

- Test çalıştırmalarımızın durumunu görmek için konsol veya html raporuna bakıyoruz. Bir şey başarısız olursa, oradan buluruz.
- Çerçevemizde oturum açmayı uygularsak, otomasyon yürütme adımlarına bakmanın başka bir yolu olacak ve testimiz başarısız olduğunda sorunu bulmamıza yardımcı olun

- 75 -

Sayfa 76

TESTNG & JUNIT

1. TestNG nedir?

- 500 test vakanız var → Her test vakası için bir Java Paketi ve 500 Sınıfı oluşturuyoruz
Müşteri sizden sadece 40 tanesini duman testi için çalıştırmanızı istedi → Blokları ve raporlama mekanizmasıyla Jasmine'de hallediyoruz.
- TestNG bir test çerçevesidir
- Merkezi kontrolör: farklı test senaryolarının çalıştırılmasını yönetin ve ardından raporlar, günlükler oluşturun
- Toplu yürütme: 100 test durumu ve bunları birer birer çalıştırın
- İsteğe bağlı yürütme: bazı test durumlarını atlayabiliriz

2. TestNG'deki iddialar nelerdir?

- Testi çalıştırdık ve başlık testi durumu başarısız oldu. Diğer test durumlarını etkilemeyeceği için betiğimizin durmasını istemiyoruz.
 - o Kritik → dur / başarısız İddiası
 - Bir boole bağımsız değişkeni ve String mesajı alır. Bir koşulun doğru olduğunu iddia eder. Değilse, bir Onay Hatası, verilen mesajla birlikte atılır.
 - o Kritik değil → hata / devam SoftAssert
 - Soft Assert, bir iddia başarısız olduğunda bir istisna atmaz ve sonraki adımla devam eder. ifade iddia.

3. JUnit ve TestNG arasındaki fark

- Ek açıklamalar; **JUnit:** @Test, @BeforeClass, @AfterClass, @Before, @After, @Ignore
TestNG : @Test, @BeforeTest, @BeforeClass, @BeforeSuite, @BeforeMethod, @AfterTest, @AfterClass, @AfterSuite, @AfterMethod
- Her ikisi de bize yardımcı olmak için çerçeve test ediyor
otomasyon komut dosyalarını çalıştırma.
- TestNG, html raporu sağlar
- TestNG vardır @Hayalhanemersin
ek açıklama Salatalık ile aynı
Veriye Dayalı Senaryo Özeti
Test yapmak.
- TestNG'de paralel testler yapabiliriz,
ancak JUnit paralel olmayı desteklemiyor
testi, bu yüzden bunun için SauLab kullanıyoruz.
- TestNG destek grubu testi ancak JUnit desteklemiyor

- TestNG ve JUnit her ikisinde de

testi parametreleştirmek ancak TestNG
parametrelili test yapılandırması
yapılandırması çok kolay. İki tane
parametreleştirme elde etmenin yolları
TestNG;
o @Parameters ve TestNG xml dosyası
o @DataProvider

- 76 -

Sayfa 77

4. Çapraz Tarayıcı ve Paralel Test

- Mevcut projemde, çapraz tarayıcı testi için SauLab kullanıyoruz. Ama önceki projemde testng.xml dosyasını kullandım.
- Temel olarak, süitin içinde 3 anahtar var (ad, iş parçacığı sayısı, paralel) ve bunlardan biri için 2 farklı test oluşturdum.
- Chrome ve diğeri Firefox içindir.
- Parametre açıklaması da vardır ve adı ve değeri içerir; ad tarayıcıdır ve değeri Chrome'dur.

```
<? xml version = "1.0" encoding = "UTF 8" ?>
<! DOCTYPE suite SİSTEMİ ...>
< süit ... >
  < test name = "ChromeTest" ... >
    < parametre adı = "tarayıcı" value = "chrome" />
    < sınıflar >
      < class name = "testsuite ..." />
    </ sınıflar >
  </ test > <! İlk Test>
  < test adı = "FireFox" ... >
    < parametre adı = "tarayıcı" value = "FireFox" />
    < sınıflar >
      < class name = "testsuite..." />
    </ sınıflar >
  </ test > <! İkinci Test>
</ süit > <! Süit>
```