



GUIDERSOFT

SELENIUM 09

21.07.2022

Junit GIRIS ANNOTATIONS
ASERTIONS

JUnit

- ❖ Java ile en temel framework JUnit ile oluşturulabilir.
- ❖ Developerlar da unit testleri calistirmek icin kullanirlar.
- ❖ Biz testlerimizi yapmak icin JUnit'in ileri surumu oldugundan TestNG framework olusturup kullanacagiz.
- ❖ Junit maven projesi uzerinde calistigindan <https://mvnrepository.com/> sitesinden dependency'leri projemize ekleriz.
- ❖ Test'lerimizi yaparken main method, if-else gibi java kod bloklarini kullanmak yerine Junit annotationlarini ve method'larine kullanabiliriz.
- ❖ Son framework olarak kuracagimiz Cucumber framework'da da Junit kullanacagiz





Annotations

- ❖ Selenium'da kodlarımızı yazarken "@" işareti ile başlayan notasyonlar kullanırız.
- ❖ Java Annotationlar ile derleyiciye (Compiler) talimatlar verebiliriz.
- ❖ Annotation, bir veri hakkında bilgi barındıran veriyi sağlayan basit bir yapıdır. Bu sağladığı bilgiye de "metadata" denir.
- ❖ Notasyonlar(Annotation) genellikle Java'da konfigürasyon amacıyla kullanılır. Kullanıldığı bileşene ek özellikler katar. Bu bileşenler sınıf, metod, değişkenler, paket ya da parametreler olabilir. Bunların hepsinde notasyonları kullanabiliriz.

En çok kullanılan Junit annotation'ları

@Test

@BeforeClass @AfterClass

@Before , @After

@Ignore



ANNOTATIONS

@Test ve @Ignore

- ❖ Junit ile Main Method kullanma donemini bitiriyoruz.
- ❖ Junit Framework kullandigimizda yazdigimiz test metodunun calismasi icin başına @Test notasyonu eklememiz yeterlidir.
- ❖ @Test notasyonu eklemedigimiz metot test sırasında çalıştırılmaz. Ancak cagrilirsa calisir.
- ❖ Yazdigimiz bazı test metotları henuz tamamlanmamış veya değişiklikleri uğrayabileceğinden dolayı test sırasında çalıştırılmasını istemiyorsak @Ignore notasyonu eklememiz yeterlidir.
- ❖ @Ignore notasyonun tanımlı olduğu metotlar test sırasında çalıştırılmayacaktır. Ayrıca istenilirse @Ignore("açıklama") şeklinde yazılarak metodun neden test edilmesini istemediğimizide yazabiliriz.



ANNOTATIONS

@Before ve @After

- ❖ Before notasyonu, her test method'undan önce çalışır. Örneğin bir sayfa ile test yapıyorsak ve her testten önce o sayfaya gitmemiz gerekiyorsa @before kullanabiliriz.

@before notasyonunun kullanıldığı method'a genelde setup() ismi verilir.

- ❖ After notasyonu, her test method'undan sonra çalışmaktadır. Örneğin test sırasında kullandığımız sayfanın kapatılması gibi.

@after notasyonunun kullanıldığı metoda genelde teardown() ismi verilir

Not : Test method'u ve Test farklı yapılardır.

Test dedigimizde tek bir method veya içinde birçok class ve package barındıran bir yapı olabilir. Regression test, smoke test vb..

Test method'u ise @Test notasyonu kullanılarak oluşturulan ve bağımsız olarak çalıştırabileceğimiz en küçük test yapısıdır



ANNOTATIONS

@BeforeClass ve @AfterClass

- ❖ BeforeClass notasyonu, bir class'daki tüm testlerden önce yapılması gereken bir işlem varsa kullanılır (precondition).

Örneğin test metodlarımız çalışmadan driver oluşturup tüm methodlarda kullanabilirim.

- ❖ AfterClass notasyonu da, bir class'daki tüm testler tamamlandıktan sonra yapılması gereken işlemlerde kullanılır.

Örneğin açtığımız sayfayı kapatmak veya elde ettiğimiz test sonuçlarını raporlamak gibi.

- ❖ @BeforeClass ve @AfterClass notasyonları test sürecinde bir kere çalışırken, @Before ve @After notasyonları her test method'unun başında ve sonunda çalışmaktadır.