

Way2Automation - Tutorial 2 – Setting up cucumber in eclipse

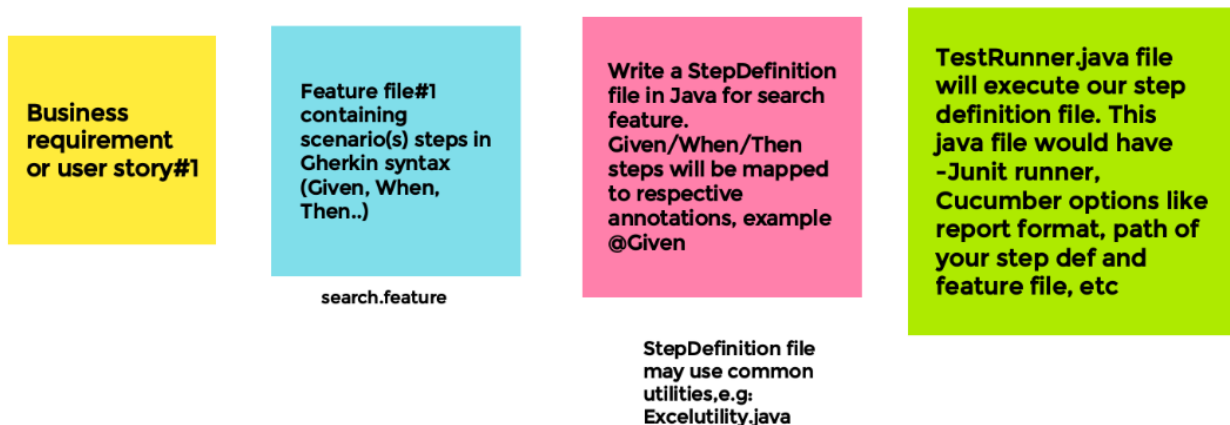
What you will Learn :

- o Cucumber-BDD Framework design flow
- o Create Maven project in eclipse
- o Download cucumber eclipse plugin
- o Delete the dummy packages
- o Add properties to pom.xml
- o Add dependencies to pom.xml
- o Add plugins to pom.xml
- o Update maven project

Cucumber-BDD Framework Design Flow

First, let us understand the flow that we would be designing in our cucumber framework. Look at the below figure. So basically, we would be given a document that would comprise of many functional requirements or **user stories**. So, this document would describe various requirements such as: how to perform a search for a particular product? how to add a product to a cart? and so on...

We would then convert or map this requirement to the respective **feature file**. The feature file would then be mapped to a **step definition** java file. Lastly, **Testrunner** java file would execute respective features. For each user story, we would write the corresponding feature file, step def file and Test Runner. So, memorize this design flow (user story > feature file > Step def java > Test runner java) while designing framework



Business requirement or user story#2

Feature file#2 containing scenario(s) steps in Gherkin syntax (Given, When, Then..)

addTocart.feature

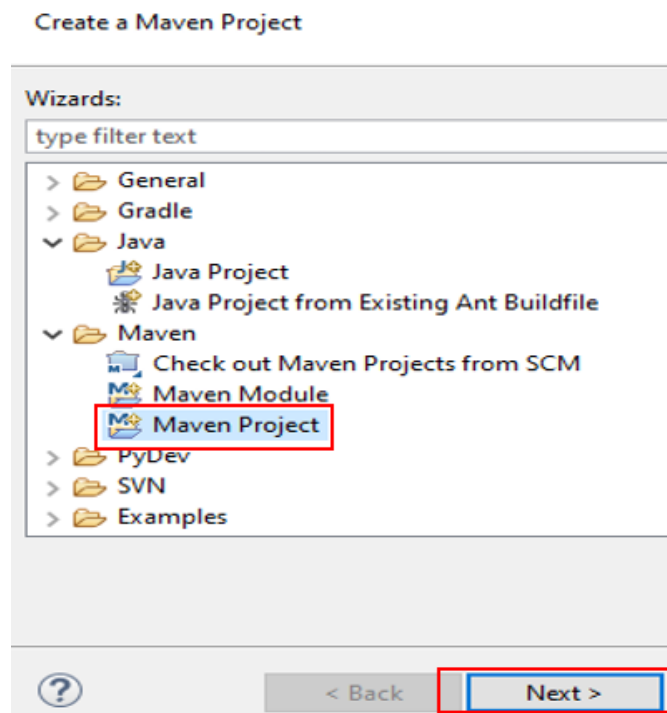
Write a StepDefinition file in Java for search feature. Given/When/Then steps will be mapped to respective annotations, example @Given

StepDefinition file may use common utilities, e.g: ExcelUtility.java

TestRunner.java file will execute our step definition file. This java file would have -JUnit runner, Cucumber options like report format, path of your step def and feature file, etc

Create Maven project in eclipse

Open eclipse, File >New >Project >Select 'Maven Project', see below



Click Next

New Maven project
Select project name and location

☐ Create a simple project (skip archetype selection)

☒ Use default Workspace location
Location:

☐ Add project(s) to working set
Working set:

► **Advanced**

Click Next. In the below window, type 'maven-archetype-quickstart' in the 'Filter' text field. Select the group id as shown below. It does not matter which version you see (here 1.1)

Catalog: All Catalogs

Filter:

Group Id	Artifact Id	Version
org.apache.maven.archetypes	maven-archetype-quickstart	1.1

An archetype which contains a sample Maven project.

☒ Show the last version of Archetype only ☐ Include snapshot archetypes

► **Advanced**

Click Next. Mention any desired 'Group Id' and 'Artifact Id'. The name you mention in 'Artifact Id' would become the name of your maven project.

Specify Archetype parameters

Group Id:

Artifact Id:

Version:

Package:

Properties available from archetype:

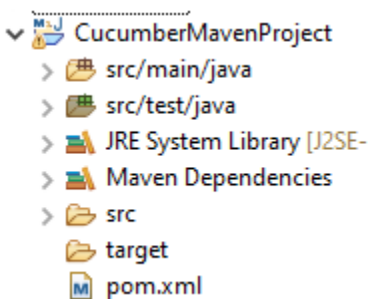
Name	Value

► Advanced

? < Back Next > Finish

Click Finish

Wait for a minute or so. You should see a maven project getting created having the same name as 'Artifiact Id'. See below



You should see the same folder structure that you see above

Download cucumber eclipse plugin

This plugin is very useful that will help us to create multiple feature files, it will highlight all the features/annotations in respective colors, the readability will be improved.

So just copy the url <http://cucumber.github.io/cucumber-eclipse/update-site/>



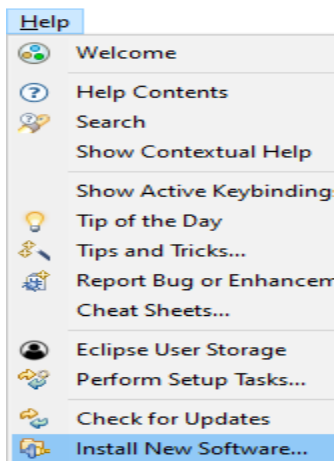
Eclipse plugin

- [Home](#)
- [News](#)
-

This is the Eclipse update site for the Cucumber-Eclipse plugin.

For more information about installing or updating the plugin, please see the [Eclipse Platform Help](#).

In the eclipse, Help > Install New Software



Paste the url in the 'Work with' field as shown below

Available Software

Check the items that you wish to install.



Hit 'Enter' key

You would see a checkbox named 'Cucumber Eclipse Plugin' as seen below


Available Software

Check the items that you wish to install.


Work with:

type filter text

Name

> ☐  Cucumber Eclipse Plugin

Select the checkbox 'Cucumber Eclipse Plugin' as seen below

Name	Version
> <input checked="" type="checkbox"/>  Cucumber Eclipse Plugin	

item selected


Details

☒ Show only the latest versions of available software ☐ Hide items that are already installed

☒ Group items by category What is [already installed?](#)

☐ Show only software applicable to target environment

☒ Contact all update sites during install to find required software

 < Back **Next >**


As soon as you select the checkbox, 'Next' button gets enabled as seen above

Click 'Next'

After a minute or so, the below window would come up


Install Details

Review the items to be installed.

Name	Version	Id
 Cucumber Eclipse Feature	0.0.23.201811220126	cucumber

Size: Unknown

Details

 < Back Next >

Click 'Next'

Accept the license terms

 Install

Review Licenses

Licenses must be reviewed and accepted before the software can be installed.

License text (for Cucumber Eclipse Feature 0.0.23.201811220126):

The MIT License


Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

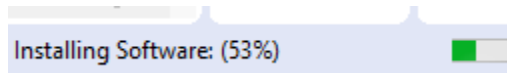
☒ I accept the terms of the license agreement

☐ I do not accept the terms of the license agreement

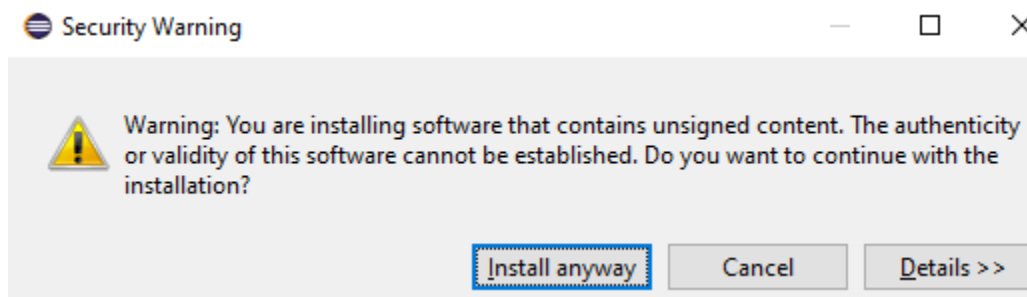
 < Back Next > Finish

Click Finish

The s/w starts getting installed (see right hand side bottom screen of eclipse)

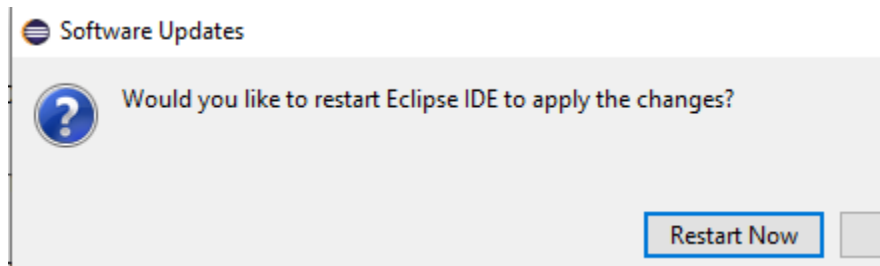


After sometime you will get below popup



Click 'Install anyway'

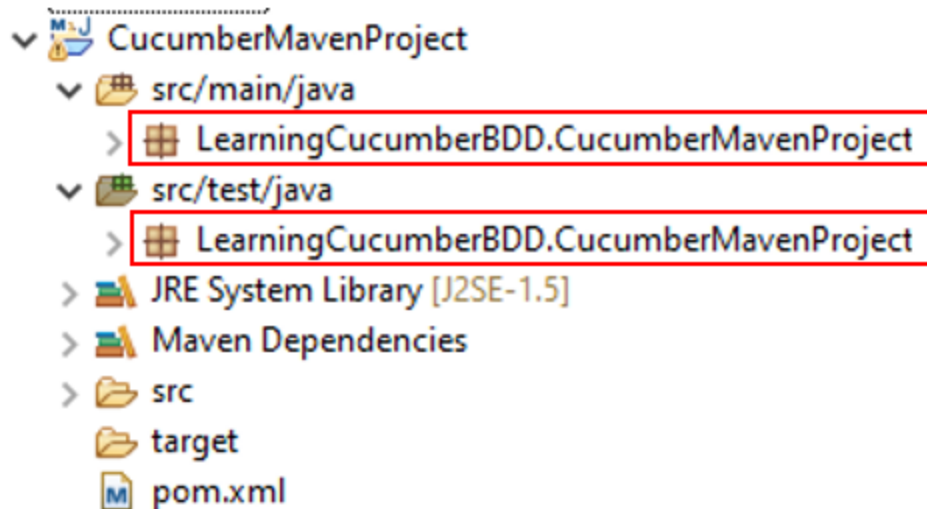
After the installation, below popup comes up



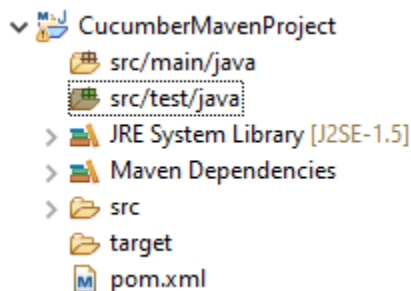
Click 'Restart Now'

Delete the dummy packages

Next, delete the 2 dummy packages that were created when maven project was created



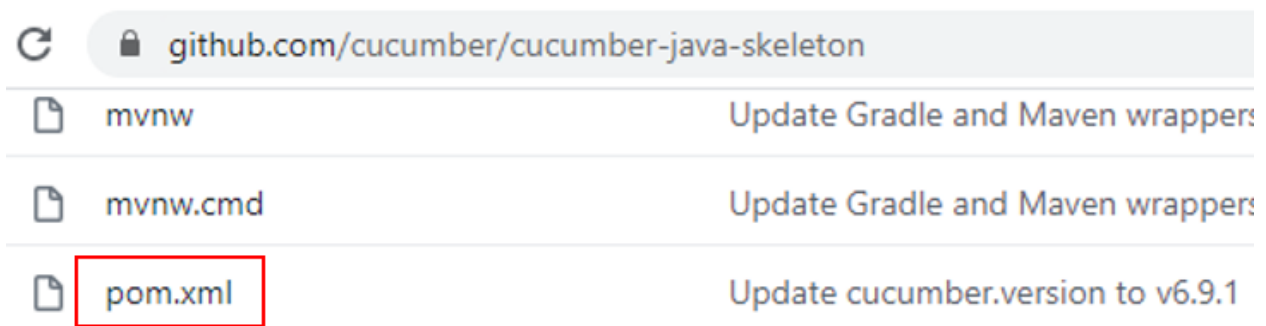
So after deleting, the project folders looks as below



Add properties to pom.xml

Go to <https://github.com/cucumber/cucumber-java-skeleton>

This java skeleton is created by cucumber team. Scroll down the page, you would see pom.xml



Open pom.xml.

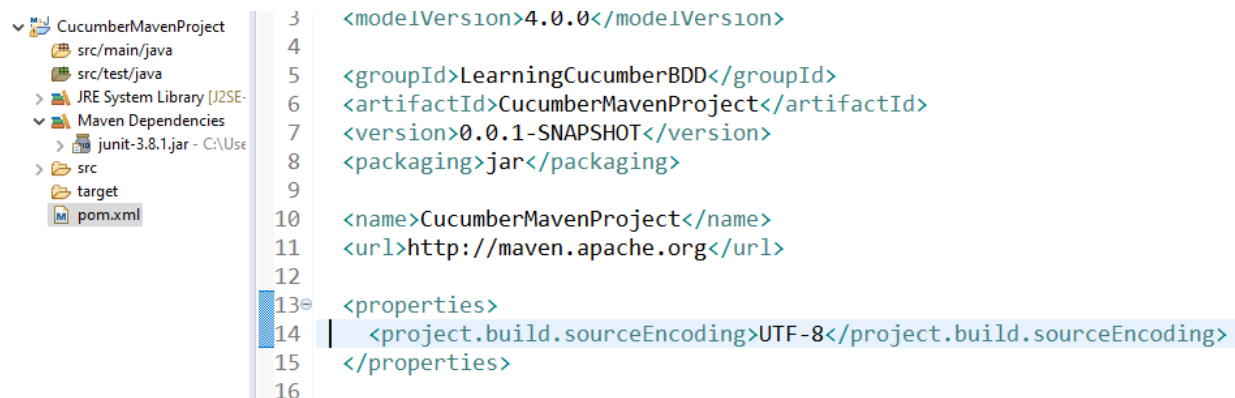
The below page would open, you would see some properties within the <properties> tag. As can be seen below, the jdk version is 1.8, junit ver is 4.13.1, cucumber version is 6.9.1

github.com/cucumber/cucumber-java-skeleton/blob/main/pom.xml

```
<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <java.version>1.8</java.version>
  <junit.version>4.13.1</junit.version>
  <cucumber.version>6.9.1</cucumber.version>
  <maven.compiler.version>3.8.1</maven.compiler.version>
  <maven.surefire.version>2.22.2</maven.surefire.version>
</properties>
```

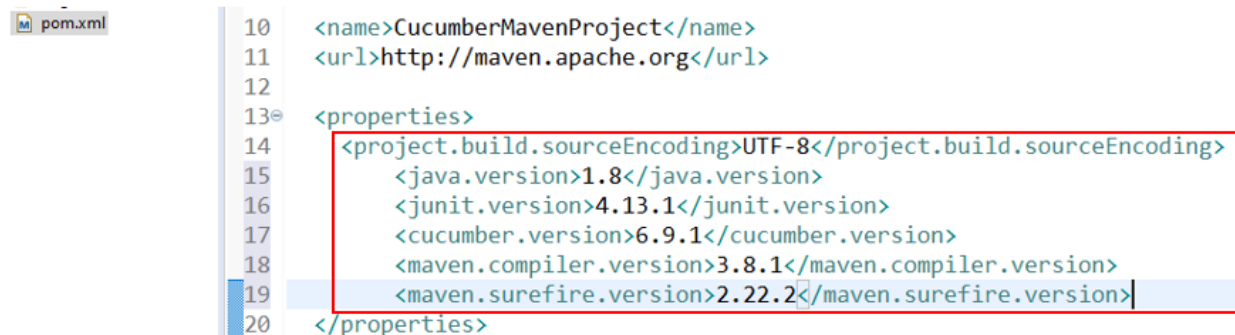
Copy all these properties

Open the pom.xml file in eclipse



```
3 <modelVersion>4.0.0</modelVersion>
4
5 <groupId>LearningCucumberBDD</groupId>
6 <artifactId>CucumberMavenProject</artifactId>
7 <version>0.0.1-SNAPSHOT</version>
8 <packaging>jar</packaging>
9
10 <name>CucumberMavenProject</name>
11 <url>http://maven.apache.org</url>
12
13 <properties>
14   <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15 </properties>
16
```

Paste the properties in pom.xml

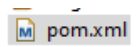


```
10 <name>CucumberMavenProject</name>
11 <url>http://maven.apache.org</url>
12
13 <properties>
14   <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15   <java.version>1.8</java.version>
16   <junit.version>4.13.1</junit.version>
17   <cucumber.version>6.9.1</cucumber.version>
18   <maven.compiler.version>3.8.1</maven.compiler.version>
19   <maven.surefire.version>2.22.2</maven.surefire.version>
20 </properties>
```

Save the xml

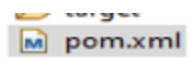
Add dependencies to pom.xml

Delete the dependency that is already present in pom.xml



```
15     <java.version>1.8</java.version>
16     <junit.version>4.13.1</junit.version>
17     <cucumber.version>6.9.1</cucumber.version>
18     <maven.compiler.version>3.8.1</maven.compiler.version>
19     <maven.surefire.version>2.22.1</maven.surefire.version>
20 </properties>
21
22 <dependencies>
23 <dependency>
24     <groupId>junit</groupId>
25     <artifactId>junit</artifactId>
26     <version>3.8.1</version>
27     <scope>test</scope>
28 </dependency>
29 </dependencies>
```

So we have



```
10 <name>CucumberMa
11 <url>http://mave
12
13 <properties>
14     <project.build
15         <java.vers
16         <junit.ver
17         <cucumber.
18         <maven.com
19         <maven.sur
20 </properties>
21
22 <dependencies>
23
24 </dependencies>
```

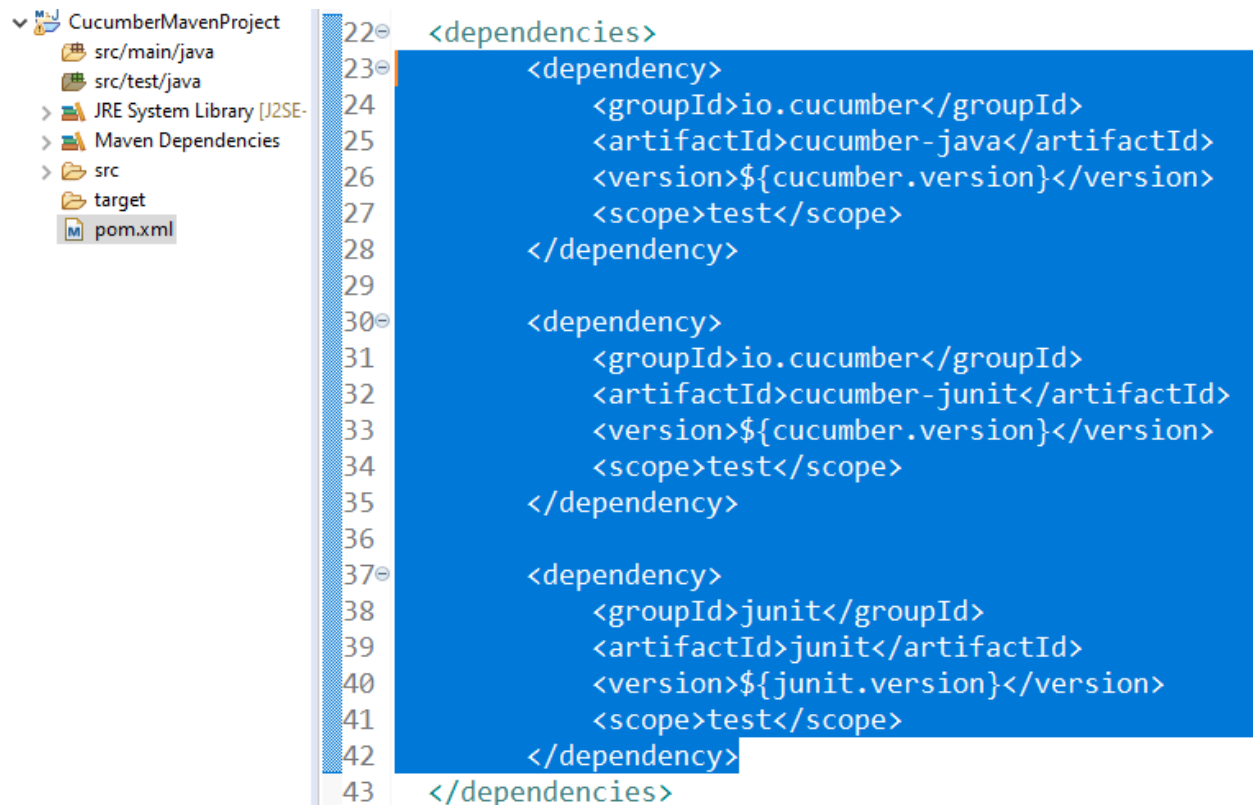
Next, go back to the skeleton, you would see 3 dependencies

```

20 <dependencies>
21   <dependency>
22     <groupId>io.cucumber</groupId>
23     <artifactId>cucumber-java</artifactId>
24     <version>${cucumber.version}</version>
25     <scope>test</scope>
26   </dependency>
27
28   <dependency>
29     <groupId>io.cucumber</groupId>
30     <artifactId>cucumber-junit</artifactId>
31     <version>${cucumber.version}</version>
32     <scope>test</scope>
33   </dependency>
34
35   <dependency>
36     <groupId>junit</groupId>
37     <artifactId>junit</artifactId>
38     <version>${junit.version}</version>
39     <scope>test</scope>
40   </dependency>
41 </dependencies>

```

Copy these and paste it in pom.xml



```

22 <dependencies>
23   <dependency>
24     <groupId>io.cucumber</groupId>
25     <artifactId>cucumber-java</artifactId>
26     <version>${cucumber.version}</version>
27     <scope>test</scope>
28   </dependency>
29
30   <dependency>
31     <groupId>io.cucumber</groupId>
32     <artifactId>cucumber-junit</artifactId>
33     <version>${cucumber.version}</version>
34     <scope>test</scope>
35   </dependency>
36
37   <dependency>
38     <groupId>junit</groupId>
39     <artifactId>junit</artifactId>
40     <version>${junit.version}</version>
41     <scope>test</scope>
42   </dependency>
43 </dependencies>

```

Save the xml

Add plugins to pom.xml

Add a blank line before `</project>` tag

```
43     </dependencies>
44
45 </project>
```

Now go to the skeleton and copy the entire `<build>` tag section, see below

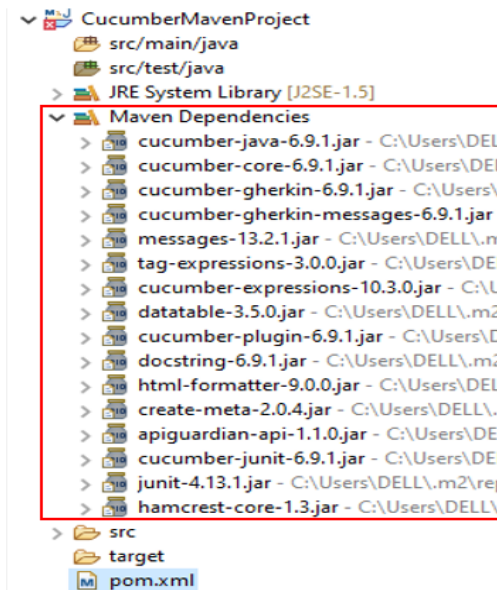
```
> C github.com/cucumber/cucumber-java-skeleton/blob/main/pom.xml
42
43     <build>
44         <plugins>
45             <plugin>
46                 <groupId>org.apache.maven.plugins</groupId>
47                 <artifactId>maven-compiler-plugin</artifactId>
48                 <version>${maven.compiler.version}</version>
49                 <configuration>
50                     <encoding>UTF-8</encoding>
51                     <source>${java.version}</source>
52                     <target>${java.version}</target>
53                 </configuration>
54             </plugin>
55             <plugin>
56                 <groupId>org.apache.maven.plugins</groupId>
57                 <artifactId>maven-surefire-plugin</artifactId>
58                 <version>${maven.surefire.version}</version>
59             </plugin>
60         </plugins>
61     </build>
62 </project>
```

Paste it at line#44 in pom.xml



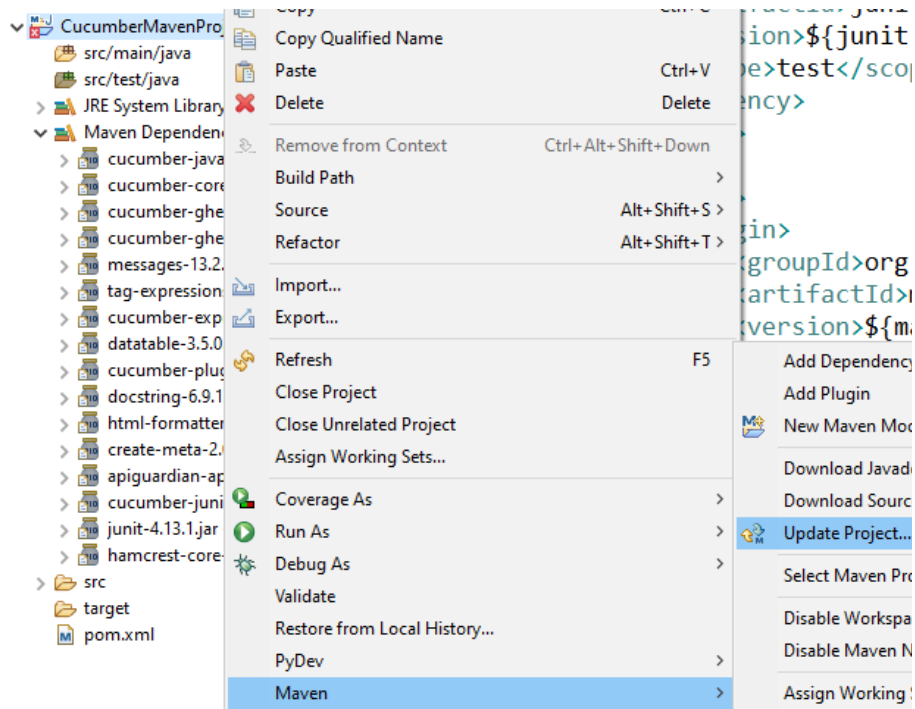
Save the xml.

Expand 'Maven Dependencies' folder, see below. You would see that the various jars get downloaded automatically



Update maven project

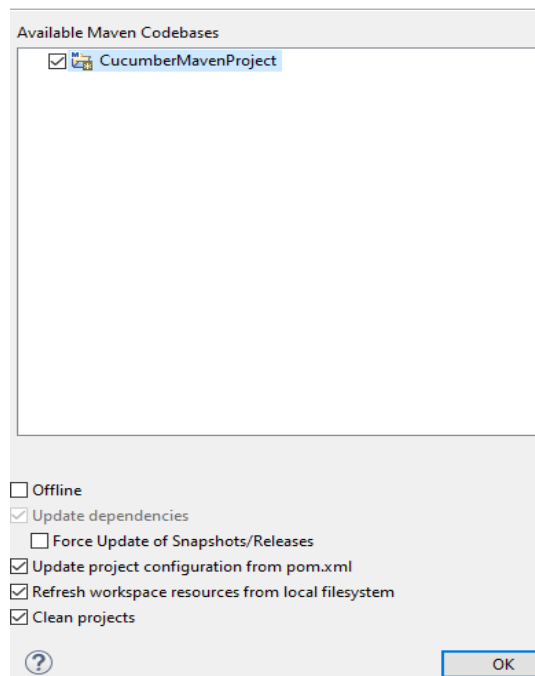
Next, right click your project > Maven > Update Project



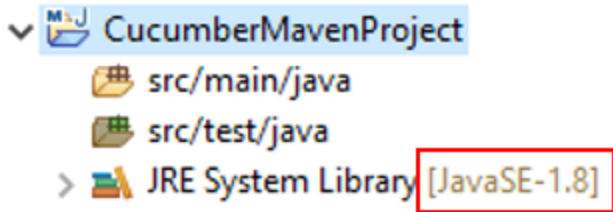
The below popup comes

Update Maven Project

Select Maven projects and update options



Click OK. The project gets updated with jdk 8, see below



Close pom.xml file in eclipse.

So this is how we setup cucumber in eclipse.

Thank you for reading!