

Mockito

TDD ve özel unit testlerde sistemde kullanılan service, repository gibi katmanları izole etmek için kullanılır. Örnek verecek olursa service içinde repository kullanıyorsak service içinde ki repository'i mocklarız service'in bağımlı olduğu repository'nin yerini mockladığımız repository tutar.

Gerçek sınıfı birebir taklit eden nesneler oluşturuyoruz da diyebiliriz. Mocklama işleminde istediğin gibi davranmasını sağlıyor.

Neden mock kullanmalıyız ?

Gerçek nesneler yavaş.

Gerçek nesneler tekrar edilebilir değil. Yani bir sonucu yoktur. Test ederken tek bir sonucu olmalı.

Gerçek nesneler zor yapılandırılır.

Yazdığımız birim testi tek bir birimi test eder bu yüzden kullandığı bağımlılıkların test edilmemesini sağlarız.

Bağımlılıkların testlere yan etkisinin olmaması için.

Mocklama işlemini mock kütüphanesi olan Mockito kullanacağız. Mockito objeleri bizim için kendi mocklar. Geniş kullanım yöntemi sağlar.

Mockito içinde kullanabileceğimiz bazı anotasyonları açıklamak istiyorum.

@Mock – Mock objesi oluşturmak için

@InjectMock – Mock objelerimizi test edeceğimiz sınıfımıza inject etmek için.

Bu anotasyonları kullanmak istiyorsa bununla birlikte kullanmalıyız @RunWith(MockitoJUnitRunner.class)

When, thenReturn static sonuçlar döndürmek için kullanılır.

Kaynaklar:

<https://www.gokalpkuscu.com/mocking-ve-mockito-mocking-framework-nedir/>

<https://gunceljava.blogspot.com/2018/11/mockito-when-thenreturn-kullanm.html>