



POWER OF CUCUMBER

EASY AUTOMATION FOR ALL

Eafa Framework by Salman Saeed



PROBLEM

Why don't we start with automation?

- Lack of Skills
- High Cost
- Time Constraint

SOLUTION TO THE PROBLEM

Cucumber can be an easy choice...

Reason?

- Easy to Setup
- Easy to Learn
- Less Coding Skills required
- One Source of Truth
- Low Maintenance Cost

CUCUMBER – STRUCTURE

What Needs to be Understood

Feature File

- Contains executable specifications written in a language called Gherkin

Keywords:

- Feature, Background, Scenario, Scenario Outline, Given, When, Then, Examples, Tags

Step Definitions

- To translate plain text Gherkin steps into actions that will interact with the system.

Runner File

- This class just need *annotations* to understand that *cucumber features* would be run through it

CUCUMBER – GHERKIN

Feature: Google Searching

As a web surfer, I want to search Google, so that I can learn new things.

Scenario Outline: Simple Google searches

Given a web browser is on the Google page

When the search phrase “panda” is entered

Then results for “panda are shown

And the related results include “Panda Express”

CUCUMBER – STEP DEFINITIONS

Step Definition

```
public class MyStepdefs {  
    @When("the search phrase \"([^\"]*)\" is entered")  
    public void the_search_phrase_entered(String phrase) {  
        driver.findElement(By.id("phrase-id")).sendKeys(phrase);  
        System.out.println("Phrase: " + phrase);  
    }  
}
```

PROBLEM WITH THE SOLUTION

- Still Coding
- Still Framework building and Maintenance Cost
- Building another framework for Mobile Testing
- Same effort again for each project

RESOLUTION TO THE PROBLEMATIC SOLUTION

Our Plug & Play Eafa-Framework – Codeless Automation

- No skills required
- Least possible output time
- Free for all
- Web and Mobile in single setup
- Can be written by BA's, Project Managers and Clients also
- Only two files to be maintained in the whole project

CODELESS AUTOMATION! REALLY?

What if you need to:

- open a different browser?
- upload a file?
- select a value from dropdown?
- take a screenshot?
- check page spellings?
- get Token or OTP from user email?
- open new or switch browser tab?
- get or post API results?
- generate a comprehensive report?

HOW?

Here is the formula:

Manual Test Cases \times Magic = Automated Test Scripts

CUCUMBER – HOW DID WE ACHIEVE

- Wrote simple actions in feature files with tags
e.g. user clicks on “Submit” button
- Wrote dynamic step definitions which takes values from feature files
- Created dynamic methods for almost all technologies to perform actions such as:
‘Click, Input, File upload, Open browser, Spellcheck, Screenshot etc.

CUCUMBER – WHAT DO I NEED TO SETUP?

1. Java Environment Setup
2. Maven
3. Eclipse
4. Appium
5. Download our Magical Code

REPORTS

Features Statistics

The following graphs show passing and failing statistics for features



Feature	Scenarios			Steps							Duration	Status
	Total	Passed	Failed	Total	Passed	Failed	Skipped	Pending	Undefined	Missing		
Login	1	1	0	4	4	0	0	0	0	0	51s 138ms	Passed
Registration Form	2	2	0	6	6	0	0	0	0	0	58s 516ms	Passed
2	3	3	0	10	10	0	0	0	0	0	1m 49s 654ms	Totals

Feature Report

Feature	Steps						Scenarios			Features	
	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
demo/callfeature/call-feature.feature	27	0	0	0	0	27	1	0	1	1s 366ms	Passed



Feature demo/callfeature/call-feature.feature

calling another feature file

Background

654ms

Steps

* url demoBaseUrl

654ms

Scenario create kittens and then create parent cat

712ms

Steps

* call create-two-cats.feature

000ms

* url demoBaseUrl

000ms

Given path 'cats'

000ms

And request { name: 'Bob' }

000ms

When method post

000ms

Doc string

```
08:10:05.182 DEBUG -
1 > POST http://127.0.0.1:58317/cats
1 > Accept-Encoding: gzip, deflate
```

CUCUMBER – TIME FOR HANDS ON!

The whole process from configuration till you execute your first automation test case is of maximum 30 minutes and it'll take even less time if you have a little setup knowledge.

Downloading time is not included 😊

Let's now start with some hands on !

SETUP GUIDE – STEP 1 (CONT.)

DOWNLOAD AND CONFIGURE JDK

JDK Download and setup help links:

- <https://www.wikihow.com/Install-the-Java-Software-Development-Kit>
- <https://www.youtube.com/watch?v=Wp6uS7CmivE>

SETUP GUIDE – STEP 1

DOWNLOAD AND CONFIGURE JDK

- Download JDK from below link:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk10-downloads-4416644.html>

SETUP GUIDE – STEP 2

DOWNLOAD AND CONFIGURE MAVEN

Open link and follow the instruction to download and setup Maven:

- <https://www.javatpoint.com/how-to-install-maven>

SETUP GUIDE – STEP 3

VERIFY JDK AND MAVEN ARE INSTALLED

Before you download Eclipse please verify that JDK and Maven are installed and configured.

Open Command prompt with administrative rights

cmd > right click and Run as Administrator

Type “java --version”

Without quotes to verify jdk is installed

Then type “mvn --version”

To verify that Maven is installed and configured

Google it for more details... but its important

SETUP GUIDE – STEP 3

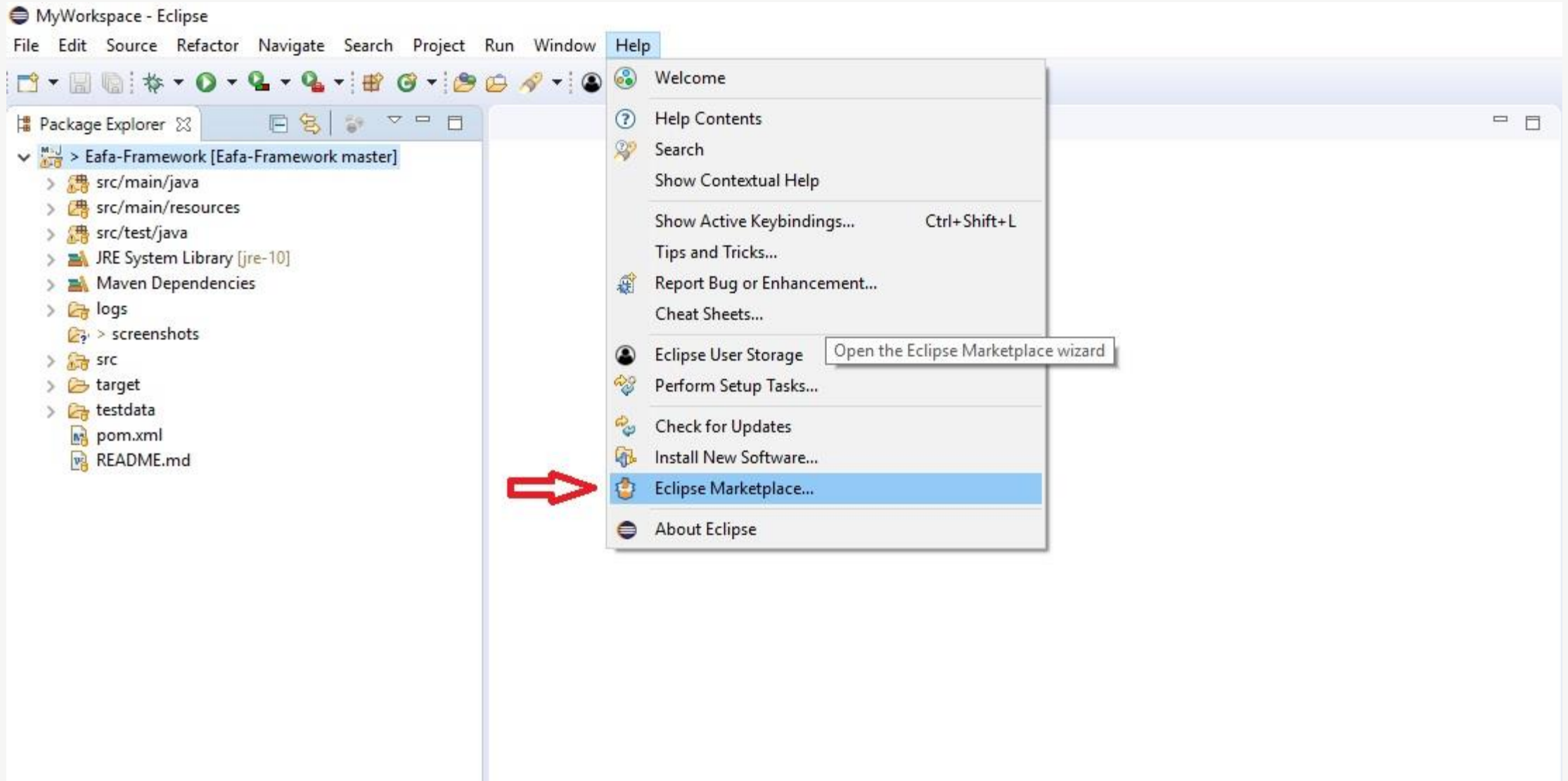
DOWNLOAD AND INSTALL ECLIPSE OXYGEN

Download Eclipse any latest version and set a workspace:

- <https://www.eclipse.org/downloads/>

SETUP GUIDE – STEP 4

INSTALL CUCUMBER PLUGIN IN ECLIPSE



SETUP GUIDE – STEP 4 (CONT.)

INSTALL CUCUMBER PLUGIN IN ECLIPSE

The screenshot shows the Eclipse IDE interface with the Eclipse Marketplace window open. The Package Explorer on the left shows a project named 'Eafa-Framework' with a file tree including 'src/main/java', 'src/main/resources', 'src/test/java', 'JRE System Library [jre-10]', 'Maven Dependencies', 'logs', 'screenshots', 'src', 'target', 'testdata', 'pom.xml', and 'README.md'. The Eclipse Marketplace window displays search results for 'cucumber'. The 'Cucumber Eclipse Plugin 0.0.21.201802160922' is highlighted with a red box. Below it, the 'Natural 0.7.6' plugin is visible. The Marketplace window also shows a 'Marketplaces' section with icons for various marketplaces. At the bottom of the Marketplace window, there are buttons for '< Back', 'Install Now >', 'Finish', and 'Cancel'.

MyWorkspace - Eclipse
File Edit Source Refactor Navigate Search Project Run Window

Package Explorer
Eafa-Framework [Eafa-Framework master]
src/main/java
src/main/resources
src/test/java
JRE System Library [jre-10]
Maven Dependencies
logs
screenshots
src
target
testdata
pom.xml
README.md

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

Search Recent Popular Favorites Installed Eclipse Newsletter: Boot Build E
Find: :cucumber All Markets All Categories Go

Cucumber Eclipse Plugin 0.0.21.201802160922
An eclipse plugin for Cucumber. New Feature list in the version available from the update site: Lambda Expression support for Cucumber-Java8 Content... [more info](#)
by Cucumber, MIT
[cucumber Eclipse Plugin](#) [eclipse-plugin JVM](#) [eclipse plugins](#)
★ 22 Installs: 14.1K (3,382 last month) **Installed**

Natural 0.7.6
Natural is a set of plugins providing a smooth experience in editing and maintaining BDD/ATDD files. The currently supported languages are Cucumber (Gherkin... [more info](#)
by Unknown, EPL
[cucumber Gherkin JBehave ATDD BDD ...](#)
★ 25 Installs: 65.6K (3,832 last month) **Install**

Marketplaces

< Back Install Now > Finish Cancel

SETUP GUIDE – STEP 5

DOWNLOAD EAFA-FRAMEWORK

Link to download Eafa-Framework

- <https://github.com/salmansaeed321/eafa>

SETUP GUIDE – STEP 5 (CONT.)

Click on 'Clone or Download' and select Download as Zip

The screenshot shows a web browser window displaying a GitHub repository page for 'eafa' by user 'salmansaeed321'. The browser's address bar shows the URL 'https://github.com/salmansaeed321/eafa'. The page features a 'Join GitHub today' banner at the top, followed by a description of the 'Eafa framework'. Below the description, repository statistics are shown: 2 commits, 1 branch, 0 releases, and 1 contributor. A 'Clone or download' button is highlighted in green, and its dropdown menu is open, showing options for cloning with HTTPS and downloading as a ZIP file. The 'Download ZIP' option is selected and highlighted in blue. The repository file list includes folders like '.settings', 'logs', 'src', 'testdata' and files like '.classpath' and '.gitignore', all marked as 'Initial commit'.

Join GitHub today
GitHub is home to over 20 million developers working together to host and review code, manage projects, and build software together.
Sign up

Eafa framework (Easy automation for all) - Power of Cucumber: created by using Cucumber, Selenium and Java mainly to help testing community so that they can automate any application without professional coding skills.

2 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Find file Clone or download

Clone with HTTPS
Use Git or checkout with SVN using the web URL.
`https://github.com/salmansaeed321/eafa`

Open in Desktop Download ZIP

File/Folder	Commit	Time
Initial commit	Initial commit	
.settings	Initial commit	
logs	Initial commit	
src	Initial commit	
testdata	Initial commit	
.classpath	Initial commit	30 minutes ago
.gitignore	Initial commit	30 minutes ago

<https://github.com/salmansaeed321/eafa/archive/master.zip>

SETUP GUIDE – STEP 6

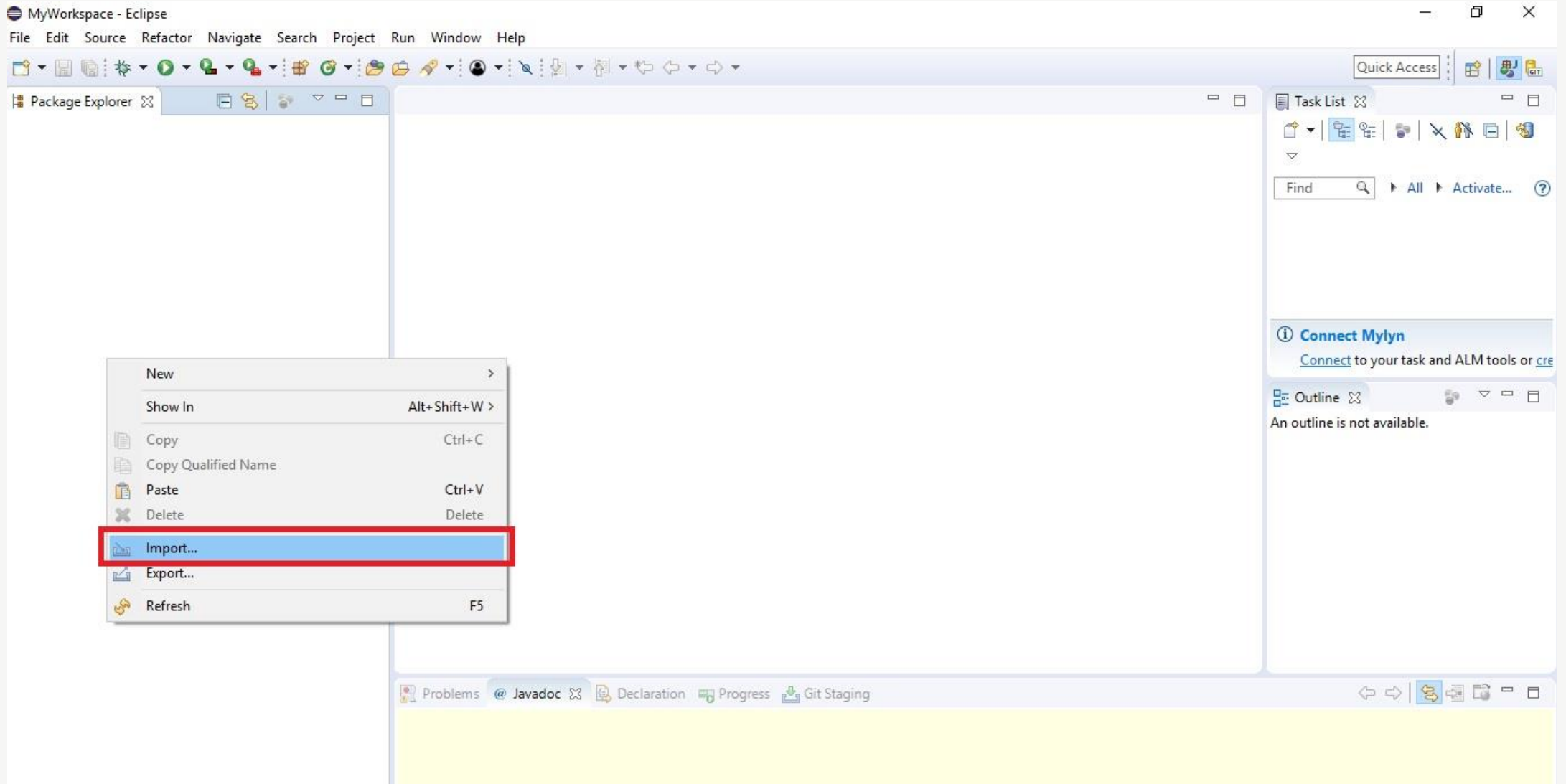
Extract downloaded folder

SETUP GUIDE – STEP 7

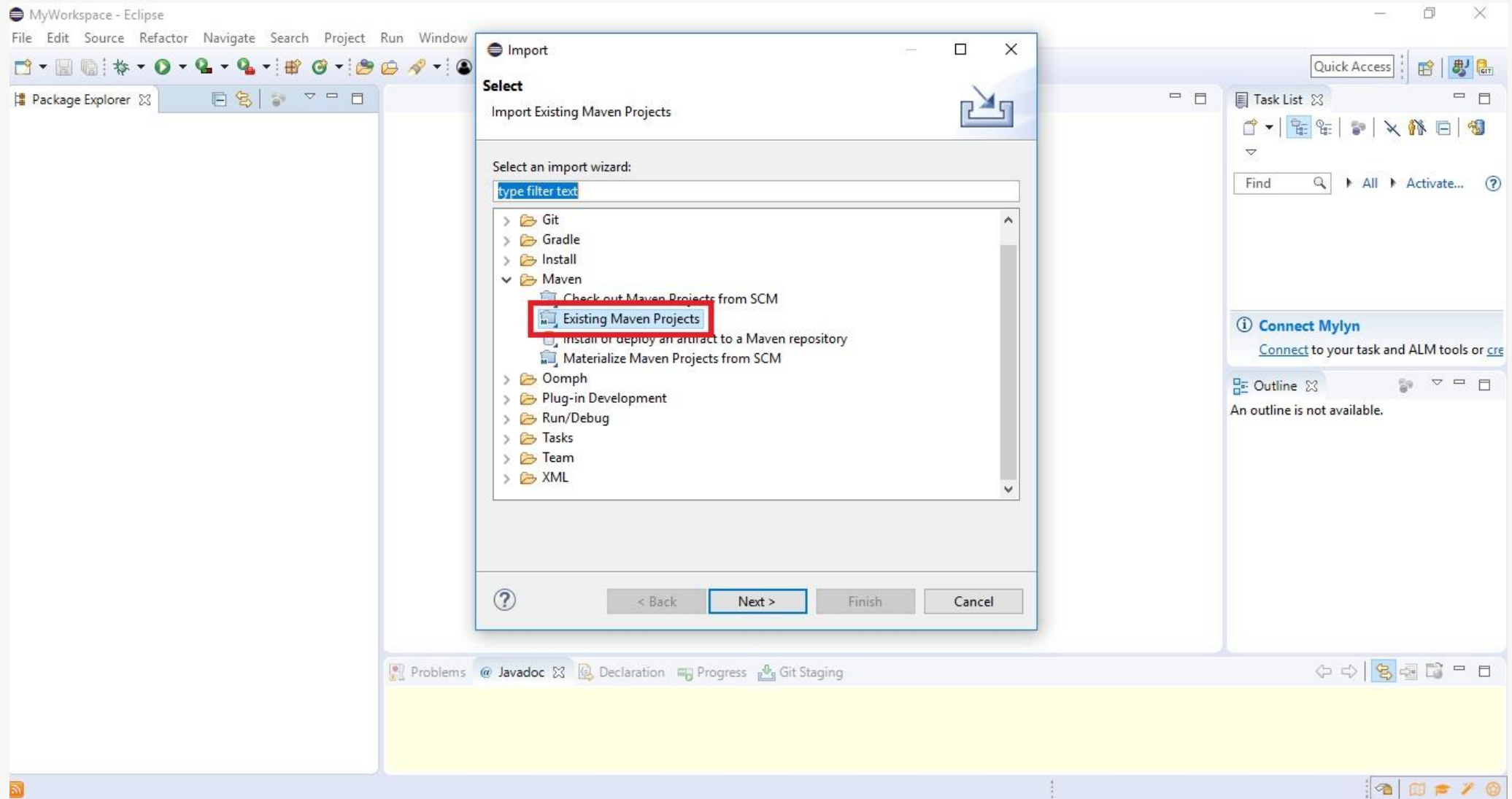
Import project in Eclipse

- i. Open Eclipse
- ii. Right click in project explorer area
- iii. Select Import
- iv. Select Maven
- v. Select Existing Maven Project
- vi. Select Downloaded Project folder
- vii. Click Finish

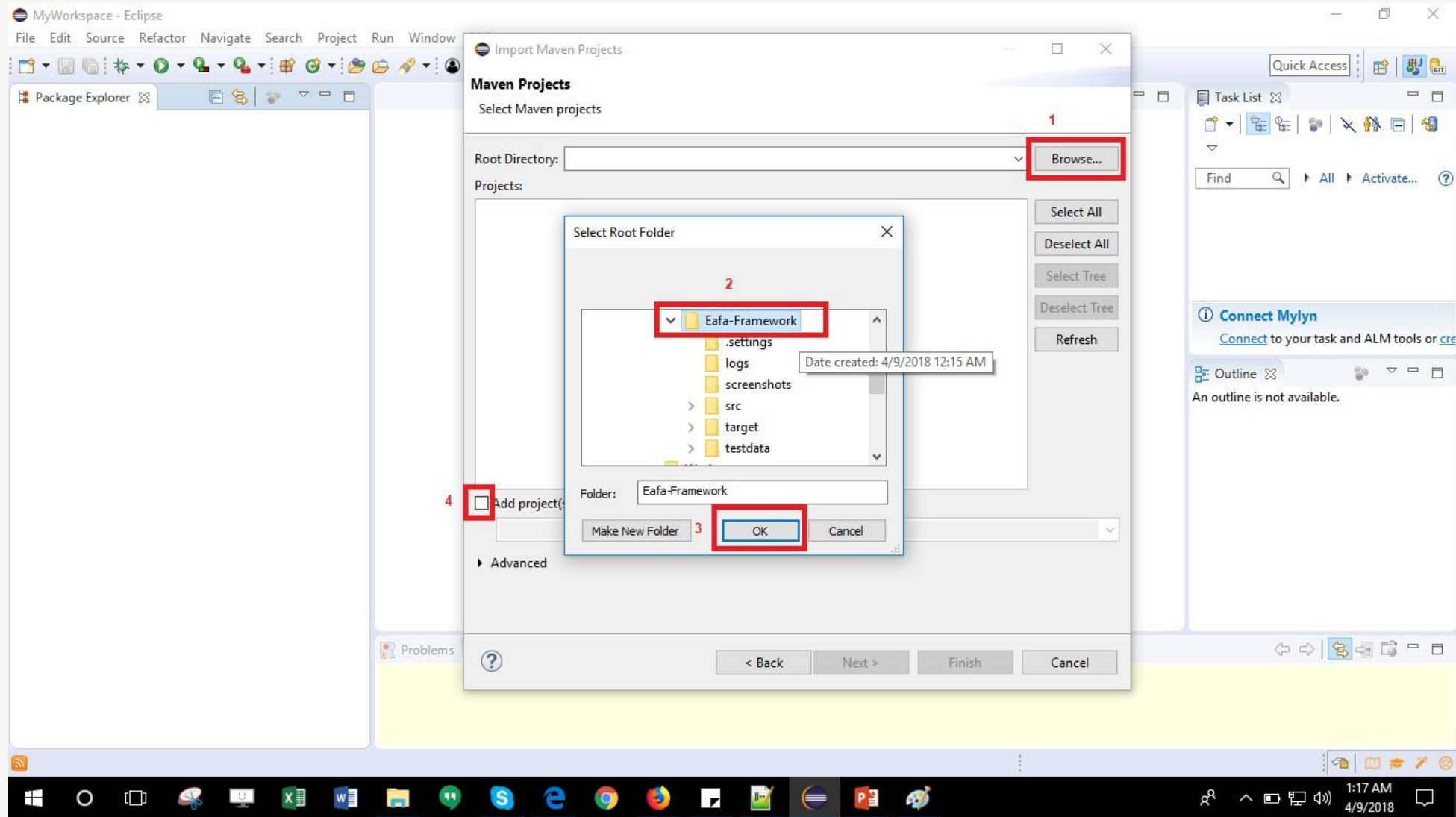
SETUP GUIDE – STEP 7 (CONT.)



SETUP GUIDE – STEP 7 (CONT.)

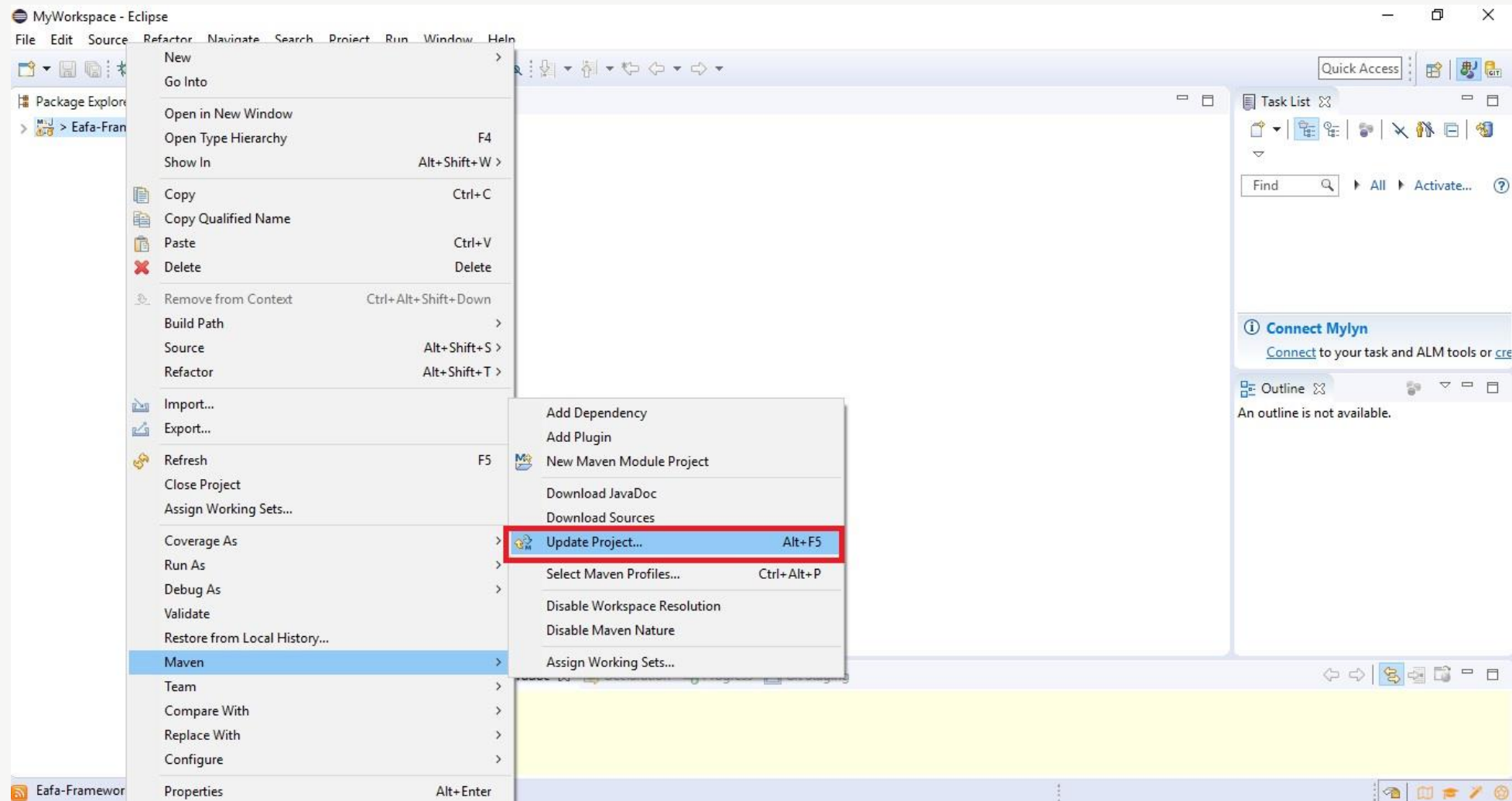


SETUP GUIDE – STEP 7 (CONT.)



SETUP GUIDE – STEP 8

Right click on the project, select Maven and then 'Update Project' it will take a while when doing for the first time



SETUP GUIDE – STEP 9

Open 'CodeRunner' file, right click and select 'Run As' Junit Test. It should run your Google sampe test in Chrome browser

The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows the project structure, with 'CodeRunner.java' selected under 'src/test/java/com.project.userflow'. The main editor shows the code for 'CodeRunner.java', which includes package declarations, imports, annotations, and a public class definition. A right-click context menu is open over the class, with the 'Run As' option expanded to show 'JUnit Test' selected. The 'Run As' option is highlighted in blue, and the 'JUnit Test' option is also highlighted in blue. The 'Run As' option is also highlighted in blue. The 'Run As' option is also highlighted in blue.

```
1 package com.project.userflow;
2
3 import org.junit.runner.RunWith;
4
5
6
7
8 /**
9  * @author Salman Saeed
10  * @email salmansaeed321@hotmail.com
11  * @version 3.1
12  */
13
14 @RunWith(Cucumber.class)
15 @CucumberOptions(
16     // To generate report in .json and .html formats
17     // .json format is also mandatory for cucumber main
18     plugin = {"json:target/cucumber.json", "html:target/cucumber.html"},
19
20     // To define feature files folder
21     features = "src/main/resources/features",
22
23     // We can comment/uncomment tags as per requirement
24     tags = {"@UploadDocNegativeFlows,@uopladDoc1"},
25     tags = {"@Google"}
26 )
27 public class CodeRunner {
28
29 }
```

The context menu options include: Undo (Ctrl+Z), Revert File, Save (Ctrl+S), Open Declaration (F3), Open Type Hierarchy (F4), Open Call Hierarchy (Ctrl+Alt+H), Show in Breadcrumb (Alt+Shift+B), Quick Outline (Ctrl+O), Quick Type Hierarchy (Ctrl+T), Open With, Show In (Alt+Shift+W), Cut (Ctrl+X), Copy (Ctrl+C), Copy Qualified Name, Paste (Ctrl+V), Quick Fix (Ctrl+1), Source (Alt+Shift+S), Refactor (Alt+Shift+T), Local History, References, Declarations, Add to Snippets..., Coverage As, Run As (highlighted), Debug As, Validate, Create Snippet..., Team, Compare With, Replace With, and Preferences... The 'Run As' option is highlighted in blue, and the 'JUnit Test' option is also highlighted in blue.

SETUP GUIDE

You will need to maintain mainly two files only but for setup please note total 4 files you will need for your project:

I. Feature file

- i. You can use existing feature files and can create as many new feature files as required.
Each feature per file with multiple scenarios.
- ii. Please do open and read existing feature files for understanding
- iii. While creating a new file, must add file extension as .feature e.g. Login.feature
- iv. All feature files must be placed under 'feature folder
- v. Under Feature folder there is a ReadMe.txt file which has all possible statements according to the situation. You will not be able to use any statement other than those which are mentioned in this file. Or if you wish to and have knowledge, you can add new step definitions and methods and play with however you like to 😊

SETUP GUIDE

2. OR.properties

- i. This file contains all the id's, xpath's, linktext etc.
- ii. In order to call any element we need a locator
- iii. To get locator of any element > Open any url, right click and select "Inspect Element" and copy xpath or id. You can also Google it for more details.
- iv. Paste found locators of buttons, input fields, etc. in Or.properties file with unique reference name which will be used later in feature file to call these elements for any action.
- v. Element reference name must have a suffix of its type e.g. xpath, id, linktext etc. "login-button-xpath" if you have xpath or "login-button-id" if you have id of an element.

SETUP GUIDE

3. Config.properties

- i. This file contains all project configurations
- ii. Host url on which you wish to perform testing
- iii. Page load confirmation xpath which confirms that the page has been loaded
- iv. Mobile device and apk configurations

SETUP GUIDE

4. CodeRunner

- i. We use coderunner file to execute test cases
- ii. Right click on coderunner file and select Run As > Junit and it will execute your test
- iii. You can also tag any scenario or feature and mention that tag in coderunner in tag section so that only that scenario/feature will be executed. As in example it only executes Google feature.
- iv. It also takes care of test reports after execution.

SETUP GUIDE

Optional downloads for mobile application testing. You will need actual mobile device attached or simulator configured.

<http://appium.io/downloads.html>

<https://experitest.com/mobile-test-automation/appium-studio/>

Questions?

Salman Saeed

Email: salmansaeed321@hotmail.com

Skype: [salmansaeed321](https://www.skype.com/people/salmansaeed321)