# Not a great format to use-

If there are 100 fields what we will do?

A picture containing text, font, screenshot

Description automatically generated

Feature file:

|  |
| --- |
| Feature: user registration  Scenario: user registration with different data  Given user is on registration page  When user enters the following details  | naveen | automation | nav@gmail | 234324 | bangalore |  | tom | manual | tom@yahoo | 23432 | pune |  | robert | sdet | robert@michi | 234234 | london |  Then user registration is successful |

Step def:

|  |
| --- |
| package StepDefinitions;  import java.util.List;  import io.cucumber.datatable.DataTable;  import io.cucumber.java.en.Given;  import io.cucumber.java.en.Then;  import io.cucumber.java.en.When;  public class UserRegistration {    @Given("user is on registration page")  public void user\_is\_on\_registration\_page() {  System.out.println("user navigates on registration page");  }  @When("user enters the following details")  public void user\_enters\_the\_following\_details(DataTable dataTable) {  List<List<String>> userList=dataTable.asLists(String.class);// this step will first take the String input.  //as Lists returns List of List of Objects.  //this for loop will print row by row data.  for(List<String> e:userList) {  System.out.println(e);  }      }  @Then("user registration is successful")  public void user\_registration\_is\_successful() {  System.out.println("user registration is successful");  }  } |

Output:

A picture containing text, screenshot, font, number

Description automatically generated

Data table used when we want to fill in large quantities of data.

Example, form filling.

Data table can be applied for any step – given or when or then etc.

Created runner file:

|  |
| --- |
| **package** testRunners;  **import** io.cucumber.junit.Cucumber;  **import** io.cucumber.junit.CucumberOptions;  **import** org.junit.runner.RunWith;  @RunWith(Cucumber.**class**)  @CucumberOptions( plugin= {"pretty", "json:target/MyReports/report.json",  "junit:target/MyReports/report.xml"},  features = {"src/test/resources/AppFeatures/UserRegistration.feature" },  glue = { "StepDefinitions"},  monochrome=**false**,  dryRun =**false**  )  **public** **class** UserRegistrationTest {  } |

Output:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

See report how it is seen.

A screenshot of a computer

Description automatically generated

# Project structure-

A screenshot of a computer program

Description automatically generated with low confidence

# Codes for this chapter-

Feature.

|  |
| --- |
| Feature: user registration  Scenario: user registration with different data  Given user is on registration page  When user enters the following details  | naveen | automation | nav@gmail | 234324 | bangalore |  | tom | manual | tom@yahoo | 23432 | pune |  | robert | sdet | robert@michi | 234234 | london |  Then user registration is successful |

Step def.

|  |
| --- |
| package StepDefinitions;  import java.util.List;  import io.cucumber.datatable.DataTable;  import io.cucumber.java.en.Given;  import io.cucumber.java.en.Then;  import io.cucumber.java.en.When;  public class UserRegistration {    @Given("user is on registration page")  public void user\_is\_on\_registration\_page() {  System.out.println("user navigates on registration page");  }  @When("user enters the following details")  public void user\_enters\_the\_following\_details(DataTable dataTable) {  List<List<String>> userList=dataTable.asLists(String.class);// this step will first take the String input.  //as Lists returns List of List of Objects.  //this for loop will print row by row data.  for(List<String> e:userList) {  System.out.println(e);  }      }  @Then("user registration is successful")  public void user\_registration\_is\_successful() {  System.out.println("user registration is successful");  }  } |

Runner:

|  |
| --- |
| **package** testRunners;  **import** io.cucumber.junit.Cucumber;  **import** io.cucumber.junit.CucumberOptions;  **import** org.junit.runner.RunWith;  @RunWith(Cucumber.**class**)  @CucumberOptions( plugin= {"pretty", "json:target/MyReports/report.json",  "junit:target/MyReports/report.xml"},  features = {"src/test/resources/AppFeatures/UserRegistration.feature" },  glue = { "StepDefinitions"},  monochrome=**false**,  dryRun =**false**  )  **public** **class** UserRegistrationTest {  } |