28 ARALIK 2020
DERS 27

Cucumber
Ilk Proje Olusturma

Mehmet BULUTLUOZ
Elektronik Muh.

- Cucumber bizim son framework'umuz olacak.
- Cucumber BDD(behaviour driven development) (Davranış güdümlü geliştirme) yaklaşımı için kullanılmakta olan açık kaynak kodlu bir kütüphanedir.
- Cucumber bir iş ararken önemli bir rol alacaktir.

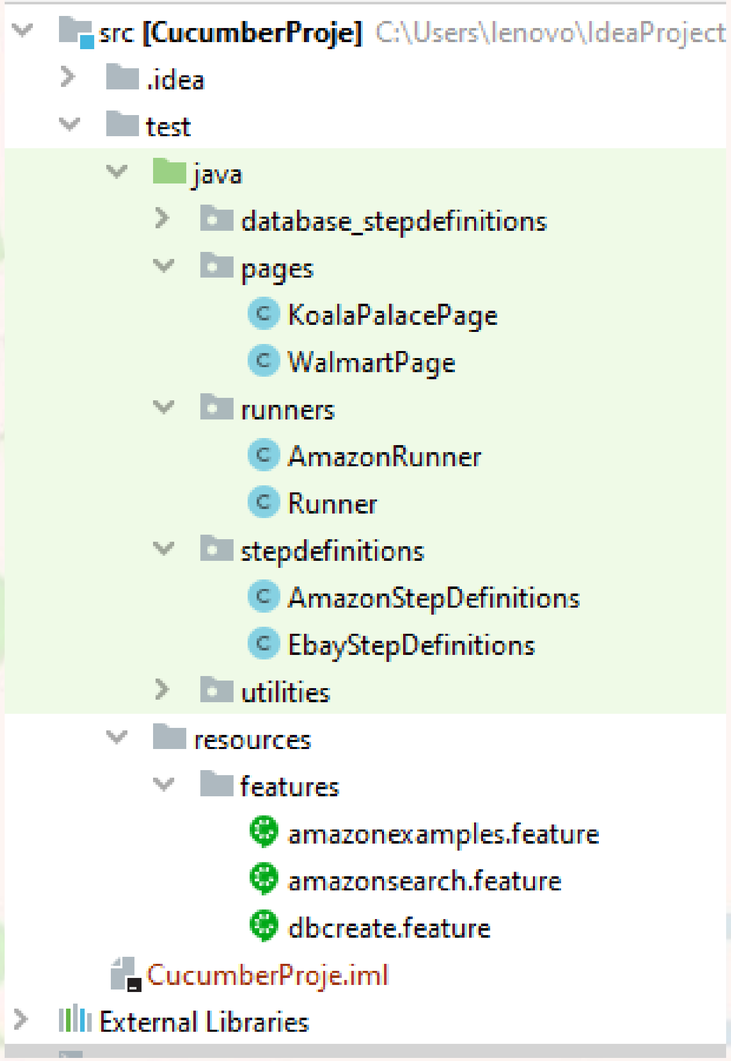
```
Feature: US1001 Ck Hotels Login
  @wip
  Scenario: TC01 kullanici gecerli bilgilerle giris yapar
    Given kullanici Ck Hotels ana sayfasinda
    Then Log in yazisina tiklar
    And gecerli username girer
    And gecerli password girer
    And Login butonuna basar
    Then sayfaya giris yaptigini kontrol eder
    And kulllanici sayfayi kapatir
```

- TestNg hakkında her şeyi bilmemek sorun değil ama Cucumber hakkında her şeyi bilmeniz GEREKİR.
- Agile methodolojisinde, insanlar uygulamanın işlevselliğini geliştirmek için birlikte çalışmak zorundadır. İnsanlar development sırasında birlikte hareket etmeli.

- BDD(behaviour driven development) (Davranış güdümlü geliştirme)- ilk olarak behavior(davranis) veya functionalitileri yaziyorsunuz (Epic=Feature, Story, AC, etc), daha sonra development and testing basliyor.
- BDD'de behaviour'lar başarısız olduğunda kod başarısız olur..
- Anlasilabilir Gherkin Language nedeniyle BDD development icin Cucumber harika bir uygulamadir.
- Gherkin:Projede her bir davranış için .feature uzantılı bir Gherkin dosyası oluşturulur. Bu feature dosyasına ilgili özelliğin farklı durumlardaki davranışları tanımlanır. Given anahtar kelimesi ile ön koşul yani başlangıç durumu tanımlanır, When, And anahtar kelimeleri ile olayı Then anahtar kelimesi ile de sonuç tanımlanır.

  Given-When-And-Then adımları ile Scenario oluşturulur.

- Cucumber (TDD)(Test Driven Development) test odaklı geliştirmeye izin verir, çünkü Cucumber ile Junit veya TestNG kullanabiliriz.
- Cucumber iş için önemlidir, çünkü anlaşılabilir ve harika raporlara sahiptir.
- Cucumber, teknik olmayan(none-technical) kişileri ve teknik kişileri birbirine bağlar.
- Developer veya team lead gibi teknik elemanlar da bazen testerlarin yaptığını anlayamayabilir. Gherkin onlarin da testlerimizi anlamalarini kolaylastirir



Proje Olusturma

1. Create Project: File -> New -> Project-> Select maven -> click next
2. Name: mycucumberframework->finish
3. Add Dependencies =>Selenium-java, webdrivermanager, cucumber java, cucumber junit
4. Click Maven => click “Enable auto-reload after any changes”
5. Javanin sürümüyle alakali sorunlari halletmek icin compiler dependency yuklenebilir

```
<properties>
  <maven.compiler.source>1.8</maven.compiler.source>
  <maven.compiler.target>1.8</maven.compiler.target>
</properties>
```

Proje Olusturma

6. Java'ya sag click yapip asagidaki paketleri olusturalim
    a. utilities
    b. pages
    c. runners (test case'leri calistirmak ve control etmek icin kullanacagiz)
    d. stepdefinitions (kodlarimizi burada olusturacagiz)
7- Utilities paketi altinda Driver ve ConfigReader Class'larini olusturalim
8- Projeye sag click yapip configuration.properties dosyasi olusturalim
9- test paketi altinda yeni bir klasor olusturalim : resources
10- resources klasoru altinda yeni bir klasor olusturalim : features
    (Java kodu icermeyen dosyalari buraya koyacagiz)
11- features'a sag clik yapip dosya olusturalim firstfeaturefile.feature
12- cucumber for Java plugin'i intelliJ'e ekleyelim (seetings/Plugins)
    MAC => IntelliJ Idea->Preference->Plugins->Marketplace->Type Cucumber forJava ->Install->Restart

Class Work : First Cucumber Test Case

- Yeni bir feature file olusturalim : amazonsearch.feature

```
Given kullanici amazon sayfasina gider
And iPhone icin arama yapar
Then sonuclarin Iphone icerdigini test eder

Given kullanici amazon sayfasina gider
And tea pot icin arama yapar
Then sonuclarin tea pot icerdigini test eder

Given kullanici amazon sayfasina gider
And flower icin arama yapar
Then sonuclarin flower icerdigini test eder
```

Class Work : First Cucumber Test Case

Feature File: Yeni bir feature file olusturalim : amazonsearch.feature , Test Case'i Gherken language kullanarak yazalim

stepdefinitions package: FirstFeatureFileStepDefinitions Class'ini olusturalim

runner package :

Runner class'i olusturalim (Runner class'i bir kez olusturuyoruz)

```
@RunWith(Cucumber.class)
@CucumberOptions(
features="features folder path"
glue="stepdefinitions folder path"
tags="@istediginiz tag"
dryRun=false)
```

Runner class'i calistirip step definitions'i olusturun ve iclerine kodlari yazin

Feature File

- En önemli dosya ve cucumber framework'unun ana nedenlerinden biridir.
- **Feature file** test caselerin/scenario adım adım yazildigi yerdir.
- Davranışlari(behaviours) bu dosyaya adım adım yazıyoruz.
- Feature file **Gherkin** dilinde yazılmıştır
  - Adimlarin olusturulmasi -> **Given, When, Then, And,**
  - sırayla kullanmak önemli mi? Hayır.

  Anlaşılabilir ve okunabilir olması önemlidir.

  "" => to parameterize, | => for data tables, @ => for tags, # => to comment like //
- Feature file yazdığımızda, bu adımı başka bir feature file'da her zaman kullanabiliriz
- İyi bir tester, diğer test caselerde de kullanılabilecek(reusable) faydalı adımları yazar.
- Yeni bir Scenario veya Feature file yazdığımızda, daha önce oluşturulan adımı yeniden kullanabilmemiz için aynı syntaxi kullanmalıyız. Aynı syntax demek, adımın harf harf eşleşmesi gerektiği anlamına gelir.

  Eşleşen bir adımımız olduğu sürece, cucumber step definitioni çalıştıracaktır. (stepdefinitions package)