**BDD Framework**

**BDD - Behavior Driven Development Framework**

## Components of BDD

### Feature File

### Step Definition File

### Test Runner File

## Feature file

.feature file which has below **Gherkin language Keywords**

## Feature: It is a list of scenarios

## Scenario: Business rule through a list of steps with arguments

## Given: Some precondition step

## When: Some key actions

## Then: To observe outcomes or validation

## And, But: To enumerate more Given, When, Then steps

## Scenario Outline: List of steps for data-driven as an Examples and <placeholder>

## Examples: Container for table

## Background: List of steps run before each of the scenarios

## """ (Doc Strings): This is used for representing strings

## | (Data Tables): This is used to present the data values in table form

## @(Tags/Labels): To group Scenarios

## <>: (placeholder)

## #: For comments in Feature File

## Step Definition

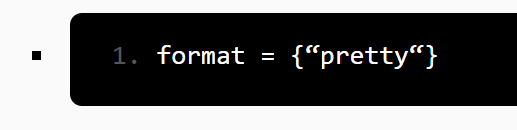
is under step definition package

* 1. Will be containing step definition for each step defined in Feature file

## Test Runner

is under runner package

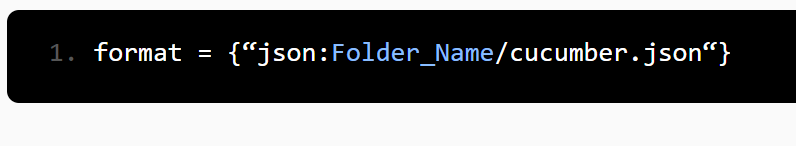
* **dry run:** Checks if the step definition contains method definition for each step mentioned in .feature file. By default it is 'false', you should set it 'true'.
* **features:** Mention the paths for the feature's files.
* **glue:** Mention paths for step definition files.
* **tags:** This will specify what tags in the feature files should be executed. (Feature file tags will learn in upcoming slides.)
* **monochrome:**Display the console output in a readable way. This should set as "true"
* **format:** Used to specify different formatting options for the output reports. Various options that can be used as for-matters are:
  + Pretty: Prints the Gherkin source with additional colors and stack traces for errors. Syntax is as below:



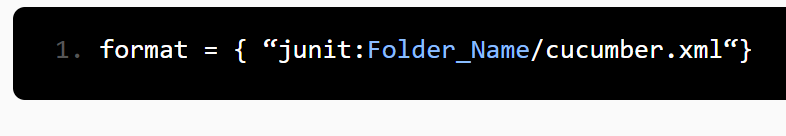
* + HTML: This will generate an HTML report at the location mentioned in the for-matter itself. Syntax is as below:



* + JSON: This report contains all the information from the gherkin source in JSON Format. Syntax is as below:



* + JUnit: This report generates XML files just like Apache Ant’s JUnit report task. This XML format is understood by most Continuous Integration servers, who will use it to generate visual reports. Syntax is as below:



## Maven Folder Structure:

1. src/main/java
   1. com.flipkart.base
   2. com.flipkart.pages
   3. com.flipkart.utils
2. src/main/resources
3. Features
   1. Login.feature
4. src/test/java
   1. stepDefinition
      1. PlaceOrdreStepDef.java
   2. Runner
      1. TestRunner.java
5. src/test/resources
6. Drivers
7. Test Data
8. Screenshots

## To Install Cucumber tool

1. Launch the Eclipse IDE. In its 'Help' menu, click 'Install New Software'.
2. You will see a dialog box in which you need to click the 'Add' button.
3. Type a name as you wish. Let us take 'Cucumber'.
4. In the location field, type the following URL " https://cucumber.github.io/cucumber-eclipse-update-site-snapshot/ " and click on 'Ok' button.

