# Bootcamp Prerequisites and Preparation

1. Install and use Java 7 SDK
2. Install Eclipse and use as your IDE
3. Send email to DOJO Support to get read access to SVN Project <https://del.tools.sapient.com/svn/Cucumber-training/>
4. Setup your userid in Jira, then send request to Shobhit Gupta 2 and have him give you access to training instance of Jira

# Java Resources

1. Safari: Maven: The Definitive Guide
2. Safari: Apache Maven 3 Cookbook
3. Safari: Getting Started with Apache Maven [Video]
4. Safari: Beginning Java 7 : Chapters 1-6, 8 as a minimum
5. Safari: JUnit in Action, Second Edition
6. Vox: The Cucumber Book, read first 5 chapters: <https://vox.sapient.com/docs/DOC-84261>

# Setup/Installation Guide

This document will guide you through the setup of the following:

1. Eclipse Maven plugin
2. Maven command line
3. Eclipse Gherkin plugin
4. Eclipse source code projects for Behavior Driven Development (BDD)/Cucumber bootcamp

**Pre-Requisite**: JDK 7 should be installed and environment variables should be properly set.

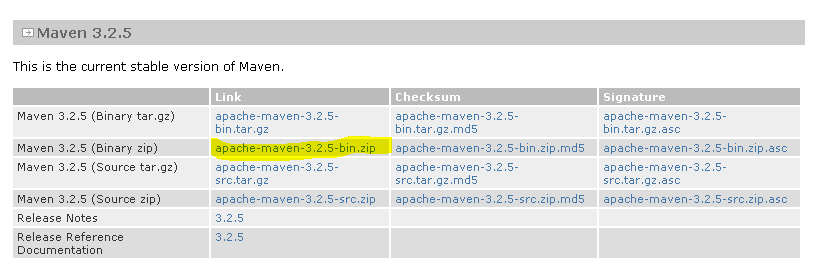
## Maven Installation

### 1) Maven Download

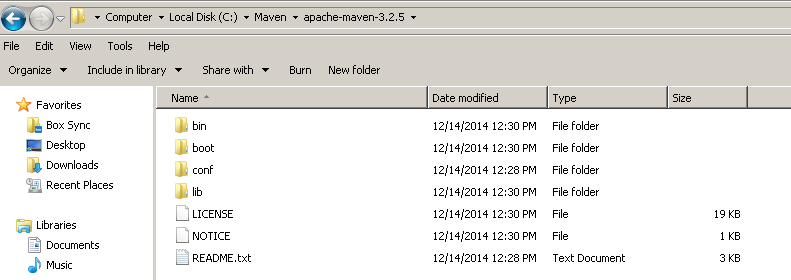
The latest version of Maven can be downloaded from the below mentioned url

<http://maven.apache.org/download.cgi>

Download the Maven 3.2.5(Binary zip) and save it to folder at your system.

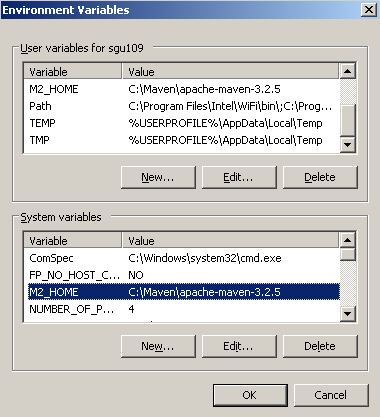


Assume you unzip to this folder – C:\Maven\apache-maven-3.2.5



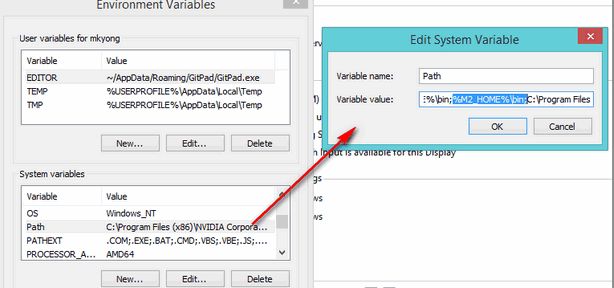
### 2) Adding System Variable

Add both M2\_HOME and MAVEN\_HOME variables in the Windows environment, and point it to your Maven folder

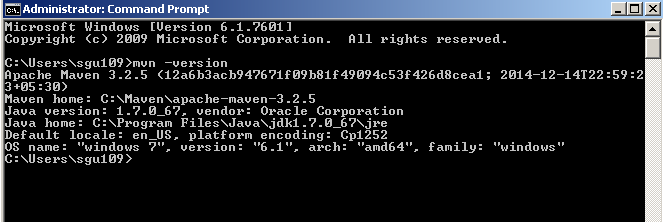


### 3) Setting up Path variable

Update PATH variable, append Maven bin folder – %M2\_HOME%\bin, so that you can run the Maven’s command everywhere.



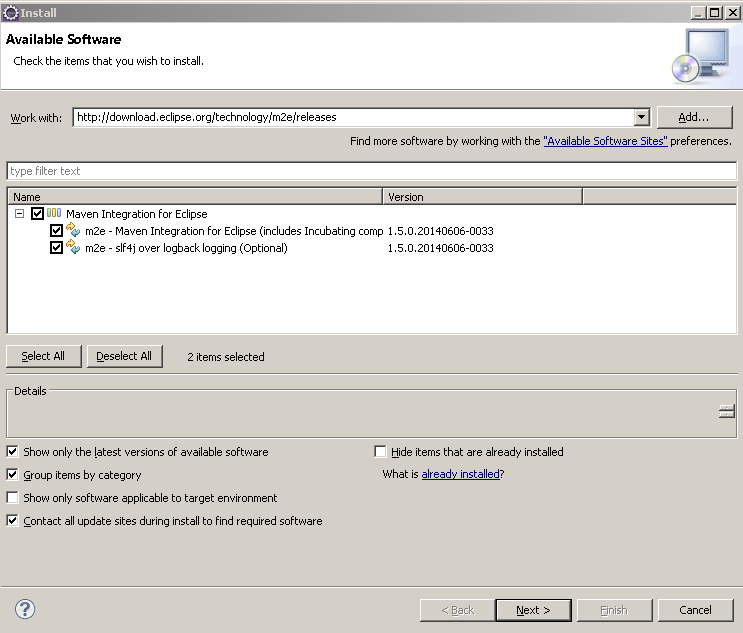
To verify the installation run mvn –version in the command prompt and you will get its version as shown below:



## Maven Eclipse Plugin

Maven plugin allows you to create Maven project in Eclipse and also manage the classpath with Maven dependencies

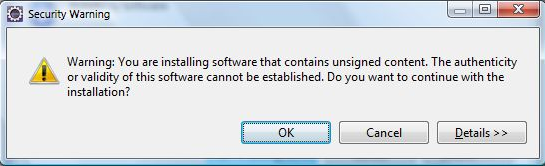
Start Eclipse and go to **Help -> Install New Software…**



Copy the url and paste it under “Work With” and hit enter. Select the “Maven Integration for Eclipse” as shown in above image.

Click on Next button to follow the installation.

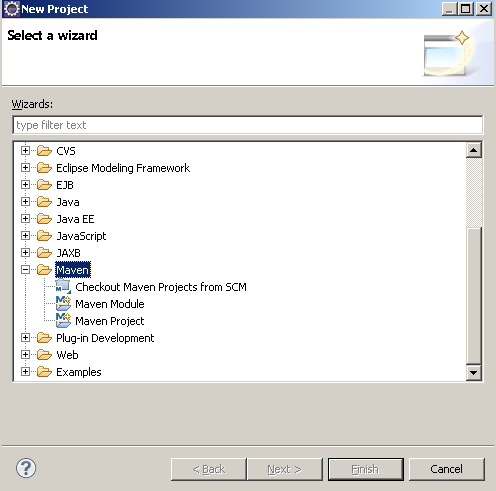
If you get any warning message when installing, click **OK** to continue.



Once the installation is complete it will ask you to restart Eclipse.

After installing the Maven plugin for Eclipse, you can check if the installation is successful by creating a new Maven project.

You should see a Maven folder in the New Project wizard as shown below.



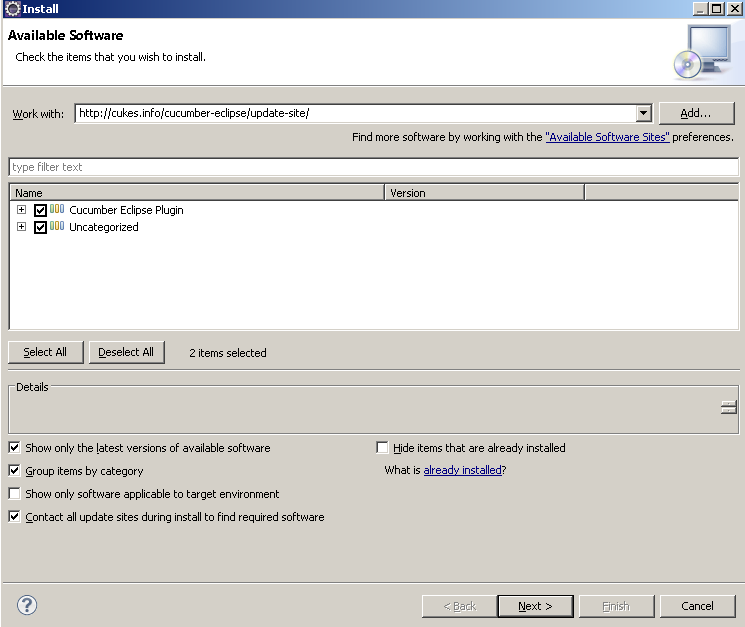
## Gherkin Plugin for Eclipse

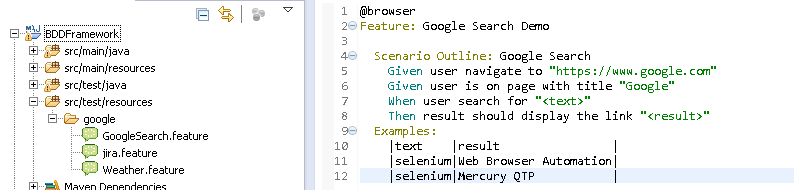
Gherkin plugin let you manage Cucumber features files in a more structured way by identifying various statements and giving an improved presentation of the feature file.

In Eclipse go to **Help -> Install New Software** … and use the below mentioned url to install the plugin

<http://cucumber.github.com/cucumber-eclipse/update-site>

Use the steps as discussed in the maven plugin installation.



After you restart the Eclipse all the features files will be displayed in more structured manner 

Note the different color codes are being used for different gherkin statements.

## Importing Cucumber Project in eclipse

Cucumber examples used for training are checked into svn under the below url:

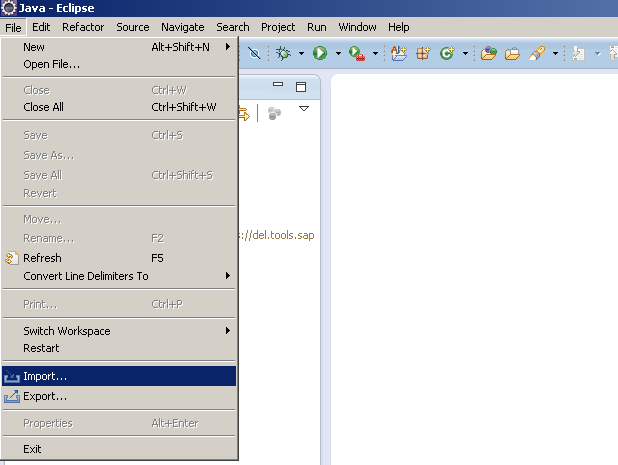
<https://del.tools.sapient.com/svn/Cucumber-training/>

Checkout the code base from the SVN to a folder at your local system.

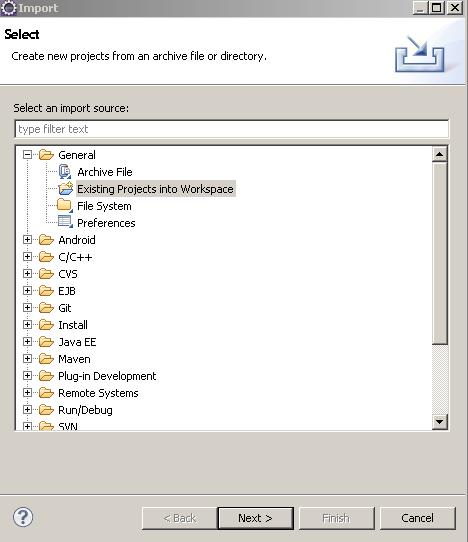
Let’s assume you check out the code to **D:\ Cucumber-training\sapient-cucumber-example**

Import the project into Eclipse using steps mentioned below :

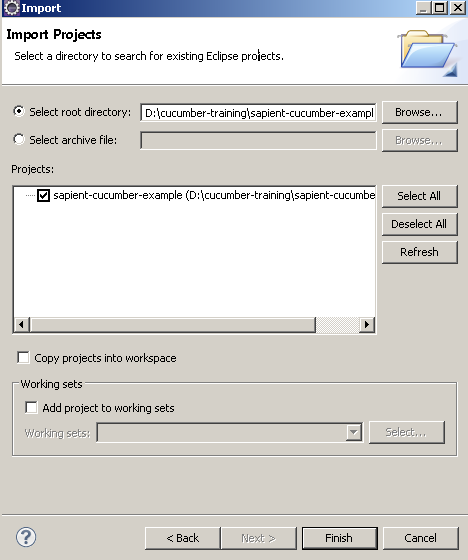
1) Open Eclipse and go to **File->Import**



2) Select **General->Existing Project** into workspace and click Next



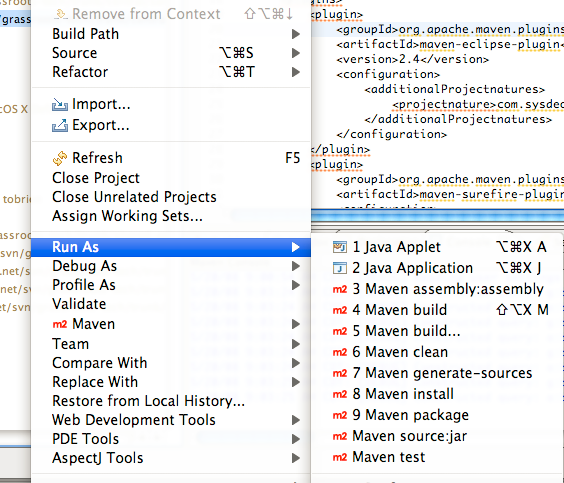
3) Under Import Projects window, in Select root directory browse to the folder where you have checked out the SVN project and click **Finish**



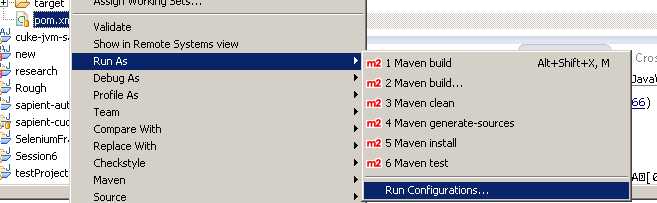
The project will be added under the project explorer. There will be a pom.xml in the project which will download all the dependencies using Maven plugin.

## Setting Maven Run configuration

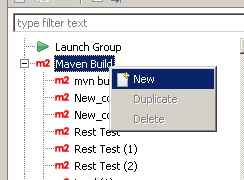
Maven Eclipse plugin adds a few more options under “Run As” and “Debug As”. From this menu you can run one of the more common Maven lifecycle phases like clean, install, or package. You can also load up the Run configuration dialog window and configure a Maven build with parameters and more options.



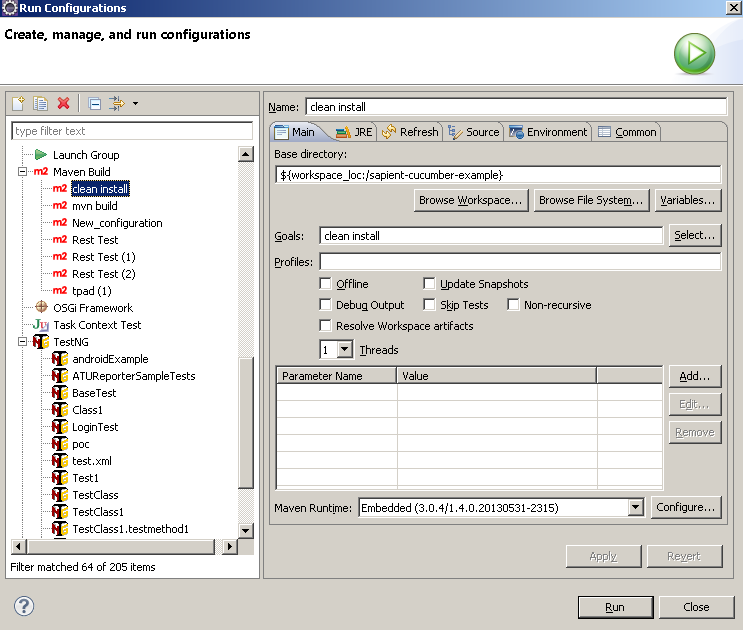
Maven build can be configured with more options using “Run Configuration”. To configure the options right click on the pom.xml and select “Run As - > Run Configuration”



Under Run Configuration window, right click on Maven Build and select New



Add the configuration as displayed below :



Use this for the name to signify project build: “BUILD sapient-cucumber-example”. Under Base directory select the project root folder. Under Goals specify the goal as “clean install”.

Similarly add new configuration that will add Maven dependencies to the Eclipse project

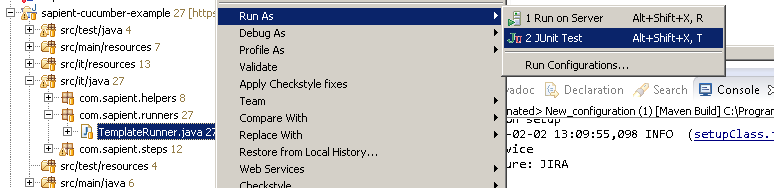
Name: “Eclipse sapient-cucumber-example.

Goal: “eclipse:eclipse”

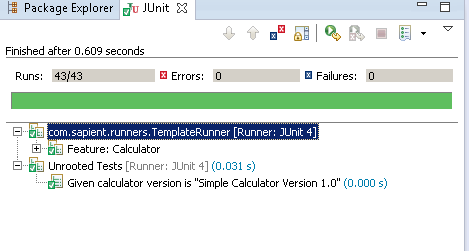
## Setup Validation

### Using JUnit Runner

Run the “TemplateRunner.java” as JUnit Test as shown below in the image

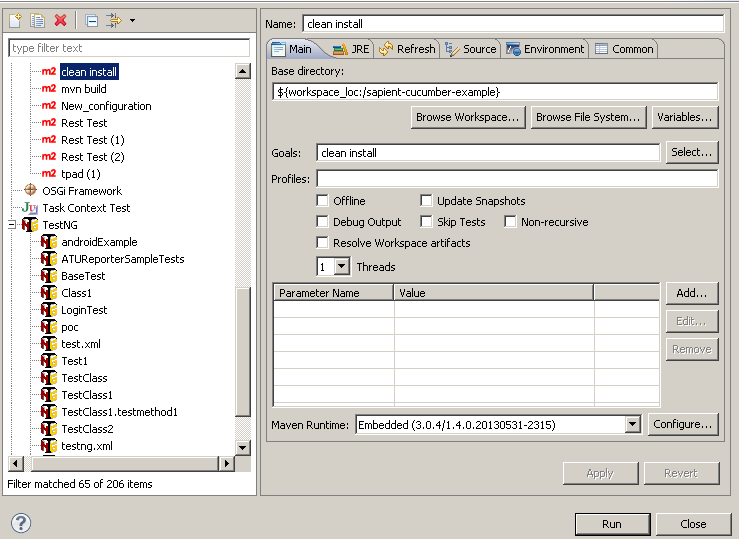


Now in the console you will some execution being done and after the execution if you open the JUnit tab , you will notice the status as shown below



### Running from Eclipse using Maven configuration

Right click at the pom.xml and select “Run As – Run Configuration”. Now select the configuration “Build sapient-cucumber-example” and click “Run”

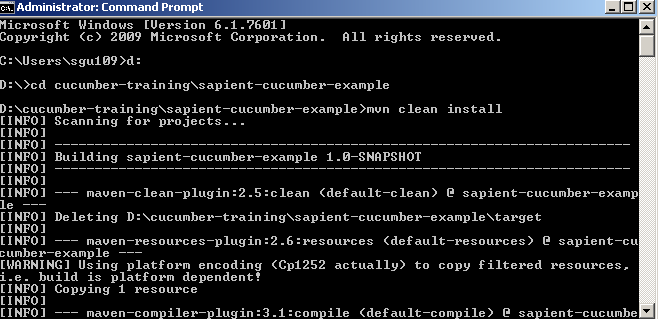


In the console you will get the output of the execution.

### Running through Command Prompt

In the command prompt navigate to the root folder of the project having the pom.xml and on command prompt run the following command :

mvn clean install



Also after the execution is complete you will see the “Build Success” message in console

