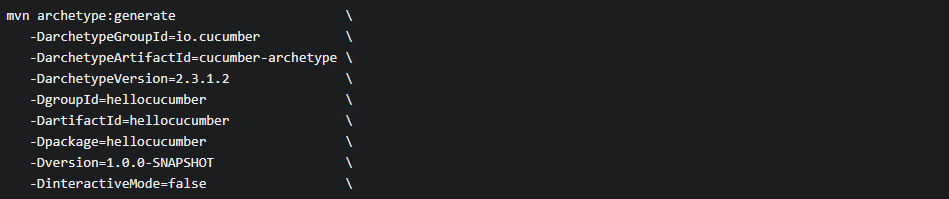
Lesson 5 Lesson End Project: Cucumber with Kotlin

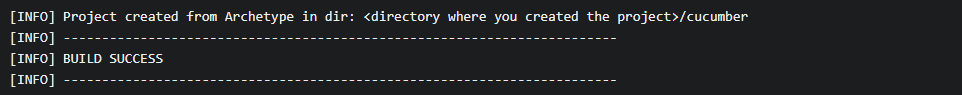
This section will guide you to:

* Use cucumber and write the scenario with Gherkin language
* Write the first step definition in Kotlin
* Confirm the output of cucumber with the positive result

**Step 1:** Create a project directory with cucumber-archetype Maven plugin. Open the terminal, navigate to the project directory, and execute the following command:



* If you haven’t encountered any errors, you will get the following result:



* Create a directory **hellocucumber** inside the project directory, and navigate inside the **hellocucumber** folder.
* Choose any IDE, select the **Open as Project** option, and choose the **pom.xml** file**.**

**Step 2:** Adding Kotlin to the project:

* Add a directory named Kotlin in the src/test directory, and mark it as **Test Sources Root**.
* Create the **hellocucumber** package inside the Kotlin directory.
* Create a Kotlin class named **RunCucumberTest** inside the **hellocucumber** package. Configure the pom.xml to bundle Kotlin.
* Copy the annotations from the **RunCucumberTest.java** class from the previous project to the **RunCucumberTest.kt** that will be created now.
* **RunCucumberTest.kt** class should have the following code generated:

package hellocucumber

import cucumber.api.CucumberOptions

import cucumber.api.junit.Cucumber

import org.junit.runner.RunWith

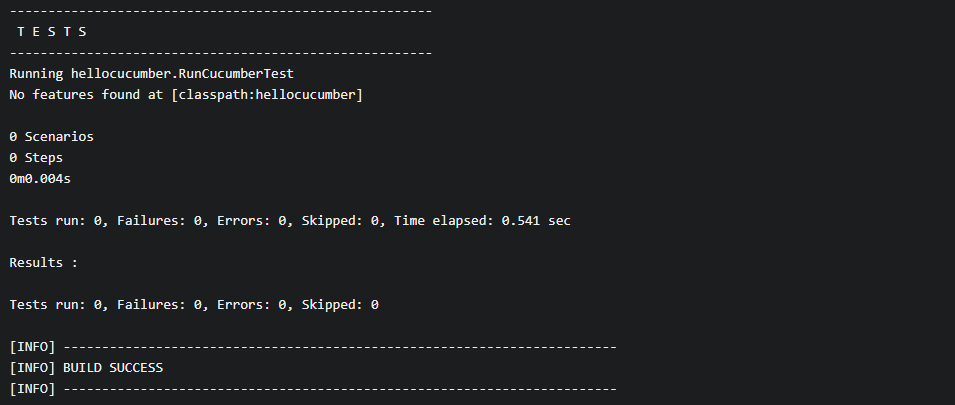
@RunWith(Cucumber::class)

@CucumberOptions(plugin = arrayOf("pretty"))

class RunCucumberTest {

}

* Delete the **RunCucumberTest.java** class, and create a Kotlin class called **Stepdefs** inside the **hellocucumber** package
* Copy the import statements from **Stepdefs.java** to **Stepdefs.kt**, and delete **Stepdefs.java**
* Verify cucumber installation by executing **mvn test**.
* If you haven’t encountered any errors, then you should see the following output.



**Step 3:** Create an empty file called **is\_it\_Friday\_yet.feature** in the directory path **src/test/resources/hellocucumber/is\_it\_Friday\_yet.feature** with the following code:

Feature: Is it Friday yet?

Everybody wants to know when it's Friday

Scenario: Sunday isn't Friday

Given today is Sunday

When I ask whether it's Friday yet

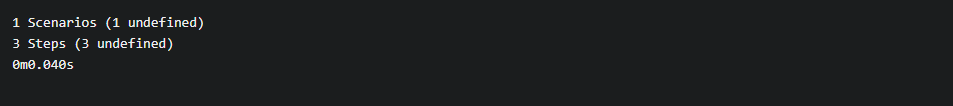
Then I should be told "Nope"

**Step 4:** See scenario reported as undefined

* Execute the following command from the terminal:



* You should confirm the following output:



**Step 5:** See scenario reported as failing

* Create a **stepdefs.kt** at the location **src/test/kotlin/hellocucumber/stepdefs.kt** if it is not already created. Add the below mentioned step definition code:

package hellocucumber

import cucumber.api.java.en.Then

import cucumber.api.java.en.Given

import cucumber.api.java.en.When

import junit.framework.Assert.assertEquals

internal object IsItFriday {

fun isItFriday(today: String): String {

return ""

}

}

class StepDefs {

lateinit var today: String

lateinit var actualAnswer: String

@Given("^today is Sunday$")

fun today\_is\_Sunday() {

this.today = "Sunday"

}

@When("^I ask whether it's Friday yet$")

fun i\_ask\_whether\_is\_s\_Friday\_yet() {

this.actualAnswer = IsItFriday.isItFriday(today)

}

@Then("^I should be told \"([^\"]\*)\"$")

fun i\_should\_be\_told(expectedAnswer: String) {

assertEquals(expectedAnswer, actualAnswer)

}

}

* Run the cucumber. You would be able to confirm that two steps will pass, and the last step will fail.

3

**Step 6:** See scenario as passing

* Add **return ‘Nope’;** to the code available in the step definitionfile.

function isItFriday(today) {

return 'Nope';

}

* Run the cucumber and confirm the following output:

