

**Problem 1:** when we are writing new scenarios we will have undefined steps. We do not want to wait until defined steps get executed. Without dryRun we have to wait execution of defined steps until cucumber reaches to undefined steps.

We use dryRun to get only undefined step definitions.

dryRun = true

**If dryRun is true** -> cucumber will NOT execute the java code inside the step definitions. It will only check if the cucumber scenarios steps have matching step definitions.

**If dryRun is false** -> cucumber will execute normal, meaning java code will be executed inside the each step definitions.

**Problem 2:** our test lead asked to run only store manager related login test.

How can I run specific scenario ?

**tags** -> tags is another cucumber option, it allows us to run certain scenario(s)

**How to apply ?**

We are labeling the scenario we want to run with @tag then in CukesRunner we are using same tag to run that scenario.

```
@store_manager
```

```
Scenario: Login as a store manager
```

```
    Given the user is on the login page
```

```
    When the user enters the store manager information
```

```
    Then the user should be able to login
```

```

@RunWith(Cucumber.class)
@CucumberOptions(
    features = "src/test/resources/features",
    glue = "com/vytrack/step_definitions",
    dryRun = false,
    tags = "@store_manager"
)
public class CukesRunner {
}

```

-Tags are case sensitive

- if the tags comes on top of the scenario, it applies to that scenario only.

We can use same tag in **different scenarios**.

CukesRunner will execute all of them.

```

@driver @smoke
Scenario: Login as a driver
    Given the user is on the login page
    When the user enters the driver information
    Then the user should be able to login
@sales_manager
Scenario: Login as a sales manager
    Given the user is on the login page
    When the user enters the sales manager information
    Then the user should be able to login
@store_manager @smoke
Scenario: Login as a store manager
    Given the user is on the login page
    When the user enters the store manager information
    Then the user should be able to login

```

For example we use **@smoke** in two scenarios, both will be get executed.

### How to run all scenarios inside the one feature file ?

```
@login
```

```
Feature: Users should be able to login
```

We put the tag above the **Feature:** keyword to point all scenarios in that feature file.

### RUN A SCENARIO THAT MATCHES BOTH OF TWO TAGS

#### New Syntax:

```
tags = "@driver and @VYT-123"
```

#### Old Syntax:

```
tags = {"@driver", "@VYT-123"}
```

- Run all scenarios that has @driver and @VYT-123

### RUN A SCENARIO THAT MATCHES EITHER OF TWO TAGS

#### New Syntax:

```
tags = "@driver or @store_manager"
```

**Old Syntax:**(Not working on version 5)

```
tags = "@driver, @store_manager"
```

- Run all scenarios that has @driver OR @store\_manager

## **SKIP THE SCENARIO THAT HAS A CERTAIN TAG**

**New Syntax:**

```
tags = "@login and not @wip"
```

**Old Syntax:** (not working with version 5)

```
tags = {"@login", "~@wip"}
```

## **TASK**

Create a new feature file **NavigationMenu.feature** and write scenarios for the given requirement and test cases. Try to reuse steps from previous feature files.

**Requirement:**

Logged in sales managers should be able to navigate to pages using the top menu. Once in the expected page, user should see correct page Title.

**Test cases:**

1.Menu option: **Fleet—> Vehicles**

Expected Title: Vehicles

Test cases:

2.Menu option: **Marketing—> Campaigns**

Expected Title: Campaigns

Test cases:

3.Menu option: **Activities—> Calendar Events**

Expected Title : Calendars

**After** finishing the scenarios, add necessary tags, run through CukesRunner and generate step defs. Cop paste the step definitions from console into new step definitions class called **NavigationMenuStepDefs**. Implement the step definitions by adding print statements.

## Cucumber Hooks

Methods that execute before and after scenarios. Hook will always run, even if the scenario fails.

Create a hooks class

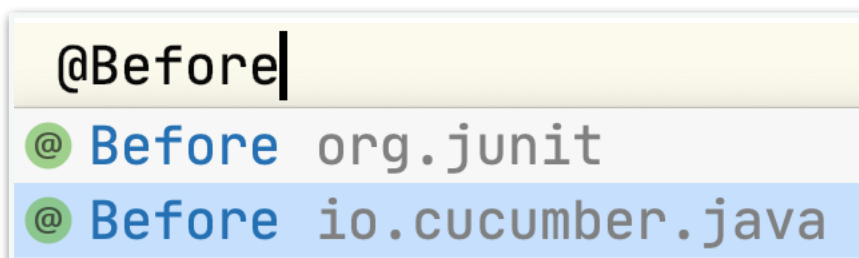
1.Right click on step\_definitions package

2.new-> class

3.Enter name: Hooks

4.OK

Import from cucumber NOT unit



There is no connection between feature files and hooks. Hooks will run always so we need to be careful about what we use inside.

But we have also custom hooks that is running if we have same tag on top of scenarios.

```
@Before("@db")
```

Tagged hooks will only run before/after scenario(s) with that given tag.