

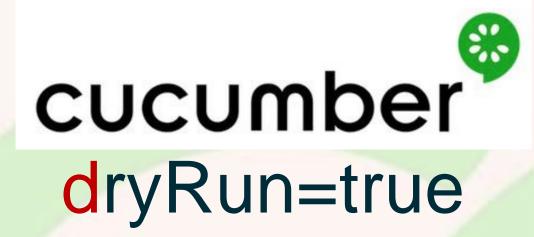
29 ARALIK 2020 DERS 28

Cucumber
@tags, Background
dryRun, Html Reports
Parametre ile kullanma

Mehmet BULUTLUOZ Elektronik Muh.

TestNG Onceki Dersten Aklimizda Kalanlar

- > Cucumber : BDD behaviour driven development (Davranis tabanli kodlama)
- ➤ Teknik olan veya olmayan herkesin test Case'lerimizi anlamasini ve takim olarak calismayi kolaylastirir
- > Testlerimizi olusturmaya baslarken once String olarak adimlarimizi feature dosyasinda olusturup, daha sonra stepdefinitions clasinda yazdigimiz adimlarin kodlarini yaziyoruz
- Feature file da sozlu olarak olusturdugumuz adimlarda Gherkin dilini kullaniyoruz. Burada kullanabilecegimiz temel kelimeler Given, When, And, Then dir. Bu kelimelerin kod acisindan hic bir farki yoktur ancak test adimlarimizi okuyan kisilerin anlamasi acisindan bu kelimeleri dogru sekilde kullanmaliyiz.
- Projemizi olusturmak icin pom xml'e Selenium-java, webdrivermanager, cucumber java, cucumber junit dependency'lerini ekliyoruz
- ➤ Bir test olusturmak icin once fature file'da sozel olarak adimlarimizi olusturuyoruz, runner'i calistirip onun olusturdugu adimlari stepdefinitions'a kopyalayip orada bu methodlarin iclerine gerekli kodlari yaziyoruz ve runner dosyasindan calistiriyoruz



- dryRun=true test yürütülmeden çalışır.
- ➤ Temel amaç, adımları uygulamadan eksik adımların(missing step definitions) oluşturulmasıdır. Bunu, özellik dosyamızda uygulanmayan herhangi bir adım olup olmadığını kontrol etmek için kullanabiliriz.
- ➤ Yürütmeyi önlemek için dryRun = true ayarlıyoruz ve yalnızca uygulanmayan feature için method üretiyoruz.
- > By default(default olarak), dryRun=false.

```
@CucumberOptions(
Features = "src/test/resources/features",
glue = "stepdefinitions"
dryRun = false
)
```



Farklı senaryoların başında ortak adımlarımız varsa:

- 1. Feature file in basina Background oluşturun.
- 2. Bu ortak adımları Background altına yazın.
- 3. Background, aynı Feature file'daki her Scenario'dan önce çalışır
- 4. Duplication olmadigindan emin olun. Background un altındaki adımı yazdıktan sonra senaryolardan silin.

```
Feature: First Feature
                                                             This two step will run
                                                             before each scenario
  Background: User search for amazon on hte google page
    Given user is on the google page
                                                            In this example there are
    And user search for amazon
                                                             two Scenario: keyword
                                                              So Background will
                                                                  run twice
  agooglesearch
  Scenario: TC02_Google search test
    Then user should should see amazon link on the search result
    Then clear the search box
    #Scenario 2 Search for flower and check if the the page related images
    Given user search for flowers on google page
    Then verify the page has flower
  agooglesearch
 Scenario: TC04_Google search test 2
    When user click on the first link
   Then user verify the amazon page displays
```

Class Work: US1001 feature file'daki tekrar eden aramalar yerine parameter kullanarak arama yapabilecegimiz sekilde US1002 feature file veTC02 parameter ile arama Scenerio'su olusturalim



- ➤ Tag lari yalnızca belirli senaryoları (scenario) çalıştırmak için kullanırız.
- Tag lari senaryolarımızı gruplandırmak için kullanabiliriz (smoke test, regression test, vs.)

```
@CucumberOptions(
features = "src/test/resources/features",
glue = "stepdefinitions",
dryRun = false,
tags="@amazon",
)
```

Gamazon @search @apple
Scenario: TC01 amazon arama testi

Given kullanici amazon sayfasina gider
And "apple" icin arama yapar
Then sonuclarin "apple" icerdigini test eder
Then sayfayi kapatir



- Cucumber raporları, şirketlerin Cucumber kullanmasının ana nedenlerinden biridir.
- Html rapor almak icin runner classına eklenti(plugin) eklememiz yeterlidir.

```
@CucumberOptions(
plugin={"html:target/cucumber_rapor.html"},
features = "src/test/resources/features",
glue = "stepdefinitions",
dryRun = false,
tags="@amazon"
```



Feature File'i Parametre ile Kullanma

- ➤ Parametreleştirmek ve dinamik hale getirmek için feature file da çift tırnak " " kullanırız.
- > Şimdi sadece " " içindeki değeri değiştirerek test datalarini feature file dan kontrol edebiliriz.
- ➤ Bu framework'u daha dinamik hale getirir, yani kodumuz artik hard coded degildir diyebiliriz.

Class Work: US1001 de kullandigimiz feature dosyasi altinda yeni bir Scenario olusturalim TC03 ve orada yaptigimiz aramayi parametre kullanarak yapalim

Scenario: TC01 iphone aramasi testi
Given kullanici amazon sayfasina gider
And iphone icin arama yapar
Then sonuclarin iphone icerdigini test eder

Scenario: TC02 tea pot aramasi testi
Given kullanici amazon sayfasina gider
And tea pot icin arama yapar
Then sonuclarin tea pot icerdigini test eder

Scenario: TC03 flower aramasi testi
Given kullanici amazon sayfasina gider
And flower icin arama yapar
Then sonuclarin flower icerdigini test eder
Then sayfayi kapatir

Feature: US1000 Amazon search test

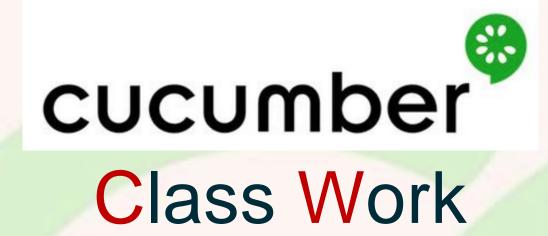
Scenario: TCO1 iphone aramasi testi
Given kullanici amazon sayfasina gider
And "iphone" icin arama yapar
Then sonuclarin "iphone" icerdigini test eder
Then sayfayi kapatir



Feature File'i Parametre ile Kullanma

- Scenario Outline: ayni testte birden fazla datayi kullanmamizi saglar
- Bir liste kullanmak istedigimiz degeri "<value>" seklinde yazariz
- Daha sonra testin sonuna Examples: yazip ilk satir olarak | value | yazariz ve altina kullanmak istedigimiz degerleri ekleriz. (|elma|,|armut|... gibi)

Class Work: US1001 de kullandigimiz feature dosyasi altinda yeni bir Scenario olusturalim TC04 ve orada yaptigimiz aramayi Scenario Outline kullanarak farkli urunler icin yapalim



Yeni bir feature file olusturalim: US1002_ConfigReaderdan_parameter_alma.feature

Yeni bir Scenario olusturalim: TC05 Configreader ile farkli parametreler kullanma

Configuration.properties dosyasinda olusturacagim farkli web sitesi url'lerini okuyup o sayfalara gidecek ve sayfa url'lerini test edecek stepDefinitions'l olusturup runner ile kodumuzu calistiralim

```
Scenario Outline: TC01 amazon arama testi

Given kullanici "<sayfa_url>" sayfasina gider
And url'in "<sayfa_url>" oldugunu test eder
Then sayfayi kapatir

Examples:
|sayfa_url|
|amazon_url|
|bestbuy_url|
|ebay_url |
```