Data table is not same as example. Data table is not for data driven testing. It is for filling in data.

Scenario outline for data driven testing.

When we use scenario-outline we need to use examples keyword.

Examples keyword won’t come with normal scenario.

“Examples” keyword used to achieve data driven.

Sometimes if the feature file is not showing the glued flags (orange highlight), right click on project and convert to cucumber or update the maven project, which we have done already many times.

Shortcut to import everything in a page is “control+shift+o”.

Move the mouse over the (given, when) statements, which we have written etc and press “control + click” and you go inside the step def file.

# Code with basic examples and scenario outline-

Feature file-

|  |
| --- |
| Feature: login feature  Scenario Outline: failed login - different combinations  Given user is on application landing page  When user clicks on signin button  Then user is displayed login page  When user enters "<userName>" in username field  When user enters "<password>" in password field  And user clicks on signin button  Then error displayed for wrong credentials    Examples:  | userName | password |  | incorrectusername | 234324 |  | naveen auto | incorrectpassword |  | incorrectusername | incorrectpassword | |

Step def-

|  |
| --- |
| **package** StepDefinitions;  **import** io.cucumber.java.en.Given;  **import** io.cucumber.java.en.Then;  **import** io.cucumber.java.en.When;  **public** **class** LoginFeatureStepDef {    @Given("user is on application landing page")  **public** **void** user\_is\_on\_application\_landing\_page() {    }  @When("user clicks on signin button")  **public** **void** user\_clicks\_on\_signin\_button() {    }  @Then("user is displayed login page")  **public** **void** user\_is\_displayed\_login\_page() {    }  @When("user enters {string} in username field")  **public** **void** user\_enters\_in\_username\_field(String string) {    }  @When("user enters {string} in password field")  **public** **void** user\_enters\_in\_password\_field(String string) {    }  @Then("error displayed for wrong credentials")  **public** **void** error\_displayed\_for\_wrong\_credentials() {    }  } |

Output-

A screenshot of a computer program

Description automatically generated with low confidence

A picture containing text, screenshot, font, number

Description automatically generated

# Now see this –

We have same numeric field taking int and decimals.

A screenshot of a computer code

Description automatically generated with low confidence

In step def there will be overloaded method with int as param and double as param.

A picture containing text, font, line, number

Description automatically generated

In double we can store integer as well as double.

# Let’s see issue with cucumber which Naveen reported but cucumber guys and girls told it’s the actual working-

Billing feature-

|  |
| --- |
| Feature: calculate billing amount  Scenario Outline: bill amount  Given user is on billing page  When user enters bill amount <billAmount>  When user enters tax amount <taxAmount>  And user clicks calculate button  Then final amount is given <finalAmount>    Examples:  | billAmount | taxAmount | finalAmount |  | 1000 | 10 | 1010 |  | 100 | 40 | 140 |  | 20 | 6.7 | 26.7 | |

Billing step def-

|  |
| --- |
| **package** StepDefinitions;  **import** io.cucumber.java.en.Given;  **import** io.cucumber.java.en.Then;  **import** io.cucumber.java.en.When;  **import** junit.framework.~~Assert~~;  **public** **class** BillingStepDef {    **int** billingAmount;  **double** taxAmount;  **double** finalAmount;    @Given("user is on billing page")  **public** **void** user\_is\_on\_billing\_page() {    }  @When("user enters bill amount {int}")  **public** **void** user\_enters\_bill\_amount(Integer billingAmount) {  **this**.billingAmount=billingAmount;  }  @When("user enters tax amount {int}")  **public** **void** user\_enters\_tax\_amount(Integer taxAmount) {  **this**.taxAmount=taxAmount;  }    @When("user enters tax amount {double}")  **public** **void** user\_enters\_tax\_amount(Double taxAmount) {  **this**.taxAmount=taxAmount;  }  @When("user clicks calculate button")  **public** **void** user\_clicks\_calculate\_button() {    }  @Then("final amount is given {int}")  **public** **void** final\_amount\_is\_given(Integer finalAmount) {  **this**.finalAmount=**this**.billingAmount+**this**.taxAmount;  ~~Assert~~.~~assertTrue~~(**this**.finalAmount==finalAmount); // if the condition is true then assertion should be passed  }    @Then("final amount is given {double}")  **public** **void** final\_amount\_is\_given(Double finalAmount) {  **this**.finalAmount=**this**.billingAmount+**this**.taxAmount;  ~~Assert~~.~~assertTrue~~(**this**.finalAmount==finalAmount);  }  } |

Run feature and we get below exception-

It says there are multiple step def’s which are matching. This is because integer can be stored in double also and it is confused which one to select.

With cucumber 4.0 it was working fine, from cucumber 6 this issue comes.

|  |
| --- |
| When user enters tax amount 40 # null  io.cucumber.core.runner.AmbiguousStepDefinitionsException: "user enters tax amount 40" matches more than one step definition:  "user enters tax amount {double}" in StepDefinitions.BillingStepDef.user\_enters\_tax\_amount(java.lang.Double)  "user enters tax amount {int}" in StepDefinitions.BillingStepDef.user\_enters\_tax\_amount(java.lang.Integer)  at io.cucumber.core.runner.CachingGlue.findStepDefinitionMatch(CachingGlue.java:373)  at io.cucumber.core.runner.CachingGlue.stepDefinitionMatch(CachingGlue.java:341)  at io.cucumber.core.runner.Runner.matchStepToStepDefinition(Runner.java:146)  at io.cucumber.core.runner.Runner.createTestStepsForPickleSteps(Runner.java:126)  at io.cucumber.core.runner.Runner.createTestCaseForPickle(Runner.java:109)  at io.cucumber.core.runner.Runner.runPickle(Runner.java:70)  at io.cucumber.core.runtime.Runtime.lambda$execute$5(Runtime.java:110)  at io.cucumber.core.runtime.CucumberExecutionContext.runTestCase(CucumberExecutionContext.java:117)  at io.cucumber.core.runtime.Runtime.lambda$execute$6(Runtime.java:110)  at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)  at java.util.concurrent.FutureTask.run(FutureTask.java:266)  at io.cucumber.core.runtime.Runtime$SameThreadExecutorService.execute(Runtime.java:233)  at java.util.concurrent.AbstractExecutorService.submit(AbstractExecutorService.java:112)  at io.cucumber.core.runtime.Runtime.lambda$run$2(Runtime.java:86)  at java.util.stream.ReferencePipeline$3$1.accept(ReferencePipeline.java:193)  at java.util.stream.SliceOps$1$1.accept(SliceOps.java:204)  at java.util.ArrayList$ArrayListSpliterator.tryAdvance(ArrayList.java:1359)  at java.util.stream.ReferencePipeline.forEachWithCancel(ReferencePipeline.java:126)  at java.util.stream.AbstractPipeline.copyIntoWithCancel(AbstractPipeline.java:498)  at java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.java:485)  at java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeline.java:471)  at java.util.stream.ReduceOps$ReduceOp.evaluateSequential(ReduceOps.java:708)  at java.util.stream.AbstractPipeline.evaluate(AbstractPipeline.java:234)  at java.util.stream.ReferencePipeline.collect(ReferencePipeline.java:499)  at io.cucumber.core.runtime.Runtime.run(Runtime.java:87)  at io.cucumber.core.cli.Main.run(Main.java:92)  at cucumber.api.cli.Main.run(Main.java:30)  at cucumber.api.cli.Main.main(Main.java:15) |

See with cucumber 4 –

Change in pom file the version of cucumber.

A screenshot of a computer program

Description automatically generated with low confidence

Save it and it will rebuild.

Run feature and it works.

A screenshot of a computer

Description automatically generated

# This is the official cucumber git hub page-

A screenshot of a web page

Description automatically generated with medium confidenceTo solve the issue of overloaded methods use the input always in form of string in real time coding-

Billing feature-

|  |
| --- |
| Feature: calculate billing amount  Scenario Outline: bill amount  Given user is on billing page  When user enters bill amount "<billAmount>"  When user enters tax amount "<taxAmount>"  And user clicks calculate button  Then final amount is given "<finalAmount>"    Examples:  | billAmount | taxAmount | finalAmount |  | 1000 | 10 | 1010 |  | 100 | 40 | 140 |  | 20 | 6.7 | 26.7 | |

Step def-

|  |
| --- |
| **package** StepDefinitions;  **import** io.cucumber.java.en.Given;  **import** io.cucumber.java.en.Then;  **import** io.cucumber.java.en.When;  **import** junit.framework.~~Assert~~;  **public** **class** BillingStepDef {    **double** billingAmount;  **double** taxAmount;  **double** finalAmount;    @Given("user is on billing page")  **public** **void** user\_is\_on\_billing\_page() {    }  @When("user enters bill amount {string}")  **public** **void** user\_enters\_bill\_amount(String billingAmount) {  **this**.billingAmount=Double.*parseDouble*(billingAmount); //convert string to double and give to the numeric guy/girl  }  @When("user enters tax amount {string}")  **public** **void** user\_enters\_tax\_amount(String taxAmount) {  **this**.taxAmount=Double.*parseDouble*(taxAmount);  }  @When("user clicks calculate button")  **public** **void** user\_clicks\_calculate\_button() {    }  @Then("final amount is given {string}")  **public** **void** final\_amount\_is\_given(String finalAmount) {  **this**.finalAmount=**this**.billingAmount+**this**.taxAmount; //this.finalAmount is already double as we have declared it.  System.***out***.println("expected final amount is " + " " + Double.*parseDouble*(finalAmount));  System.***out***.println("Actual final amount is " + " " + **this**.finalAmount);  ~~Assert~~.~~assertTrue~~(**this**.finalAmount==Double.*parseDouble*(finalAmount)); // if the condition is true then assertion should be passed  }  } |

Output-

A screenshot of a computer program

Description automatically generated with low confidence

Runner file for billing-

|  |
| --- |
| **package** testRunners;  **import** io.cucumber.junit.Cucumber;  **import** io.cucumber.junit.CucumberOptions;  **import** org.junit.runner.RunWith;  @RunWith(Cucumber.**class**)  @CucumberOptions(  plugin= {"pretty"},  features = {"src/test/resources/AppFeatures/Billing.feature" },  glue = {"StepDefinitions"}  )  **public** **class** BillingTest {  } |

Junit-

A screenshot of a computer

Description automatically generated

Console-

A screenshot of a computer program

Description automatically generated with low confidence

We have accomplished data driven testing.

# Project structure-

A screenshot of a computer

Description automatically generated with medium confidence

# Codes for this lecture-

|  |
| --- |
| Feature file for login –  Feature: login feature  Scenario Outline: failed login - different combinations  Given user is on application landing page  When user clicks on signin button  Then user is displayed login page  When user enters "<userName>" in username field  When user enters "<password>" in password field  And user clicks on signin button  Then error displayed for wrong credentials    Examples:  | userName | password |  | incorrectusername | 234324 |  | naveen auto | incorrectpassword |  | incorrectusername | incorrectpassword | |
| Step def for invalid login-  **package** StepDefinitions;  **import** io.cucumber.java.en.Given;  **import** io.cucumber.java.en.Then;  **import** io.cucumber.java.en.When;  **public** **class** LoginFeatureStepDef {    @Given("user is on application landing page")  **public** **void** user\_is\_on\_application\_landing\_page() {    }  @When("user clicks on signin button")  **public** **void** user\_clicks\_on\_signin\_button() {    }  @Then("user is displayed login page")  **public** **void** user\_is\_displayed\_login\_page() {    }  @When("user enters {string} in username field")  **public** **void** user\_enters\_in\_username\_field(String string) {    }  @When("user enters {string} in password field")  **public** **void** user\_enters\_in\_password\_field(String string) {    }  @Then("error displayed for wrong credentials")  **public** **void** error\_displayed\_for\_wrong\_credentials() {    }  } |
| Billing feature-  Feature: calculate billing amount  Scenario Outline: bill amount  Given user is on billing page  When user enters bill amount "<billAmount>"  When user enters tax amount "<taxAmount>"  And user clicks calculate button  Then final amount is given "<finalAmount>"    Examples:  | billAmount | taxAmount | finalAmount |  | 1000 | 10 | 1010 |  | 100 | 40 | 140 |  | 20 | 6.7 | 26.7 | |

|  |
| --- |
| Billing step def-  **package** StepDefinitions;  **import** io.cucumber.java.en.Given;  **import** io.cucumber.java.en.Then;  **import** io.cucumber.java.en.When;  **import** junit.framework.~~Assert~~;  **public** **class** BillingStepDef {    **double** billingAmount;  **double** taxAmount;  **double** finalAmount;    @Given("user is on billing page")  **public** **void** user\_is\_on\_billing\_page() {    }  @When("user enters bill amount {string}")  **public** **void** user\_enters\_bill\_amount(String billingAmount) {  **this**.billingAmount=Double.*parseDouble*(billingAmount); //convert string to double and give to the numeric guy/girl  }  @When("user enters tax amount {string}")  **public** **void** user\_enters\_tax\_amount(String taxAmount) {  **this**.taxAmount=Double.*parseDouble*(taxAmount);  }  @When("user clicks calculate button")  **public** **void** user\_clicks\_calculate\_button() {    }  @Then("final amount is given {string}")  **public** **void** final\_amount\_is\_given(String finalAmount) {  **this**.finalAmount=**this**.billingAmount+**this**.taxAmount; //this.finalAmount is already double as we have declared it.  System.***out***.println("expected final amount is " + " " + Double.*parseDouble*(finalAmount));  System.***out***.println("Actual final amount is " + " " + **this**.finalAmount);  ~~Assert~~.~~assertTrue~~(**this**.finalAmount==Double.*parseDouble*(finalAmount)); // if the condition is true then assertion should be passed  }  } |
| Billing runner-  **package** testRunners;  **import** io.cucumber.junit.Cucumber;  **import** io.cucumber.junit.CucumberOptions;  **import** org.junit.runner.RunWith;  @RunWith(Cucumber.**class**)  @CucumberOptions(  plugin= {"pretty"},  features = {"src/test/resources/AppFeatures/Billing.feature" },  glue = {"StepDefinitions"}  )  **public** **class** BillingTest {  } |