

Unit (Birim) Test Nedir, Nasıl Kullanılır ve JUnit5 ile Java Örneği

Genel Bilgiler

Unit test genel olarak; bir metodun (ya da fonksiyonun) kendi içerisindeki akışın doğru çalışıp çalışmadığı bilgisini developer' a gösteren bir testtir. Bu testi son kullanıcı testleri ile ayrı tutmamız lazım. Aslında ifade etmeye çalıştığım şey: bu test developer testidir. Kod doğru çalışıyor mu diye developer' in geliştirme esnasında oluşturduğu ve kodunu kontrol ettiği testtir. Yani kısacası monolitik olarak metod bazında yapılıdır.

Unit testler success olduğunda yapılan işlem son kullanıcı açısından sorunsuz anlama gelmemeli. Belki domainsel dediğimiz analiz sorunları vb. sorunlar oluşabilir.

Özetle Unit testin amacı teknik olarak metod(ya da fonksiyonun) herhangi bir exception oluşturmada hatasız çalışıp, beklenen girdilere karşın beklenen sonucu verdi mi? sorunsalı için yazılır.

Java da en çok kullanılan JUnit kütüphanesi kendi içerisinde JUnit4 ve JUnit5 olarak aktif kullanılır durumda. Ben bu yazımda JUnit5 kullanacağım. Başka bir yazımda da java da çok popüler olan başka bir unit test kütüphanesi olan mockito' dan bahsedeceğim ☺

O zaman başlıyoruz.

JUnit5 Annotationlar (Anotasyonlar)

@Disable: Test sınıfını ya da metodunu devre dışı bırakmak için kullanılır. (JUnit4'te @Ignore)

@AfterAll: Tüm test metodları çalıştırdıktan sonra çalıştırılacak olan metodu belirtir.

@BeforeAll: Tüm test metodlarından önce çalıştırılacak olan metodu belirtir.

@AfterEach: Her test metodundan sonra çalıştırılacak olan metodu belirtir. (JUnit4'te @After)

@BeforeEach: Her test metodundan önce çalıştırılacak olan metodu belirtir. (JUnit4'te @Before)

@ExtendWith: Custom extension'ları eklemek için kullanılır. (Mesela spring boot extension' u için: @ExtendWith(SpringExtension.class))

@Tag: Testleri filtrelemek için Tag ekler.

@DisplayName: Sınıf ya da metodların isimlerinin test sonuçlarında nasıl görüneceğini belirler.

@Nested: Bu annotationu alan sınıfın bir nested sınıf yani non-static sınıf olduğunu belirtir.

@TestFactory: Metodun dinamik test olduğunu belirtir.

JUnit5 İin Yazılımsal Giriş ve Geliştirme

Bilgi:

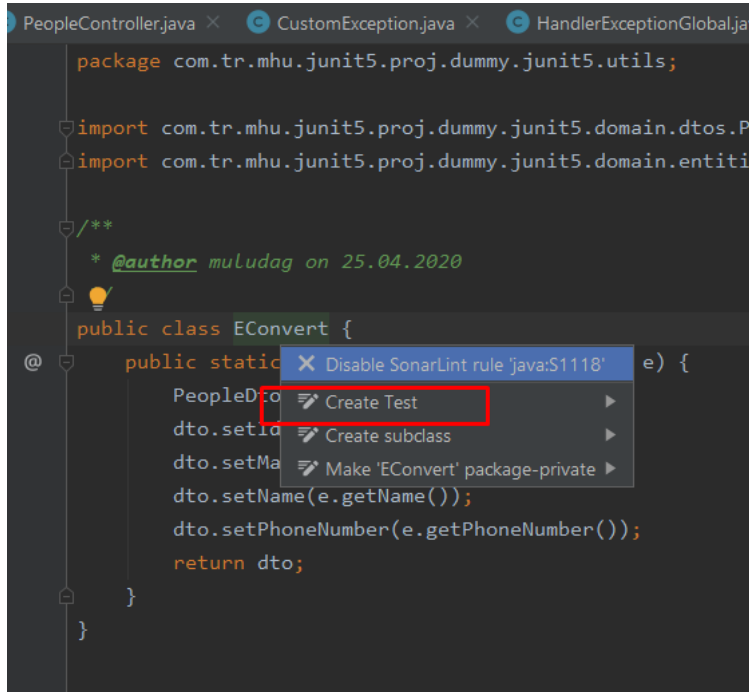
Yazılım geliştirme IDE' si olarak IntelliJ IDEA, development kit olarak JDK14.1 Release kullanılmaktadır. Proje sonlarında uygulama linki açık kaynak kodlu olarak paylaşılacaktır.

pom.xml' e junit5 için aşağıdaki dependency ekleyiniz:

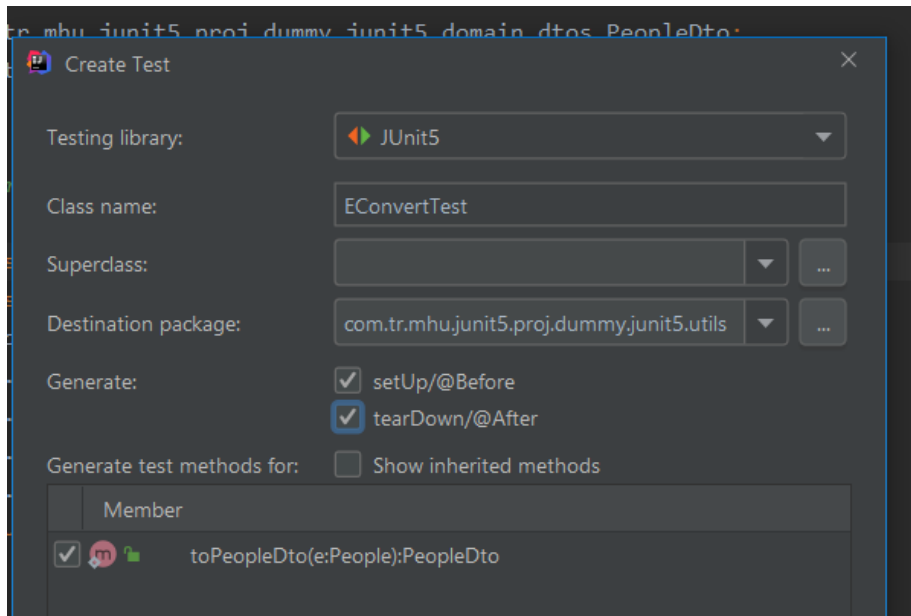
İsterseniz IntelliJ IDEA' ya da ekletebilirsiniz. Aşağıda bahsediyor olacağım [A]

```
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <scope>test</scope>
</dependency>
```

IntelliJ Otomatik Test Class' ı Oluşturmak İin:



Ben IntelliJ IDEA kullanırken keymap olarak Eclipse' nin keymap' ini kullanmaktayım. Herhangi bir class içerisinde Alt+Enter basıldığında aşağıdaki gibi **Create Test** alanına tıklayarak açılan pencere içerisinde kullanacağımız metotları ve unit test yöntemlerini seçerek otomatik olarak IDE' nin test package' ne test class' ını generate edelim.



Bilgi: [A]

Yukarıda paylaştığım pom' bilgisine JUnit5 eklemediyseniz burada JUnit5 seçtiğinizde Fix ederek IDE' ye otomatikte ekletebilirsiniz.

Class içerisinde test edeceğim metot/metotları seçerek **setUp** ve **tearDown** metotlarını da eklemek istedim. Bu metotlar yukarıda bahsettiğim anotasyonlar başlığında **BeforeEach** ve **AfterEach** işlevlerini yerine getirmek için kullanılır.

```
andlerExceptionGlobal.java x IPeopleS
* @author muludag on 25.04.2020
class EConvertTest {
    @BeforeEach
    void setUp() {
    }
    @AfterEach
    void tearDown() {
    }
    @Test
    void toPeopleDto() {
    }
}
```

Yanda görüldüğü gibi test class' im oluşturdum.

toPeopleDto isimli metodum için bir test yazalım.

Hikaye: (isteğe bağlı genel kültür bilgisi içerir)

Şimdi bu metodun amacı People adında bir entity' i response öncesinde dto olarak convert etmek için kullanılıyor. Çünkü bir api projesinde entity dönmek doğru olmaz. Dto (data transfer object) olarak bilgi girişi ve çıkışı yapmak gerekir. Çünkü entity class' lar database ile property olarak farklı yorumlanır. Spesifik kullanımlar entity class' lara has olmalıdır. CRUD vb işlemlerde Dto kullanmamız bussines olarakta doğru yöntemdir. Peki entity class lar kullanılmaz mı? kullanılır ama doğru bir yöntem olmaz. Aslında MVC katmanlı mimariye de ters bir durum oluşturulur. SOLID falan kalmaz ortalıkta. Her yer kangölü olabilir. Plase clean code...

Neyse devam ediyoruz.

Aşağıdaki resimde görüldüğü üzere kullandığım Lombok kütüphanesi içerisinde gömülü gelenSlf4j log kütüphanesini class in en üstüne ekleyerek log' lamayı sağladım.

Sahte bir entity oluşturarak çok çok çok basit bir şekilde bu entity' nin dto ya çevrilmesi işini yaptık.

```
@Slf4j
class EConvertTest {
    People entity;

    @BeforeEach
    void setUp() {
        this.entity = people();
        log.debug("test started...");
    }

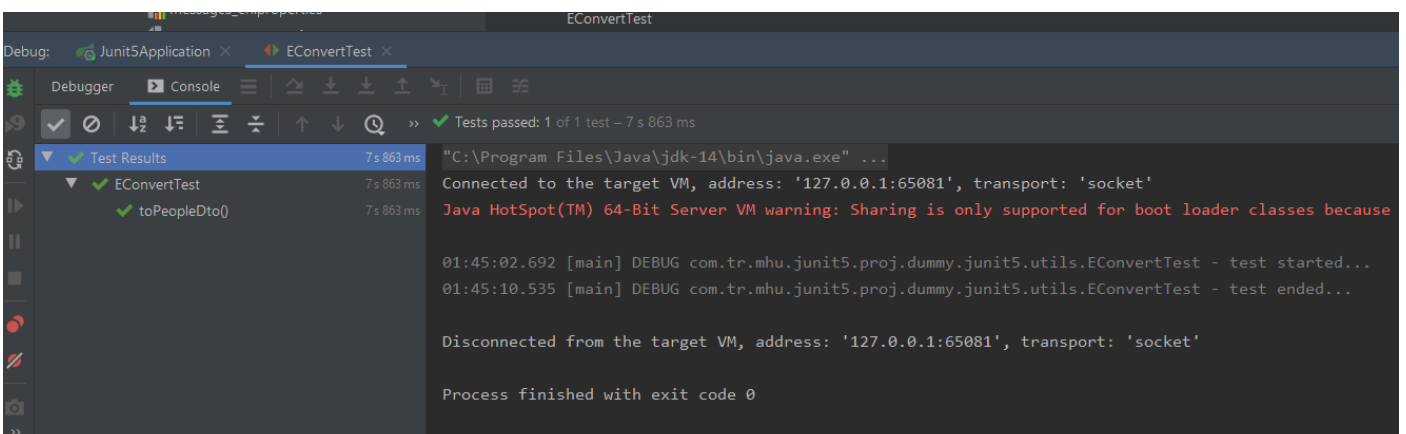
    @AfterEach
    void tearDown() {
        this.entity = new People();
        log.debug("test ended...");
    }

    @Test
    void toPeopleDto() {
        PeopleDto dto = peopleDto(this.entity);
        Assert.assertNotNull(dto);
        Assert.assertFalse(message: "There is a problem!", dto.getId().isBlank());
    }

    @PeopleDto peopleDto(People e) {
        return new PeopleDto(e.getId(), e.getName(), e.getMail(), e.getPhoneNumber());
    }

    People people() {
        People e = new People();
        e.setId("1010101");
        e.setName("dummy");
        e.setMail("dummy@mail.com");
        e.setPhoneNumber("1010101");
        return e;
    }
}
```

Test class'ımız yapısı gereği runnable haldedir. Kırmızı kareye basılırsa @Test anotasyonuna sahip her metodu teker teker işletim için sırasıyla çalıştırır. Sarı kareye basıldığında ilgili anotasyona sahip metod işletilir.



```
Debug: JUnit5Application x EConvertTest x
Debugger Console
Tests passed: 1 of 1 test - 7 s 863 ms
Test Results
  EConvertTest
    toPeopleDto()
      7 s 863 ms
      7 s 863 ms
      7 s 863 ms
"C:\Program Files\Java\jdk-14\bin\java.exe" ...
Connected to the target VM, address: '127.0.0.1:65081', transport: 'socket'
Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader classes because
01:45:02.692 [main] DEBUG com.tr.mhu.junit5.proj.dummy.junit5.utils.EConvertTest - test started...
01:45:10.535 [main] DEBUG com.tr.mhu.junit5.proj.dummy.junit5.utils.EConvertTest - test ended...
Disconnected from the target VM, address: '127.0.0.1:65081', transport: 'socket'
Process finished with exit code 0
```

Örnek basit oldu bir örnek daha yapalım. EConvert class' ına list dönen bir metod daha ekledim. Bunu da test edelim.

```
public class EConvert {
    public static PeopleDto toPeopleDto(People e) {
        return new PeopleDto(e.getId(), e.getName(), e.getMail(), e.getPhoneNumber());
    }

    public static List<PeopleDto> list() {
        return List.of(new PeopleDto(id: "1", name: "", mail: "", phoneNumber: ""), new PeopleDto(id: "1", name: "", mail: "", phoneNumber: ""));
    }
}
```

Şimdi listSize adında bir int değişken oluşturup bunu setup' ta setleyelim.

```
@Slf4j
class EConvertTest {
    People entity;
    static int listSize;

    @BeforeEach
    void setUp() {
        this.entity = people();
        log.debug("test started...");
        listSize=10;
    }
}
```

Dummy daha oluşturup listSize kadar geriye List<PeopleDto> dönen yeni bir test metodu yazalım...

```
55 @Test
56 void list() {
57     List<PeopleDto> peopleDtoList = Collections.nCopies(listSize, peopleDto(this.entity));
58     //exp
59     Assert.assertNotNull(peopleDtoList);
60     Assert.assertTrue( condition: peopleDtoList.size() > 0);
61     Assert.assertEquals(peopleDtoList.size(), listSize);
62 }
63
```

listSize kadar dto objesini klonladık. Daha sonra önce null kontrolü sonra oluşan list' imizin 0 dan büyük kontrolü sonra da klon esnasında verdiğimiz listSize kadar olup olmadığı kontrolü yaptık.

Bir sonraki makale de unit test dünyasında popüler olan mockito' dan bahsedeceğim. Orada katmanlı bir mimarisi içerisinde detaylı birim testler yazacağız. Controller, service ve repository gibi katmanlarda mock (sahte) obje ile veritabanına fake crud işlemleri yapacağız.

Anlatılan bu uygulamayı indirmek için [tıklayınız](#).

Selam ve saygılarımla,

www.melihhilmiuludag.com

Tüm hakları Melih Hilmi ULUDAĞ' a aittir.