We can create a class and define annotations that are common for all the scenarios that would be executed (means before runner class). HOOKS is just an annotations that can be executed for all the scenarios.

@BeforeAll - method that is marked with this annotation will be executed before all the scenarios, and before runner class

@AfterAll - would be executed after all scenarios and runner class

AND important note that the names of the methods should be with STATIC words for example: public static void methodName(){ bla… bla… bla…}

And make sure that you import annotations from cucumber.

**Tags**

Tags serves 2 purposes:

1) To run the subset (подмножества) of scenarios.

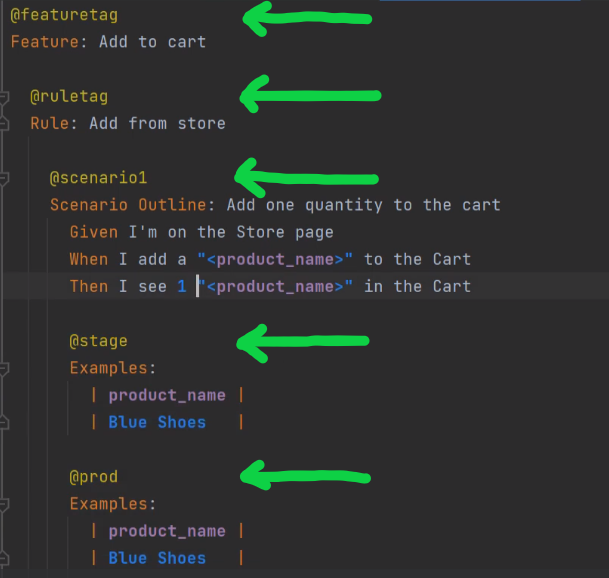
2) To restrict (ограничивать) hooks to a subset of scenarios or to a group of scenarios

We can use it at feature level, rule level, scenario level, Scenario outline level

There is a concept call “T**ag inheritance ''.** It is when we create a tag at the feature level it will execute everything in feature fail (even if other tags are presented - they will be executed as well), when we create a tag at scenario level it will execute that scenario which is marked by this tag(and if theres 2 scenarios in this feature fail, the second one would not be executed)

Scenario outline level. < - - it will be applicable to all the examples only at this level. And we can tag the examples as well to separate them.

We can have as many tags to one step as you want



If you execute scenario with this tag it will execute this part of examples

How do we tell cucumber which tags to use?

1) We can use @cucumberOptions (tags=”name of the tag”)

**Tag expressions**

It is a word that gives even more flexibility and it should be indicated in @cucumberOptions while using the ‘tags’ attribute.

0) Tag Expressions is a simple query language for tags. The simplest tag expression is simply a single tag, for example:

@smoke

1) **or**

@cucumberOptions(tags = “tag#1 **or** tag#2”) <- - during run time will be executed both steps that are marked as tag#1 and tag#2.

2)**And**

@cucumberOptions(tags = “tag#1 **and** tag#2”) <- - during runtime will be executed only the step which has 2 tags (it is possible to give one step more than 1 tag, it is for flexibility).

3)**And not**

@wip

Scenario…

@wip

@slow

Scenario…

So we have 2 different scenarios and both of them have Tag @wip, but only one has @wip and @slow. So if we indicate in the runner in @CucumberOptions(tags=”@wip”). Then both scenarios will be executed.

But if you need to execute only **@wip,** then **@CucumberOptions(tags=”@wip and not @slow”)**

4)**Not**

@CucumberOptions(tags=”not @wip”) <- - everything (all steps) will be executed but not the step with @wip