Cucumber\_Java\_Install

Installed before (IntelliJ IDEA; Java; TestNG; Appium; Apache POI; MySQL-connector):

<**dependency**>

<**groupId**>org.testng</**groupId**>

<**artifactId**>testng</**artifactId**>

<**version**>6.10</**version**>

<**scope**>test</**scope**>

</**dependency**>

<**dependency**>

<**groupId**>io.appium</**groupId**>

<**artifactId**>java-client</**artifactId**>

<**version**>7.1.0</**version**>

</**dependency**>

<**dependency**>

<**groupId**>org.apache.poi</**groupId**>

<**artifactId**>poi-ooxml</**artifactId**>

<**version**>4.0.1</**version**>

</**dependency**>

<**dependency**>

<**groupId**>mysql</**groupId**>

<**artifactId**>mysql-connector-java</**artifactId**>

<**version**>8.0.17</**version**>

<**exclusions**>

<**exclusion**>

<**groupId**>com.google.protobuf</**groupId**>

<**artifactId**>protobuf-java</**artifactId**>

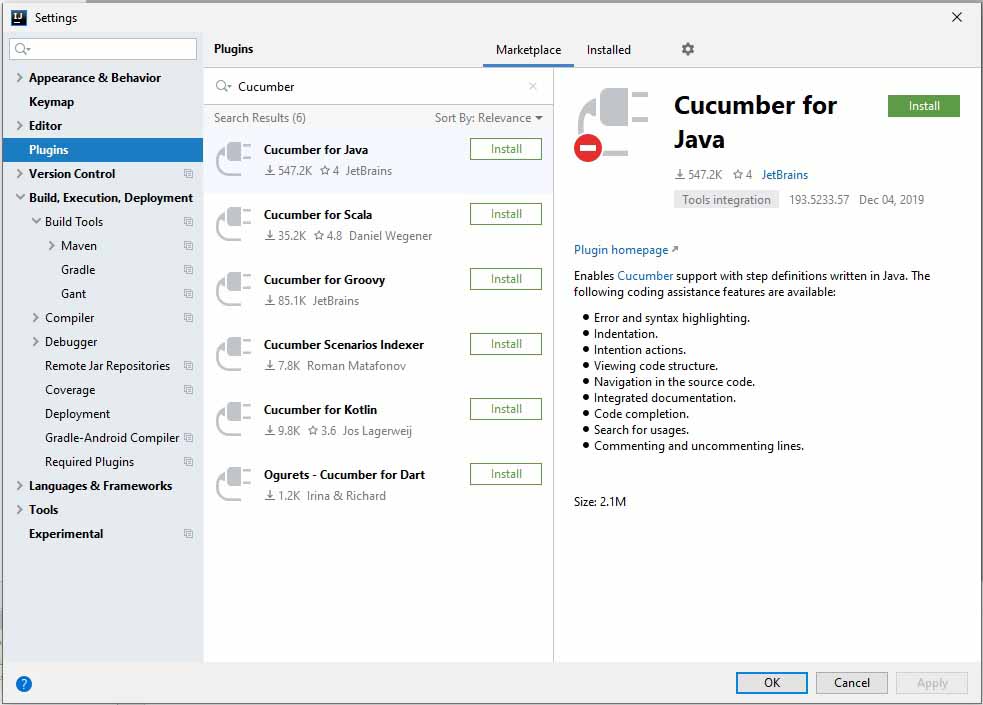
</**exclusion**>

</**exclusions**>

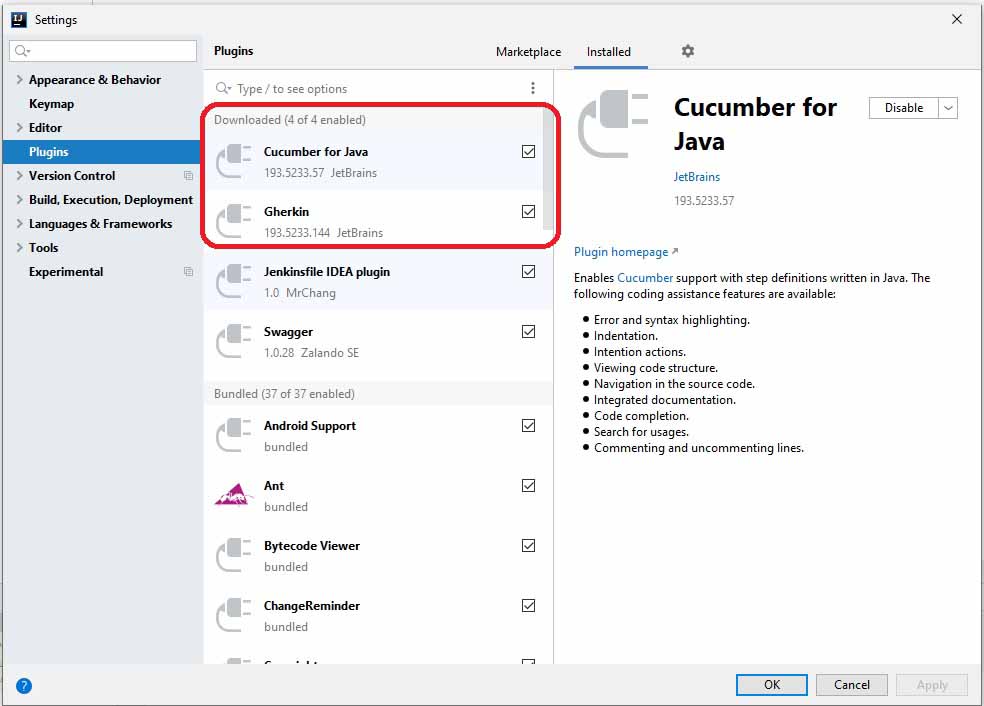
</**dependency**>

Install Cucumber & Gherkin Plugins

Open IntelliJ. **Go to: File -> Settings** and click on **Plugins.** Select Marketplace and search for **Cucumber**.



Install **Cucumber for Java** and restart IntelliJ.



Open the POM.xml file and add dependencies for cucumber:

<https://mvnrepository.com/artifact/io.cucumber/cucumber-java8/4.8.0>

<dependency>

<groupId>io.cucumber</groupId>

<artifactId>cucumber-java8</artifactId>

<version>4.8.0</version>

</dependency>

<https://mvnrepository.com/artifact/io.cucumber/cucumber-testng>

<dependency>

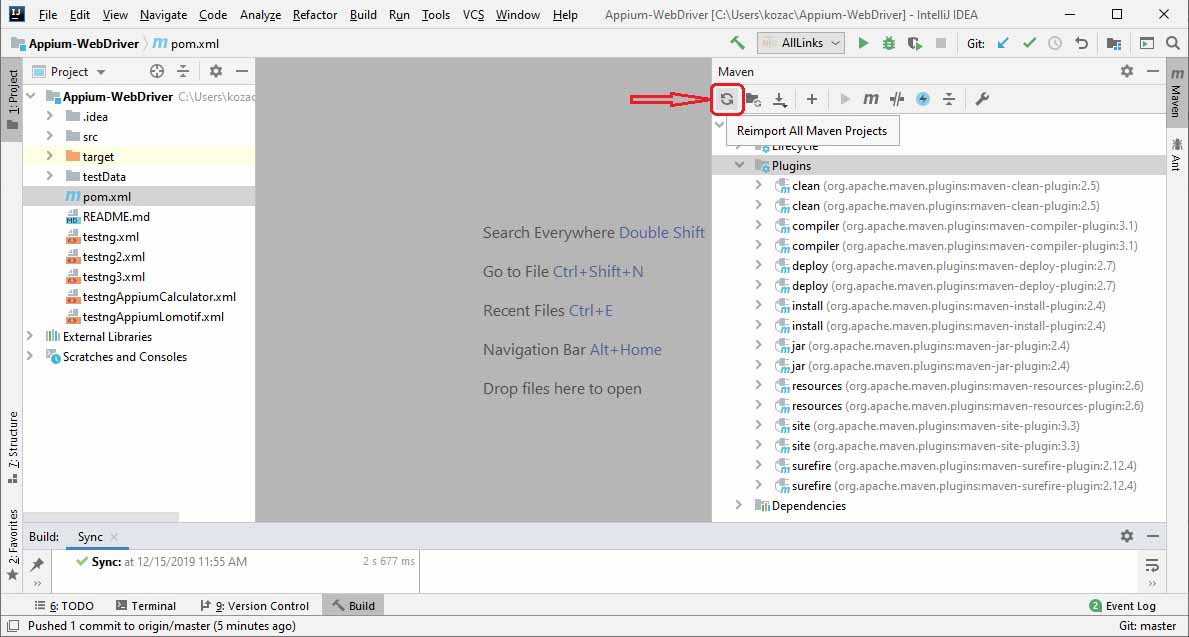
<groupId>io.cucumber</groupId>

<artifactId>cucumber-testng</artifactId>

<version>4.8.0</version>

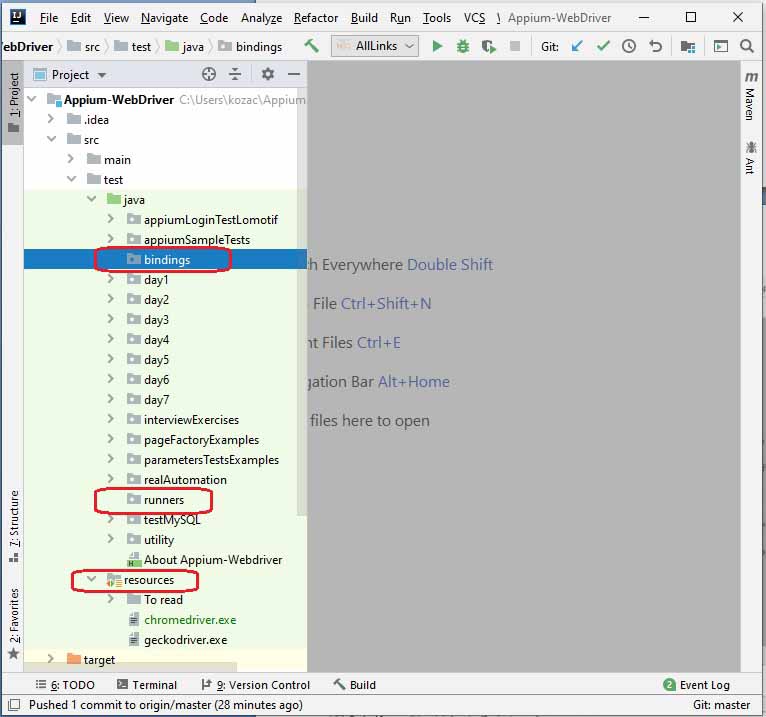
</dependency>

On the right-hand side of the IntelliJ IDE, click **Maven Projects**tab to show Maven lifecycle. Click **re-import Maven Projects** button to update from the POM.



Right-click on the **java** folder under **test**, and add 2 new packages: **"bindings" & "runners".**

**Also, we have directory "**resources**"** under the "test" folder.



Feature File

Create new directoryshugaringFactory **in the folder** resources**.**

Create new filebasket.feature **in the folder** shugaringFactory**.**

Write your feature, example:

@run

**Feature**: basket

*#shopping basket features*

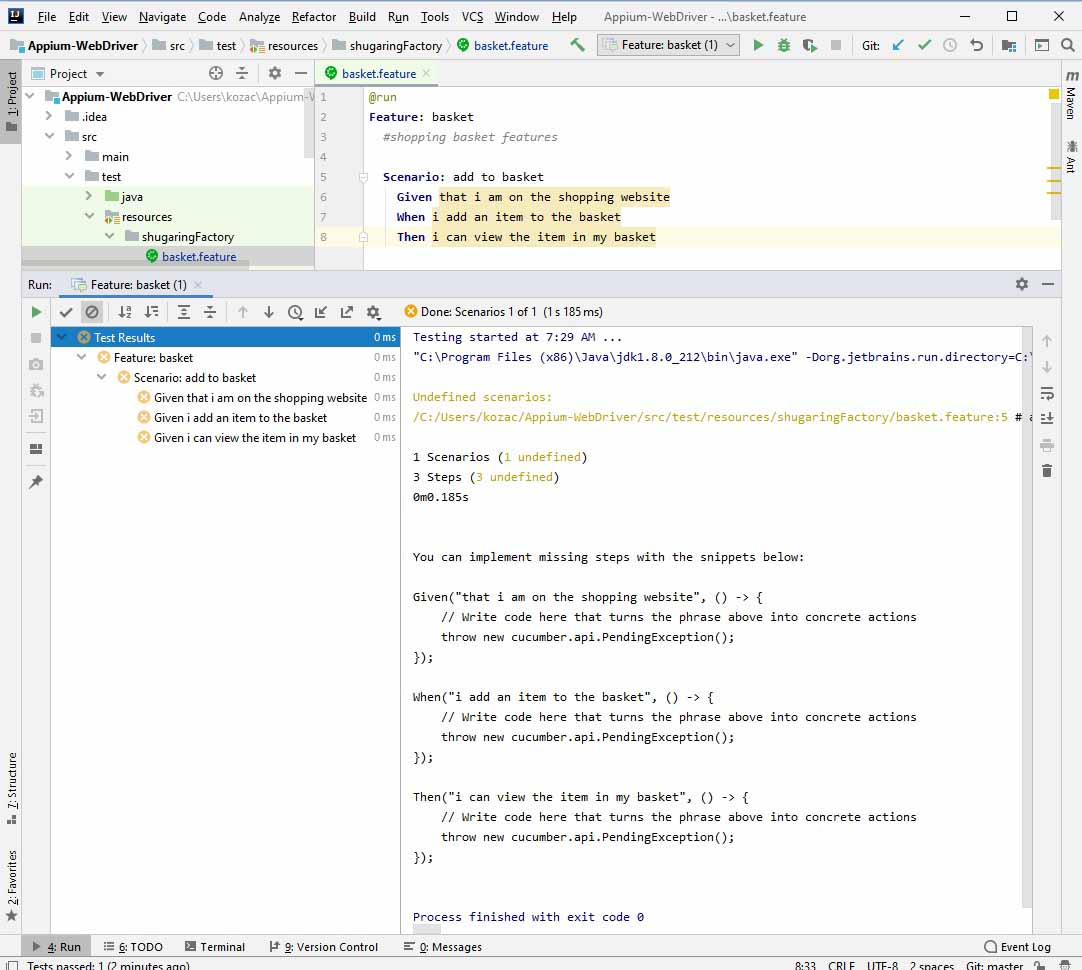
**Scenario**: add to basket

**Given** that i am on the shopping website

**When** i add an item to the basket

**Then** i can view the item in my basket

**Run** basket.feature file. This Feature file has no code bindings.



It will suggest some pattern code bindings for you, in the Run pane.

Select these and copy to the clipboard.

Given("that i am on the shopping website", () -> {

// Write code here that turns the phrase above into concrete actions

throw new cucumber.api.PendingException();

});

When("i add an item to the basket", () -> {

// Write code here that turns the phrase above into concrete actions

throw new cucumber.api.PendingException();

});

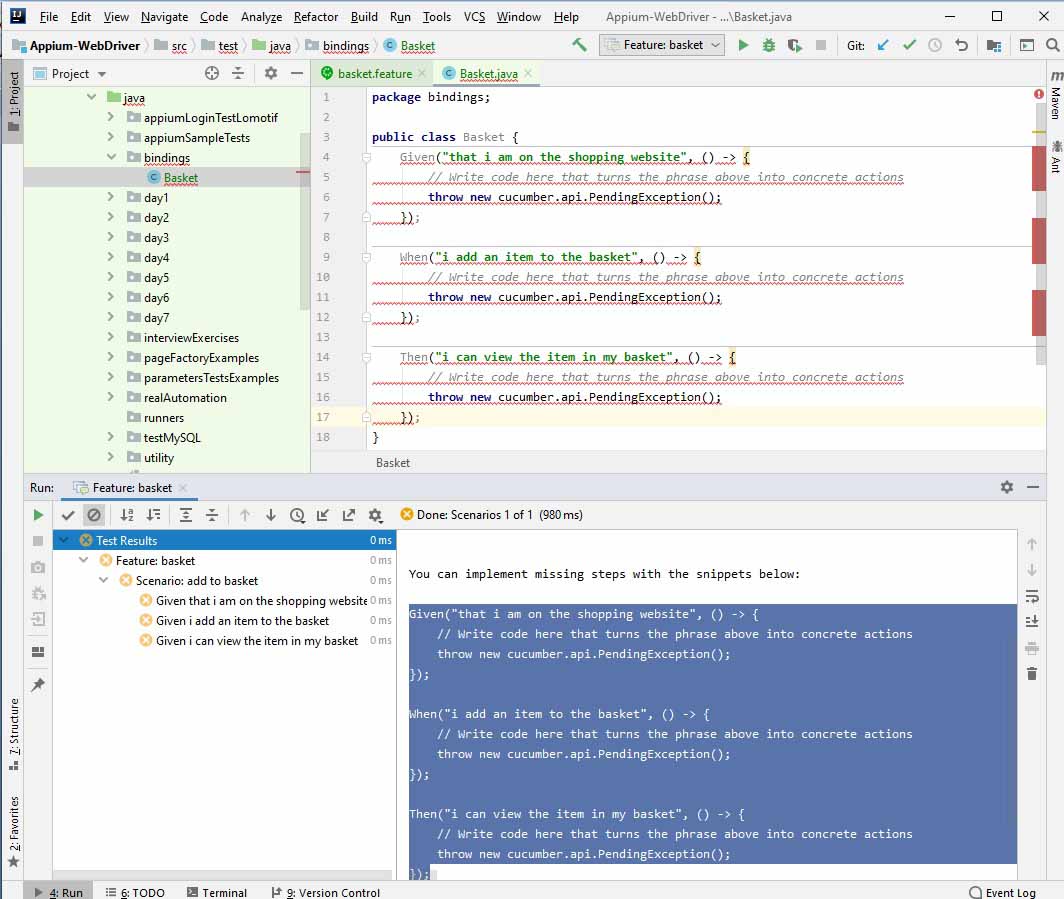
Then("i can view the item in my basket", () -> {

// Write code here that turns the phrase above into concrete actions

throw new cucumber.api.PendingException();

});

Create new Java Class Basket.java in the package bindings. Paste in your pattern code.



Put @ - annotations to Given; When; Then. Click in a line of code error and use Alt-Enter to fix imports.

**package** bindings;

**import** io.cucumber.java.en.Given;

**import** io.cucumber.java.en.Then;

**import** io.cucumber.java.en.When;

**import** org.openqa.selenium.\*;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.interactions.Action;

**import** org.openqa.selenium.interactions.Actions;

**import** org.openqa.selenium.support.ui.ExpectedConditions;

**import** org.openqa.selenium.support.ui.WebDriverWait;

**import** java.util.concurrent.TimeUnit;

**import static** org.testng.Assert.*assertEquals*;

**import static** org.testng.Assert.*assertTrue*;

**public class** Basket {

WebDriver **driver**;

String **quantity** = **"20"**;

String **actualQuantity**;

@Given(**"^that i am on the shopping website$"**)

**public void** that\_i\_am\_on\_the\_shopping\_website() **throws** Throwable {

System.*setProperty*(**"webdriver.chrome.driver"**, System.*getProperty*(**"user.dir"**) +

**"\\src\\test\\resources\\chromedriver.exe"**);

**driver** = **new** ChromeDriver();

**driver**.manage().timeouts().implicitlyWait(15, TimeUnit.***SECONDS***);

**driver**.get(**"https://test.sugaringfactory.com/"**);

**driver**.manage().window().maximize();

}

@When(**"^i add an item to the basket$"**)

**public void** i\_add\_an\_item\_to\_the\_basket() **throws** Throwable {

**driver**.findElement(By.*xpath*(**"//div[contains(text(),'Soft')]"**)).click();

**driver**.findElement(By.*xpath*(**"//input[@name='quantity']"**)).clear();

**driver**.findElement(By.*xpath*(**"//input[@name='quantity']"**)).sendKeys(**quantity**);

**driver**.findElement(By.*xpath*(**"//input[@id='button-cart']"**)).click();

**driver**.manage().timeouts().implicitlyWait(15, TimeUnit.***SECONDS***);

Thread.*sleep*(3000);

**driver**.findElement(By.*xpath*(**"//div[@class='success']//i[@class='icon-remove-sign']"**)).click();

**driver**.manage().timeouts().implicitlyWait(15, TimeUnit.***SECONDS***);

}

@Then(**"^i can view the item in my basket$"**)

**public void** i\_can\_view\_the\_item\_in\_my\_basket() **throws** Throwable {

Actions builder = **new** Actions(**driver**);

Action mouseOverHome = builder

.moveToElement(**driver**.findElement(By.*xpath*(**"//div[@id='cart']"**)))

.build();

**for**(**int** r=0; r<10; r++) {

**try** {

**driver**.manage().timeouts().implicitlyWait(15, TimeUnit.***SECONDS***);

mouseOverHome.perform();

**driver**.manage().timeouts().implicitlyWait(15, TimeUnit.***SECONDS***);

Thread.*sleep*(3000);

**actualQuantity** = **driver**.findElement(By.*xpath*(**"//span[@class='quantity']"**)).getText();

System.***out***.println(**"Added to cart: "** + **actualQuantity**);

*assertEquals*(**actualQuantity**,**"X "** + **quantity**);

**break**;

} **catch**(StaleElementReferenceException e) {

e.toString();

System.***out***.println(**"Trying to recover from a stale element :"** + e.getMessage());

}

}

Thread.*sleep*(3000);

**driver**.quit();

}

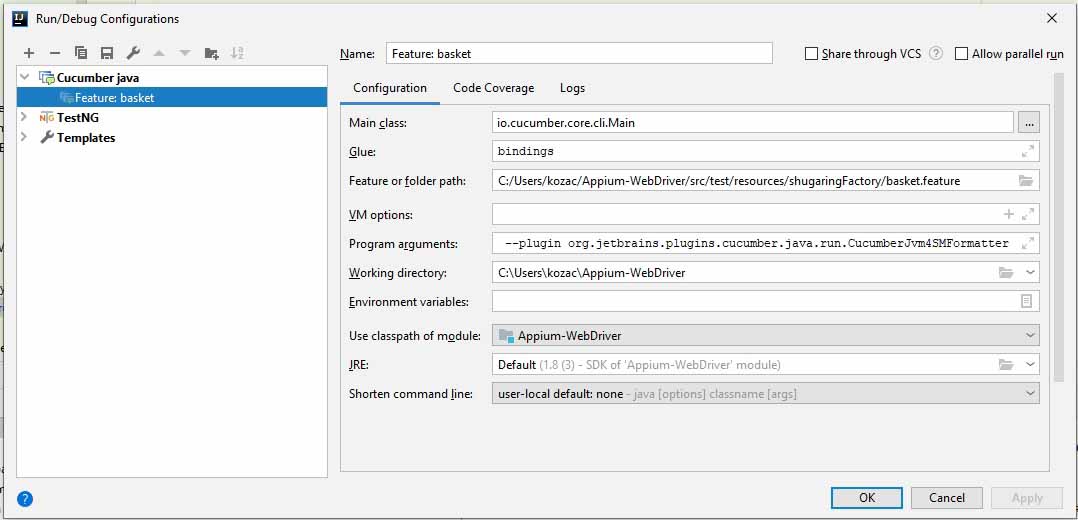
}

Update chromedriver, if it is necessary: <https://sites.google.com/a/chromium.org/chromedriver/downloads>

Before we can execute feature files, we need to tell IntelliJ where the feature files & code are bindings.

Open basket.feature file. In IntelliJ menu, Run->Edit Configurations.

Click on the feature name on the left (Feature: basket) and − provide the **Glue** (package that the code bindings will reside) (bindings) and − the **Feature or Folder Path** (where are Features files reside) (C:/Users/kozac/Appium-WebDriver/src/test/resources/shugaringFactory/basket.feature).



OK. Now we can execute the Feature file. It should open a Web Browser and perform the test.

