

# Uygulamalı Laboratuvar: Fabrikalar ve Sahte Veriler Kullanımı

Gerekli tahmini süre: 30 dakika

**Fabrikalar ve Sahte Veriler Kullanımı** laboratuvara hoş geldiniz. Test etmek için genellikle sahte verilere ihtiyaç duyarsınız. Elbette, testlerinizde bazı sabit kodlanmış örnek verileri kullanabilirsiniz. Ama ya yüzlerce ya da binlerce test verisi kaydına ihtiyacınız varsa? Bu, oluşturmalarını ve bakımını yapması sıkıcı hale gelebilir.

Bu laboratuvara, testler için sahte veriler sağlamak üzere **FactoryBoy** adlı popüler bir Python paketini nasıl kullanacağınızı göreceksiniz.

## Öğrenme Hedefleri

Bu laboratuvari tamamladıktan sonra şunları yapabileceksiniz:

- Bir Factory sınıfı oluşturmanın özétini çıkarmak
- Gerçekçi test verileri sağlamak için Faker sınıfını ve Fuzzy özelliklerini kullanmak
- Test verileri sağlamak için Factory sınıflarını kullanan test senaryoları yazmak

## Theia Hakkında

Theia, masaüstünde veya bulutta çalıştırılabilen açık kaynaklı bir IDE'dir (Entegre Geliştirme Ortamı). Bu laboratuvari yapmak için Theia IDE'sini kullanacaksınız. Theia ortamına giriş yaptığınızda, yalnızca sizin için ayrılmış 'buluttaki özel bir bilgisayar' ile karşılaşırısz. Laboratuvarlar üzerinde çalıştığınız sürece bu sizin kullanımınıza açıktır. Çıkış yaptığınızda, bu 'buluttaki özel bilgisayar' ve oluşturmuş olabileceğiniz dosyalar silinir. Bu nedenle, laboratuvarlarınızı tek bir oturumda tamamlamak iyi bir fikirdir. Laboratuvarın bir kısmını tamamlayıp daha sonra Theia laboratuvarına döndüğünüzde, başlangıçtan başlamak zorunda kalabilirsiniz. Tüm Theia laboratuvarlarınızı tamamlamak için yeterli zamanınız olduğunda çalışmayı planlayın.

## Laboratuvar Ortamını Kurun

Laboratuvara başlamadan önce biraz hazırlık yapmanız gerekiyor.

### Bir Terminal Açıın

Editördeki menüyü kullanarak bir terminal penceresi açın: Terminal > Yeni Terminal.

Terminalde, eğer /home/projects klasöründe değilseniz, şimdi proje klasörünüze geçin.

```
cd /home/project
```

## Kodu Klonlayın

Şimdi test etmeniz gereken kodu alın. Bunu yapmak için, git deposunu klonlamak için `git clone` komutunu kullanın:

```
git clone https://github.com/ibm-developer-skills-network/duwjqx-tdd_bdd_PracticeCode.git
```

## Laboratuvar Klasörüne Geçin

Depoyu klonladıkten sonra, laboratuvar dizinine geçin:

```
cd duwjqx-tdd_bdd_PracticeCode/labs/05_factories_and_fakes
```

## Python Bağımlılıklarını Yükle

Son hazırlık adımı, laboratuvar için gerekli Python paketlerini yüklemek için `pip` kullanmaktadır:

```
python3.8 -m pip install -r requirements.txt
```

Artık laboratuvara başlamaya hazırlısanız.

## Opsiyonel

Eğer terminalde çalışmak zorlaşırsa çünkü komut istemi çok uzunsa, aşağıdaki komutu kullanarak istemi kısaltabilirsiniz:

```
export PS1="\[\033[01;32m\]\u\[033[00m\]: \[\033[01;34m\]\w\[033[00m\]]\$ "
```

## Koda Git

IDE'de, duwjx-tdd\_bdd\_PracticeCode/labs/05\_factories\_and\_fakes klasörüne gidin. Bu klasör, bu laboratuvar için kullanacağınız tüm kaynak kodlarını içerir.

```
duwjx-tdd_bdd_PracticeCode/labs/05_factories_and_fakes
```

## Adım 1: nosetests'i Çalıştır

Kodunuzda herhangi bir değişiklik yapmadan önce, tüm test durumlarının geçtiğinden emin olmalısınız. Aksi takdirde, daha sonra başarısız olan test durumlarıyla karşılaşırsanız, bunların başarısız olmasına sizin sebep olup olmadığını ya da değişiklik yapmadan önce zaten başarısız olduklarını bilemezsiniz.

nosetests komutunu çalıştırın ve tüm testlerin **%100** test kapsamıyla geçtiğinden emin olun.

```
nosetests
```

Aşağıdaki çıktıyı görmelisiniz:

Tüm testler yeşil renkte! Bu, hepsinin geçtiği anlamına geliyor, bu yüzden artık kodu değiştirmeye gelebilirsiniz.

## Adım 2: Bir AccountFactory Sınıfı Oluşturun

Bu adımda, bir AccountFactory sınıfı oluşturacaksınız.

Account sınıfının niteliklerini öğrenmek için `models/account.py` dosyasını açın. Bu, AccountFactory sınıfına eklemeniz gereken aynı niteliklerdir.

[Open account.py in IDE](#)

IDE editöründe `tests/factories.py` dosyasını açın. Bu, Account sınıfının niteliklerini AccountFactory sınıfına ekleyeceğiniz dosyadır.

[Open factories.py in IDE](#)

FactoryBoy'un Faker sınıfıyla birlikte geldiğinden yararlanmak istiyorsunuz. Bu sınıfın [Sahte sağlayıcıları](#) ve birçok [Belirsiz nitelikleri](#) bulunmaktadır.

Faker sınıfı için bazı yararlı sağlayıcılar şunlardır:

```
Faker("name")
Faker("email")
Faker("phone_number")
```

İşte faydalı bulabileceğiniz bazı bulanık (Fuzzy) özellikler:

```
FuzzyChoice(choices=[True, False])
FuzzyDate(date(2008, 1, 1))
```

## Göreviniz

Faker sağlayıcılarını ve Fuzzy özelliklerini kullanarak `id`, `name`, `email`, `phone_number`, `disabled` ve `date_joined` alanları için sahte veriler oluşturun ve bunları AccountFactory sınıfına ekleyin.

# Cözüm