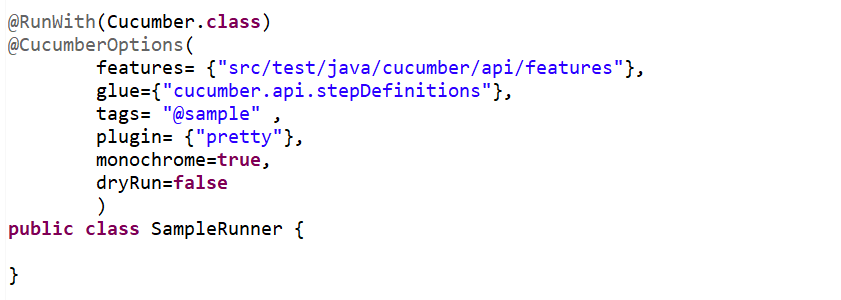
**CucumberOptions:**



**Monochrome**: If it is set as true, it means that the console output for the Cucumber test are much more readable

The default option for monochrome is false

**Strict:** if strict option is set to false then at execution time if cucumber encounters any undefined/pending steps then cucumber does not fail the execution and undefined steps are skipped and BUILD is SUCCESSFUL.

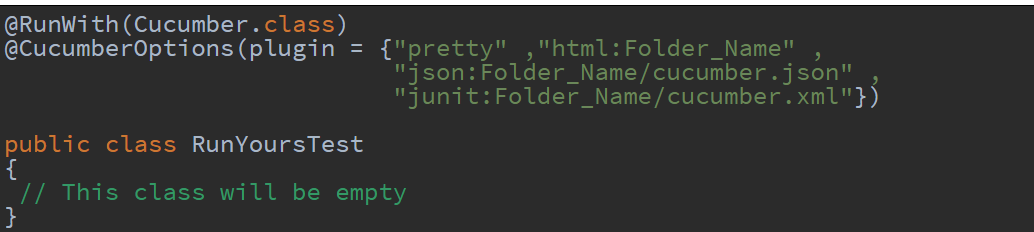
The default option for strict is true

**DryRun:** If it is set as true, it means that Cucumber will only checks that every Step mentioned in the Feature File have corresponding code written in Step Definition file or not.

The default option for DryRun is false

**Plugin:**plugin Option is 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
* **HTML**: This will generate a HTML report at the location mentioned in the for-matter itself.
* **JSON**: This report contains all the information from the gherkin source in JSON Format.
* **JUnit**: This report generates XML files just like Apache Ant’s JUnit report task.



**Features:** Features Options helps Cucumber to locate the Feature file in the project folder structure.All we need to do is to specify the **folder path** and Cucumber will automatically find all the ‘.features‘ **extension files in the folder**

**Glue:**It helps Cucumber to locate Step Definition files. Whenever Cucumber encounters a Step, it looks for a Step Definition inside all the **sub packages** present in the package mentioned in Glue Option.

**Tags:** Tags can be used when specifying what tests to run through tags define in feature file.

## Scenario hooks

Scenario hooks run for every scenario.

@Before and @After

@After

public void doSomethingAfter(Scenario scenario){

// Do something after after scenario

}

**Step hooks**

Step hooks invoked before and after a step. The hooks have ‘invoke around’ semantics. Meaning that if a BeforeStep hook is executed the AfterStep hooks will also be executed regardless of the result of the step. If a step did not pass, the following step and its hooks will be skipped.

**Dependencies Injection :** Dependency injection is needed to share the state of object across the class (steps). Pickocontainer is example of it, (spring and google guice etc.).