Automating with Cucumber

Introductory Example

Install Cucumber 4.2 on Eclipse

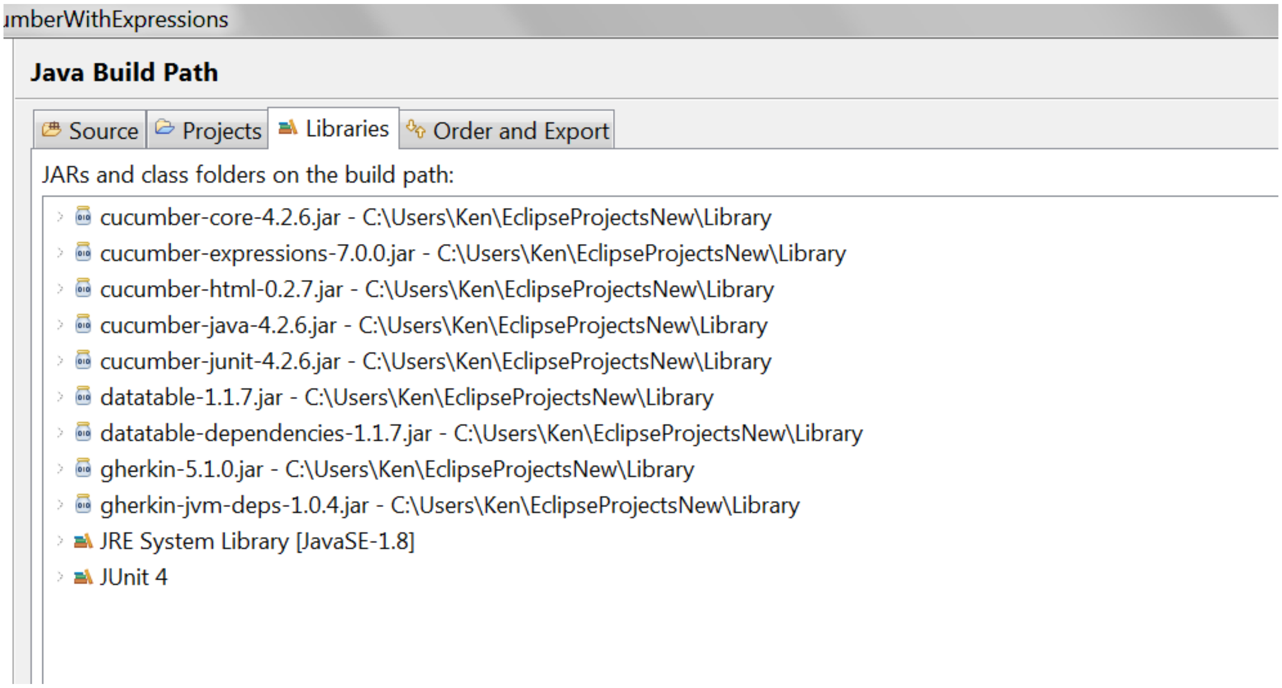
1. Download from http://central.maven.org/maven2/io/cucumber/

   ```
   cucumber-expressions-7.0.0.jar
   cucumber-html-0.2.7.jar
   cucumber-java-4.2.6.jar
   cucumber-junit-4.2.6.jar
   datatable-1.1.7.jar
   datatable-dependencies-1.1.7.jar
   gherkin-5.1.0.jar
   gherkin-jvm-deps-1.0.4.jar
   tag-expressions-1.1.1.jar
   cucumber-core-4.2.6.jar
   ```

2. **Create a new Project in Eclipse**. Using an existing project may cause issues. Creating the project in a new workspace may also eliminate issues.

3. Add the previous jars as dependencies (in the build path as libraries). If you prefer using maven, then add the dependencies to the project file.

   ```
   <dependency>
     <groupId>io.cucumber</groupId>
     <artifactId>cucumber-java</artifactId>
     <version>4.2.6</version>
   </dependency>

   <dependency>
     <groupId>io.cucumber</groupId>
     <artifactId> cucumber-html</artifactId>
     <version>0.2.7</version>
   </dependency>
   ```



Java Build Path

JARs and class folders on the build path:

- cucumber-core-4.2.6.jar - C:\Users\Ken\EclipseProjectsNew\Library
- cucumber-expressions-7.0.0.jar - C:\Users\Ken\EclipseProjectsNew\Library
- cucumber-html-0.2.7.jar - C:\Users\Ken\EclipseProjectsNew\Library
- cucumber-java-4.2.6.jar - C:\Users\Ken\EclipseProjectsNew\Library
- cucumber-junit-4.2.6.jar - C:\Users\Ken\EclipseProjectsNew\Library
- datatable-1.1.7.jar - C:\Users\Ken\EclipseProjectsNew\Library
- datatable-dependencies-1.1.7.jar - C:\Users\Ken\EclipseProjectsNew\Library
- gherkin-5.1.0.jar - C:\Users\Ken\EclipseProjectsNew\Library
- gherkin-jvm-deps-1.0.4.jar - C:\Users\Ken\EclipseProjectsNew\Library
- JRE System Library [JavaSE-1.8]
- JUnit 4

Install Cucumber 5.6 on Eclipse

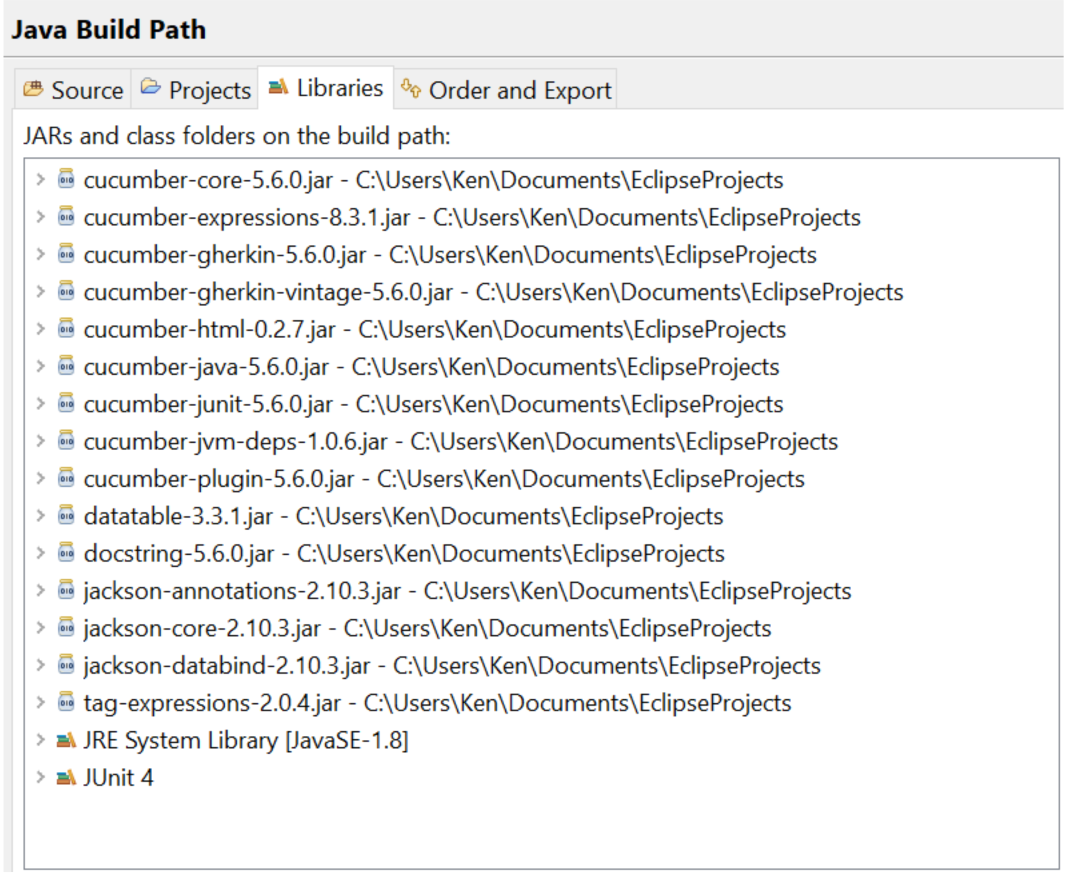
1. Download from http://central.maven.org/maven2/io/cucumber/

   cucumber-core-5.6.0.jar
   cucumber-expressions-8.3.1.jar   // Using a later one does not work
   cucumber-gherkin-5.6.0.jar
   cucumber-gherkin-vintage-5.6.0.jar
   cucumber-html-0.2.7.jar
   cucumber-java-5.6.0.jar
   cucumber-junit-5.6.0.jar
   cucumber-jvm-deps-1.0.6.jar
   cucumber-plugin-5.6.0.jar
   datatable-3.3.1.jar
   docstring-5.6.0.jar
   jackson-annotations-2.10.3.jar
   jackson-core-2.10.3.jar
   jackson-databind-2.10.3.jar
   tag-expressions-2.0.4.jar

2. **Create a new Project in Eclipse**. Using an existing project may cause issues.   Creating the project in a new workspace may also eliminate issues.

3. Add the previous jars as dependencies (in the build path as libraries). If you prefer using maven, then add the dependencies to the project file.

```
<dependency>
   <groupId>io.cucumber</groupId>
   <artifactId>cucumber-java</artifactId>
   <version>5.6.0</version>
</dependency>

<dependency>
   <groupId>io.cucumber</groupId>
   <artifactId> cucumber-html</artifactId>
   <version>0.2.7</version>
</dependency>
```

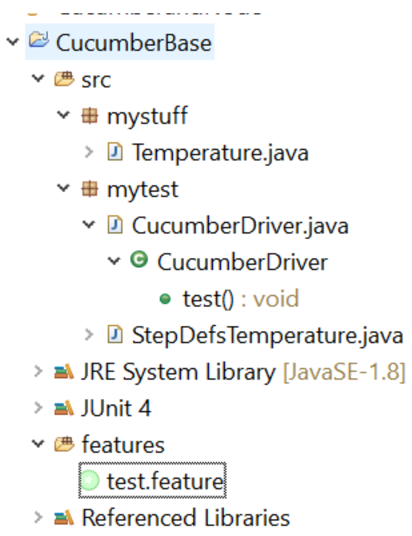


JARs and class folders on the build path:

```
> cucumber-core-5.6.0.jar - C:\Users\Ken\Documents\EclipseProjects
> cucumber-expressions-8.3.1.jar - C:\Users\Ken\Documents\EclipseProjects
> cucumber-gherkin-5.6.0.jar - C:\Users\Ken\Documents\EclipseProjects
> cucumber-gherkin-vintage-5.6.0.jar - C:\Users\Ken\Documents\EclipseProjects
> cucumber-html-0.2.7.jar - C:\Users\Ken\Documents\EclipseProjects
> cucumber-java-5.6.0.jar - C:\Users\Ken\Documents\EclipseProjects
> cucumber-junit-5.6.0.jar - C:\Users\Ken\Documents\EclipseProjects
> cucumber-jvm-deps-1.0.6.jar - C:\Users\Ken\Documents\EclipseProjects
> cucumber-plugin-5.6.0.jar - C:\Users\Ken\Documents\EclipseProjects
> datatable-3.3.1.jar - C:\Users\Ken\Documents\EclipseProjects
> docstring-5.6.0.jar - C:\Users\Ken\Documents\EclipseProjects
> jackson-annotations-2.10.3.jar - C:\Users\Ken\Documents\EclipseProjects
> jackson-core-2.10.3.jar - C:\Users\Ken\Documents\EclipseProjects
> jackson-databind-2.10.3.jar - C:\Users\Ken\Documents\EclipseProjects
> tag-expressions-2.0.4.jar - C:\Users\Ken\Documents\EclipseProjects
> JRE System Library [JavaSE-1.8]
> JUnit 4
```

Add the Sample Project (Cucumber 4.2)

1. Create the files shown below. Here's the layout:



```
CucumberBase
  src
    mystuff
      Temperature.java
    mytest
      CucumberDriver.java
        CucumberDriver
          test() : void
      StepDefsTemperature.java
  JRE System Library [JavaSE-1.8]
  JUnit 4
  features
    test.feature
  Referenced Libraries
```

```
// test.feature into the features directory

//   (create features directory in the project folder )

Feature: Temperature Converter

Scenario:  Convert Freezing
Given Celsius is 0
When converted to Fahrenheit
Then result is 32

Scenario:  Convert Boiling
Given Celsius is 100
When converted to Fahrenheit
Then result is 212
```

```
// StepDefsTemperature.java (package mytest)

package mytest;
import static org.junit.Assert.assertEquals;
import cucumber.api.java.en.Given;
import cucumber.api.java.en.Then;
import cucumber.api.java.en.When;
import mystuff.Temperature;
public class StepDefsTemperature {
       int fahrenheit;
       int celsius;

       @Given("Celsius is {int}")
       public void celsius_is(int int1) {
           celsius = int1;
       }
```

```
      @When("converted to Fahrenheit")
      public void converted_to_Fahrenheit() {
          fahrenheit = Temperature.convertCelsiusToFahrenheit(celsius);
      }

      @Then("result is {int}")
      public void result_is(int int1) {
         assertEquals(int1, fahrenheit);
      }
}
```

//Temperature.java (package mystuff)

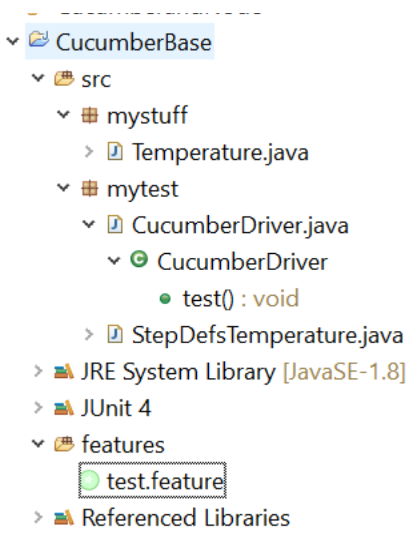
```
package mystuff;
public class Temperature {
      public static int convertCelsiusToFahrenheit(int celsius) {
             return 32;
      }
}
```

// CucumberDriver.java (package mytest) for 4.2

```
package mytest;
import org.junit.Test;
import org.junit.runner.RunWith;
import cucumber.api.CucumberOptions;
import cucumber.api.junit.Cucumber;

@RunWith (Cucumber.class)
@CucumberOptions (features ="features",
   plugin = {"pretty", "html:target/cucumber"},
   glue = "classpath:mytest")
public class CucumberDriver {
@Test
      public void
          test() {
      }
}
```

Add the Sample Project (5.6)

1. Create the files shown below. Here’s the layout:



```
CucumberBase
  src
    mystuff
      Temperature.java
    mytest
      CucumberDriver.java
        CucumberDriver
          test() : void
      StepDefsTemperature.java
  JRE System Library [JavaSE-1.8]
  JUnit 4
  features
    test.feature
  Referenced Libraries
```

```
// test.feature into the features directory

//   (create features directory in the project folder )
```

```
Feature: Temperature Converter

Scenario:  Convert Freezing
Given Celsius is 0
When converted to Fahrenheit
Then result is 32

Scenario:  Convert Boiling
Given Celsius is 100
When converted to Fahrenheit
Then result is 212
```

```
// StepDefsTemperature.java (package mytest) with 5.6 imports
```

```java
package mytest;
import static org.junit.Assert.assertEquals;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
import mystuff.Temperature;
public class StepDefsTemperature {
       int fahrenheit;
       int celsius;

       @Given("Celsius is {int}")
       public void celsius_is(int int1) {
           celsius = int1;
       }

       @When("converted to Fahrenheit")
       public void converted_to_Fahrenheit() {
           fahrenheit = Temperature.convertCelsiusToFahrenheit(celsius);
       }
```

```
      @Then("result is {int}")
      public void result_is(int int1) {
         assertEquals(int1, fahrenheit);
      }
}
```

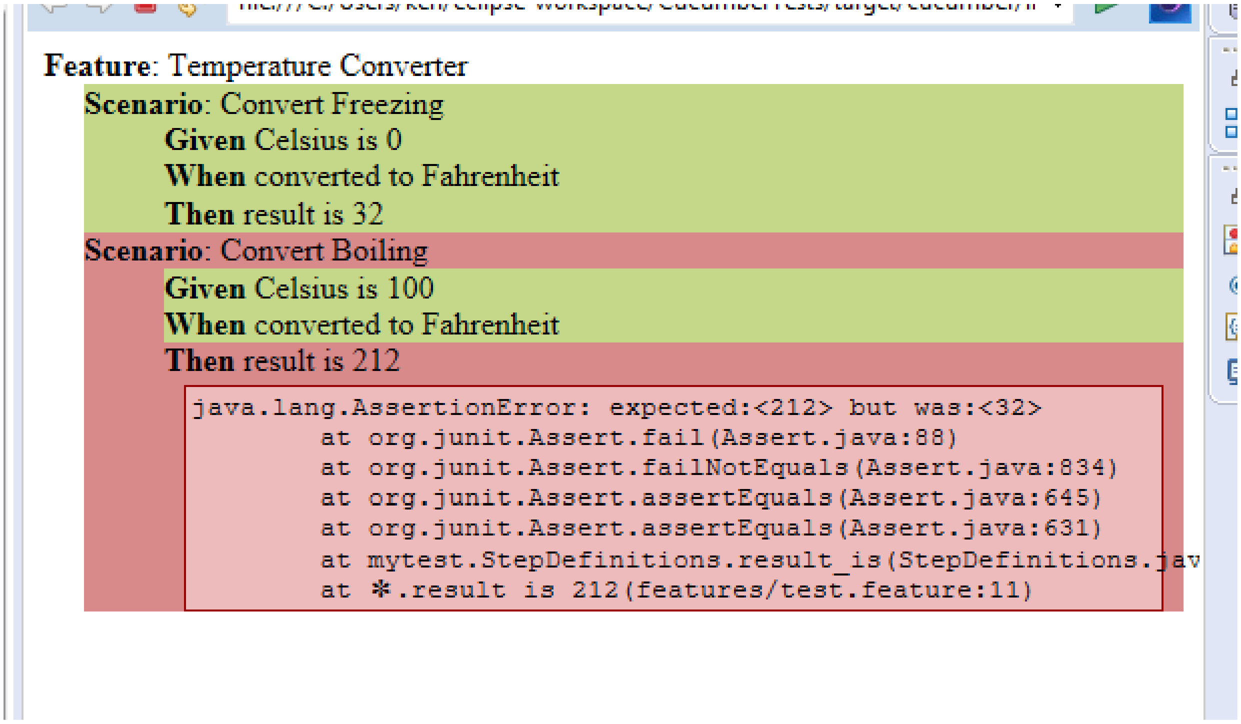
```
//Temperature.java (package mystuff)

package mystuff;
public class Temperature {
      public static int convertCelsiusToFahrenheit(int celsius) {
            return 32;
      }
}
```

```
// CucumberDriver.java (package mytest) for 5.6

package mytest;
import org.junit.Test;
import org.junit.runner.RunWith;
import io.cucumber.junit.Cucumber;
import io.cucumber.junit.CucumberOptions;

@RunWith(Cucumber.class)
@CucumberOptions(
      features ="feature",
      glue = "classpath:",
      strict = true,
      plugin = { "pretty","html:target/cucumber"})

public class CucumberDriver {
@Test
public void test() {
      }
}
```

Run the Project

1. Run CucumberDriver.java as a Junit test. JUnit will also need to be added when you attempt to run it or you can add it as a library to the project.

2. Then browse to “target/cucumber/index.html”. (Do a refresh on the project to see these files in Eclipse). The results should show one success (0 to 32) and one error (100 to 212).



Feature: Temperature Converter

Scenario: Convert Freezing

Given Celsius is 0

When converted to Fahrenheit

Then result is 32

Scenario: Convert Boiling

Given Celsius is 100

When converted to Fahrenheit

Then result is 212

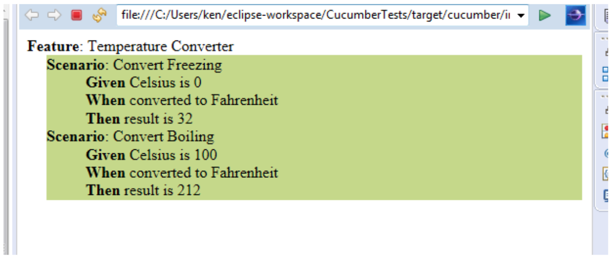
```
java.lang.AssertionError: expected:<212> but was:<32>
        at org.junit.Assert.fail(Assert.java:88)
        at org.junit.Assert.failNotEquals(Assert.java:834)
        at org.junit.Assert.assertEquals(Assert.java:645)
        at org.junit.Assert.assertEquals(Assert.java:631)
        at mytest.StepDefinitions.result_is(StepDefinitions.jav
        at ✽.result is 212(features/test.feature:11)
```

3. Change Temperature class to make the 100 to 212 test pass.

See https://cucumber.io/docs/cucumber/ for more information on details not covered in this course. .

Possible Solution

```
//Temperature.java

public class Temperature {
      public static int convertCelsiusToFahrenheit(int celsius) {
            return (celsius * 9)/5 + 32;
      }
}
```



file:///C:/Users/ken/eclipse-workspace/CucumberTests/target/cucumber/i

```
Feature: Temperature Converter
   Scenario: Convert Freezing
         Given Celsius is 0
         When converted to Fahrenheit
         Then result is 32
   Scenario: Convert Boiling
         Given Celsius is 100
         When converted to Fahrenheit
         Then result is 212
```