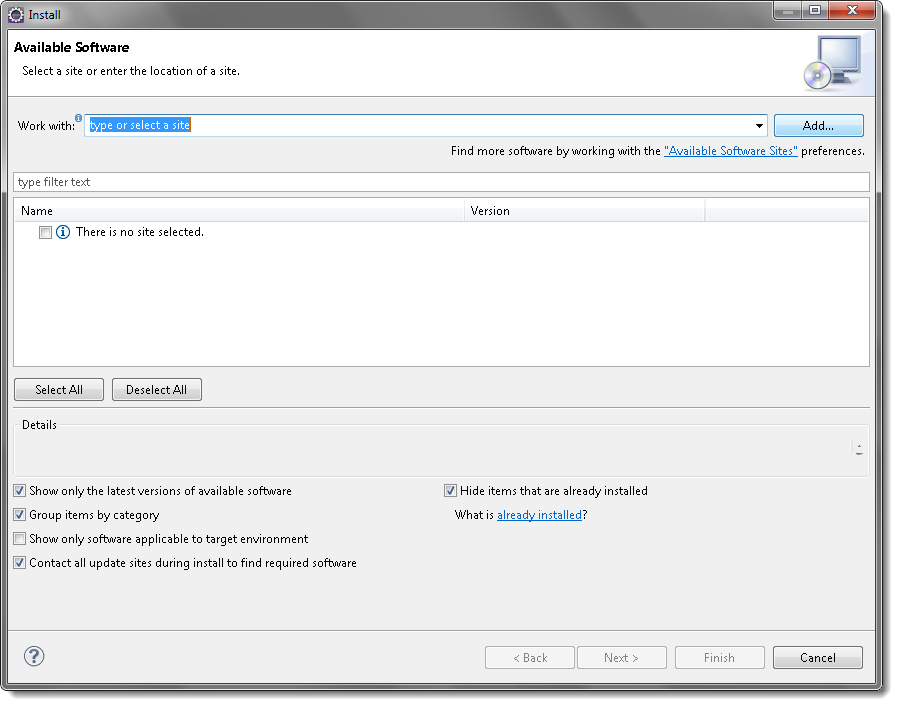
Go to [***http://www.eclipse.org/download***s](https://www.eclipse.org/downloads)

### **Steps to Install Maven in Eclipse IDE**

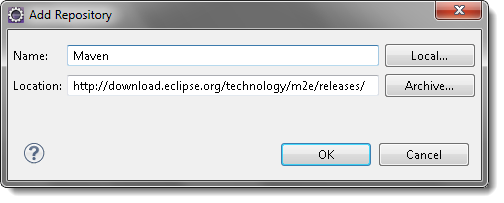
1. *Click* on the ***Help***from the top menu in ***Eclipse***and select ‘***Install New Software*** ‘



1. *Click* on the ***Add***button on the newly opened window.

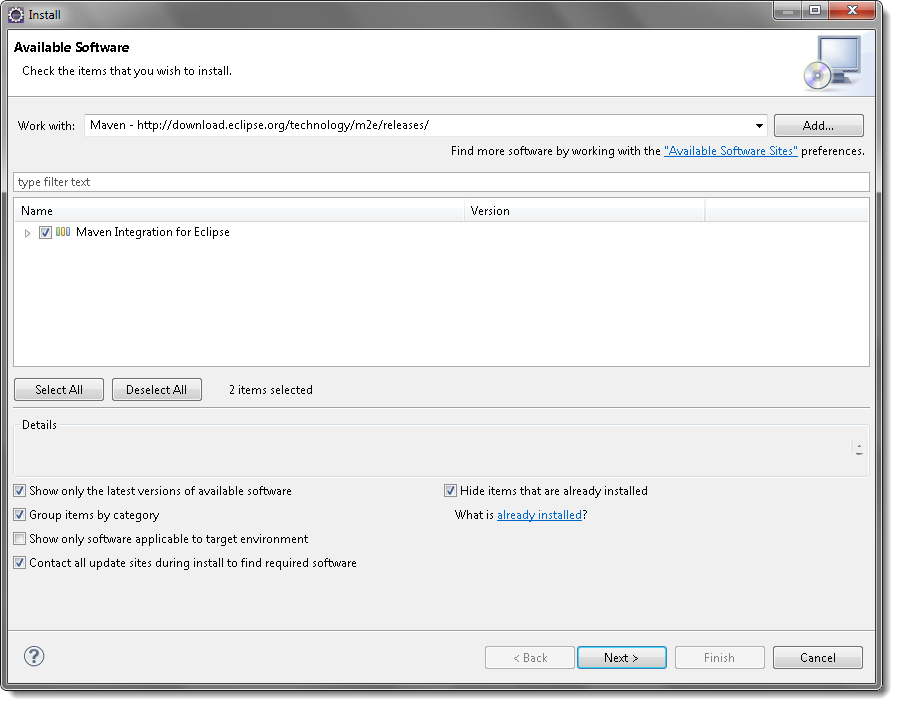


1. In the *Name* box, type ‘**Maven** and in the *Location* box, type ‘***http://download.eclipse.org/technology/m2e/releases/***‘

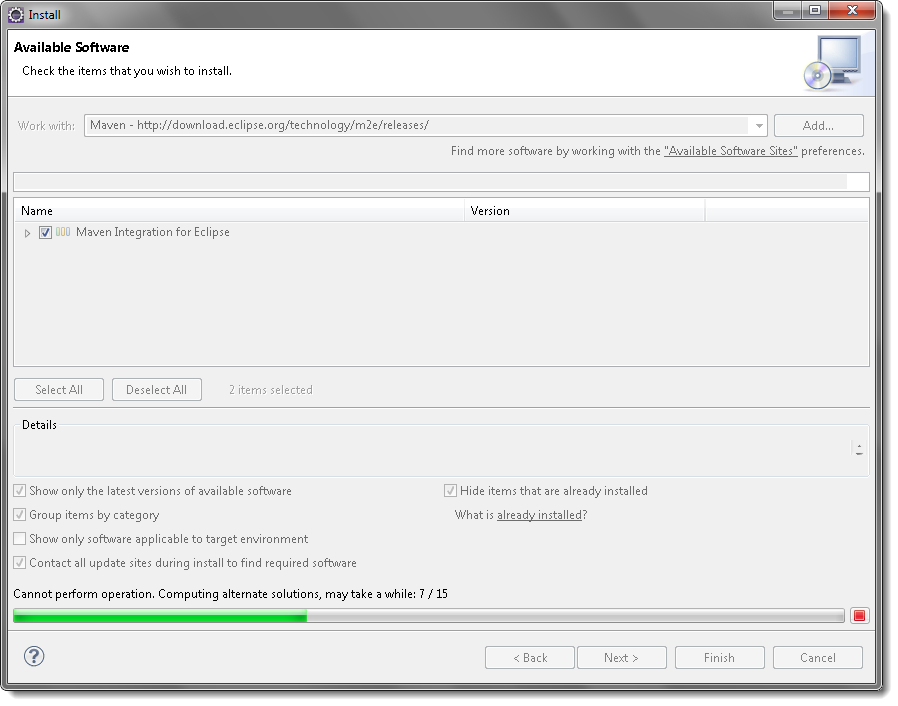


***Note:****The URL is the location where you can download the Maven for Eclipse.*

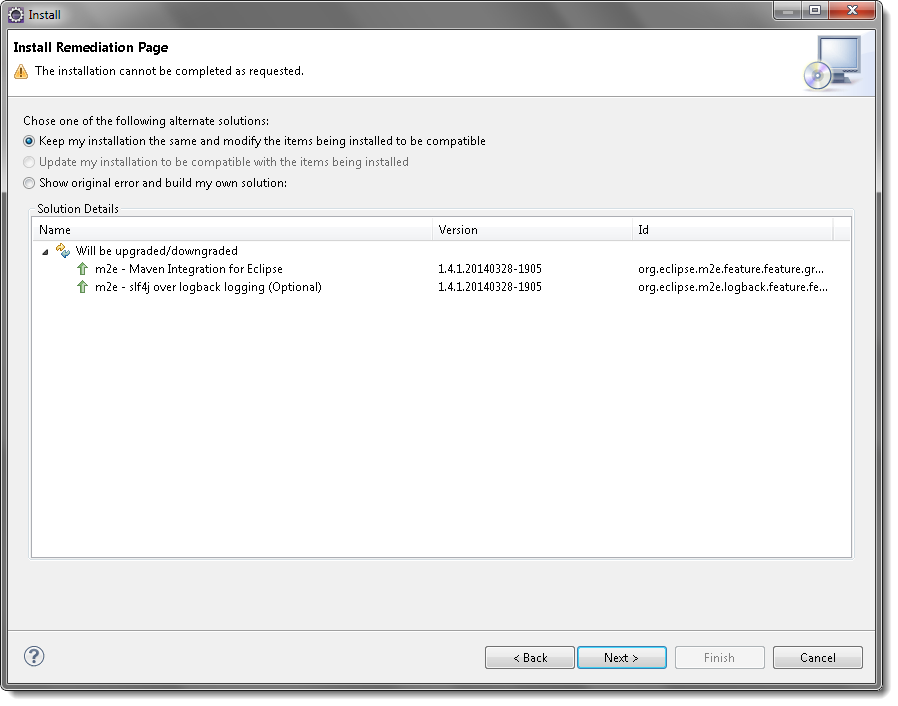
1. A *checkbox* will appear in the pop- window, ***Check***the *checkbox* and *click* on ***Next***button.



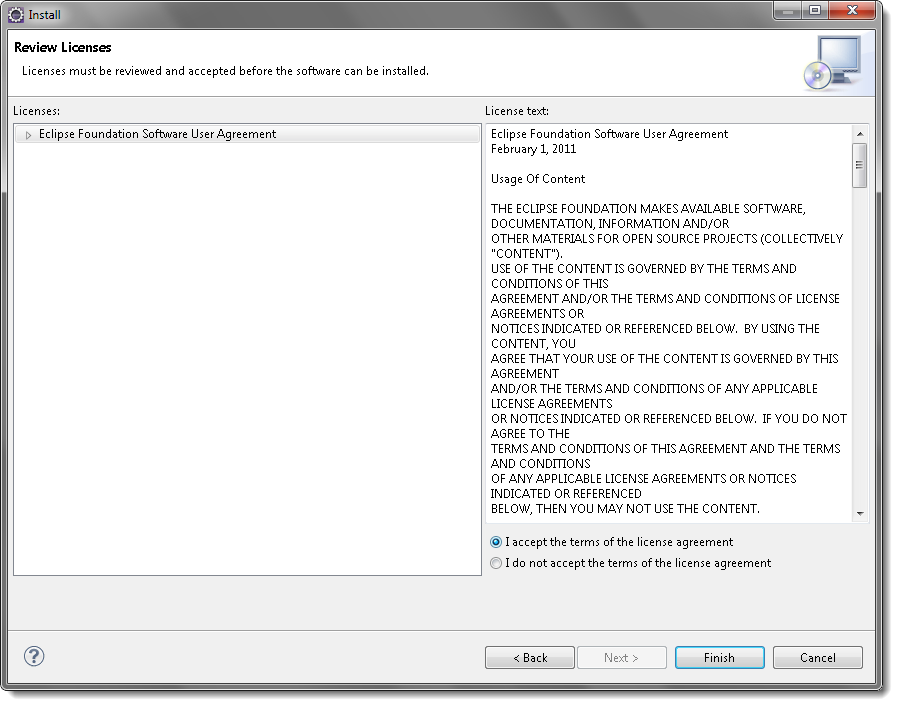
1. Please wait for some time and let the window complete its processing. It will not take long but 2 or 3 minutes.



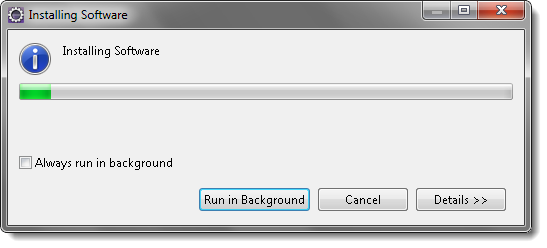
1. Keep the default settings and *click* on ***Next*** button.



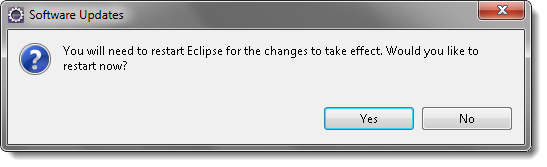
1. Accept the ‘***Terms and Conditions*** and move forward by clicking on ***Finish*** button.



1. Wait while it finishes the installation.



1. Once the installation is finished, it will ask you to *restart* the *Eclipse*. Please *click* on ***Yes***, so that changes can be reflected properly.



This is all for the Installation of Maven in Eclipse.

### **Installing BDD plugin**

1. Go to help > Eclipse Marketplace

A screenshot of a cell phone

Description automatically generated

1. Search for natural plugin, install it and restart eclipse

A screenshot of a cell phone

Description automatically generated

# **Adding Dependencies**

## **Adding Cucumber dependencies in pom.xml**

1. Go to package explorer on the left-hand side of Eclipse.
2. Expand the project
3. Locate **pom.xml** file
4. Add dependency for selenium

A screenshot of a cell phone

Description automatically generated

### Add dependency for Cucumber-Java

A close up of a logo

Description automatically generated

### Add dependency for cucumber-Junit

A close up of a logo

Description automatically generated

### Adding serenity dependencies

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

# **Page Object Model – Design Pattern**

**A screenshot of a social media post

Description automatically generated**

POM.xml consists of all the dependencies that needs to be included to build the project

This is a properties file, global serenity properties that are used

This package consists of runner file

This package consists of steps file

This package has utilities file

This package has all the code for different webpages

References :

<https://medium.com/tech-tajawal/page-object-model-pom-design-pattern-f9588630800b>

<https://www.guru99.com/page-object-model-pom-page-factory-in-selenium-ultimate-guide.html>

# **Code Flow**

1. Since, Maven is used for execution;

I have defined a goal for maven execution in run configurations,

Go to project properties and click run as > run configurations

A pop-up window will open;

A screenshot of a cell phone

Description automatically generated

Here in Goals: I have described cucumber.options, you can read more about it here

<https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html>

1. The execution begins from runner class which is under com.iRobotCI.cucumber package.

References to learn more about runner class;

<http://www.automationtestinghub.com/cucumber-test-runner-class-junit/>

<https://www.toolsqa.com/cucumber/junit-test-runner-class/>

1. From runner class, the execution goes to feature file which is defined under cucumber options in runner class. Then whatever written under @CI tag will be executed
2. Feature file: the gherkin syntax.

In Behavior Driven Development, we use gherkin syntax to write tests. The goal of BDD is a business readable and domain-specific language that allows you to describe a system’s behavior without explaining how that behavior is implemented.

A screenshot of a social media post

Description automatically generated

<https://cucumber.io/docs/gherkin/reference/>

Here the data is passed in delimiter format, which is passed as parameter in each step under scenario outline

1. Steps file: steps written in feature file are plain English and that needs to be glued to some java code. We define those methods in steps file. Each step written under scenario outline in feature file is glued to each corresponding step in iRobot\_steps.java file.

A screenshot of a computer

Description automatically generated

1. Object repository function definition for each are written under com.iRobotCIcucumber.serenity/ package.

A screenshot of a computer

Description automatically generated

1. You can generate gherkin steps by mentioning a tag “**dryRun =true”** under CucumberOptions in runner class.

A screenshot of a cell phone

Description automatically generated

# **Reporting**

## Excel Report

Whenever execution gets completed, an excel file with all the details related to all the orders placed in that execution gets generated.

I have written its code under com.iRobotCI.utils/ExcelWriter and the method to write in excel file has been invoked from com.iRobotCI.testbase/testbase class.

A screenshot of a social media post

Description automatically generated

The @AfterClass tag will call the function in the end of the execution which will invoke the writer method written in ExcelWriter class.

Location is given in the excel writer function where the generated excel file going to be dumped.

Serenity framework also generates an html report under folder target > site > reports in index.html

Location on local drive:

[/Users/pjain/eclipse-workspace/iRobotCI/target/site/reports/index.html](file:///Users/pjain/eclipse-workspace/iRobotCI/target/site/reports/index.html)

## Serenity Reports:

The html reports which is generated after execution consists of all the details of scenarios along with screenshots of the steps failing.

The default directory of serenity reports target > site > reports in index.html

You can change the output directory in serenity.properties file.

Change the value of this property if you want to update the output directory:

serenity.outputDirectory

You can also modify the behavior of screenshots through this property in serenity.properties file: serenity.take.screenshots

## Centralizing framework on bitbucket:

<https://bbdm.irobot.com/login?next=/projects/AT/repos/testing-framework/browse>

Shared Network drive URL to access the frameworks:

Mac OSx:

1. Navigate to the top left pane of your screen and please **click** on **Go**
2. Scroll down to the last option and **click**on **Connect to Server**
3. With the server address, please apply the following network path - <smb://hq-nas-01/media/macserve-files/web-dev-team>
4. Click connect

Windows 7/10:

1. Within your start menu or pinned to your taskbar, please **type in** or **click** on **file explorer**
2. You have two options to either browse directly to the network share or Map a Network drive from File Explorer  **This PC** menu
3. Network shared path - [\\hq-nas-01\media\macserve-files\web-dev-team](file:///hq-nas-01/media/macserve-files/web-dev-team)