

## Unit Test Nedir ?

Unit test uygulamamızdaki en küçük parçanın tek başına test edilmesidir. Amaç basittir her bir parça kendi görevini yerine getiriyor mu onu test etmektir. Unit test yazılımcı tarafından yapılır. Unit test yazarken belli kurallara uyulmalı.

Kuralları anlatmaya başlamadan önce bir örnek verip onun testi üzerinden devam etmek istiyorum.

User adında bir modelim var.İçinde sadece name,surname ve email alanlarını tutuyorum.UserService'i oluşturup içine basit CRUD işlemlerini koyuyorum. Service kısmının testini yazmak istiyorum.

Unit test kurallarını örnekler üzerinden anlatmaya çalışayım.

1. En temel kural en küçük parça test edilmeli ve test edilirken bir test methodu içinde iki işi birden yapayım da ikisini tek test methodunda yapıp kurtalayım diyemeyiz. Örneğin

```
@Test
public void scenario() {
    UserDTO user = new UserDTO();
    user.setId(1L);
    user.setName("Halime");
    user.setSurname("Öztürk");
    user.setEmail("ozturkk.halimee@gmail.com");
    user.setCreationDate(ZonedDateTime.now());
    user.setModificationDate(ZonedDateTime.now());
    userService.create(user);
    userService.getUserById(1L);
    verify(userService).getUserById(2L);
    verify(userService).create(user);
}
```

Burada create ve getUserById methodunun çağırıyorum. Test ederken önce id si 2 olan bir user var mı onu test etmek istiyorum. Id si 2 olan user olmadığı için testim patlar ve aşağıdaki create testi hiç çalıştırılmaz. Bu yüzden bütün parçalar ayrı ayrı test methodlarında test edilmelidir.

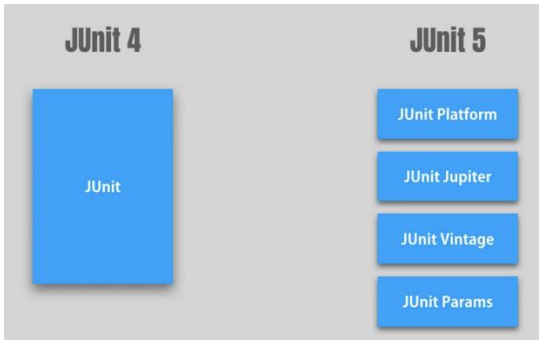
2. Bir test methodu içinde yalnızca bir senaryosu test etmeliyiz. Aşağıdaki gibi olmamalı.

```
@Test
public void scenario2() {
    UserDTO user = new UserDTO();
    user.setId(1L);
    user.setName("Halime");
    user.setSurname("Öztürk");
    user.setEmail("ozturkk.halimee@gmail.com");
    userService.create(user);
    UserDTO test = new UserDTO();
    test.setId(2L);
    test.setName("Test");
    test.setSurname("Test");
    test.setEmail("test@gmail.com");
    userService.create(test);
    verify(userService).create(user);
}
```

3. Test methodu diğer test methodlarından bağımsız bir şekilde çalışabilmeli.(Mock kavramı ileride açıklanacak)
4. Her test farklı bir sıraya göre çalışabilmeli.
5. Her zaman aynı sonucu vermeli.

## JUnit Test

Javada unit test yapabilmemizi sağlar. JUnit4 ve JUnit5 kavramlarının farklarını ve bazı keywordlerini anlatmaya çalışacağım. JUnit5 modüler bir yapıya bürünmüştür. JUnit4 tek bir .jar dosyasından oluşur. JUnit4 de kullanmadığım modüller bir paket halinde geliyordu ama JUnit5 de sadece ihtiyacımız olan modülleri projemize eklemek yeterli olacaktır.



En temel modül JUnit Platformdur. Görevi testlerin JVM de çalışmasını sağlamak.

JUnit Jupiter programlama ve extension modelini ekler.

Bu iki modül projenizde olması gerekli.

JUnit Vintage JUnit3 ve JUnit4 ile çalışabilmemizi sağlar. Projenizde eskiden yazılmış bir JUnit4 varsa ve siz projenizdeki testleri JUnit5 yapmak istiyorsanız eski testlerinizin bozulmadan çalışmasını sağlayıp yeni yazılacak testlerinizde JUnit5 ile yazabilirsiniz.

JUnit Params parametre kullanarak test yazma olanağı sağlar.

JUnit 5 de bazı anatastionlar degisti.

JUnit4	JUnit5
BeforeClass	BeforeAll
AfterClass	AfterAll
Before	BeforeEach
After	AfterEach
Ignore	Disable
Category	Tag

JUnit4 deki yaşam döngüsünden bahsedeceğim aynı yaşam döngüsü JUnit5 içinde geçerli.

Test başladıktan sonra BeforeClass çağrılır her bir sınıf için yalnızca bir kere çalışır. Daha sonra @Test ile işaretlenen methodlar çağrılır. Before her test methodu çağrılmadan önce bir kere çalışır. Test methodundan sonra After anasyonu çağrılır. Bu şekilde tüm methodlar devam eder. Methodlar bitip test bitmeden önce AfterClass çağrılır.

Kaynakça:

Resim : <https://semihsaydam.medium.com/junit5-ile-gelen-yenilikler-ae6ed95ce299>

<https://junit.org/junit5/docs/current/user-guide/#writing-tests-parameterized-tests>

[https://www.youtube.com/watch?v=ANHXTbnluyk&list=PLZYKO7600KN89x\\_tdqGhP7n8b0KXPC8EY&index=6](https://www.youtube.com/watch?v=ANHXTbnluyk&list=PLZYKO7600KN89x_tdqGhP7n8b0KXPC8EY&index=6)