

# Uygulamalı Laboratuvar: Behave ile Adım Oluřturma



**Gerekli tahmini süre:** 15 dakika

**Behave ile Adım Oluřturma** laboratuvarına hoř geldiniz. İlk görev, feature dosyasındaki ifadeleri karřılamak için adımları oluřturmaktır. Bu görevi desteklemek için, **Behave** aracı, çalışanınıza bařlamak için iskelet kodu olarak hizmet eden fonksiyon řablonları oluřturur.

Bu laboratuvar, behave komutunu kullanarak adım dosyalarınız için bařlangıç kodu oluřturmayı öğreneceksiniz.

## Öğrenme Hedefleri

Bu laboratuvarı tamamladıktan sonra řunları yapabileceksiniz:

- Adımlar için öneriler oluřturmak üzere behave aracını kullanmak
- Bu önerileri alıp onlardan bir bařlangıç adım dosyası oluřturmak.

## Theia Hakkında

Theia, masaüstünde veya bulutta çalıştırılabilen açık kaynaklı bir IDE (Entegre Geliřtirme Ortamı)dir. Bu laboratuvarı yapmak için Theia IDE'sini kullanacaksınız. Theia ortamına giriř yaptığınızda, yalnızca bulutta özel olarak sunulan bir bilgisayar ile karřılařacaksınız. Bu, laboratuvarlar üzerinde çalıştığınız sürece mevcut olacaktır. Çıkıř yaptığınızda, bu bulut üzerindeki özel bilgisayar ve oluřturmuř olabileceğiniz dosyalar silinecektir. Bu nedenle, laboratuvarlarınızı tek bir oturumda tamamlamak iyi bir fikirdir. Laboratuvarın bir kısmını bitirip daha sonra Theia laboratuvarına döndüğünüzde, bařtan bařlamamız gerekebilir. Tüm Theia laboratuvarlarınızı tamamlamak için zamanınız olduğunda çalışmayı planlayın, böylece bir laboratuvarı tek bir oturumda tamamlayabilirsiniz.

## Laboratuvar Ortamını Kurma

Laboratuvara bařlamadan önce ortamı hazırlamanız gerekiyor. Öncelikle, bir terminal açın ve bazı sistem ve Python bağımlılıklarını yükleyin.

### Terminali Açın

Editörün menüsünü kullanarak bir terminal penceresi açın: Terminal > Yeni Terminal.

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

```
cd /home/project
```

## Kodu Repo'sunu Klonlayın

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

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

## Repo Klasörüne Geçin

Ardından, laboratuvar dosyalarını içeren Repo dizinine geçmeniz gerekiyor.

```
cd /home/project/duwjx-tdd_bdd_PracticeCode
```

## Laboratuvar Bağımlılıklarını Kur

Depoyu klonladıktan sonra, geliştirme ortamına bazı ön koşul yazılımlarını kurmanız gerekiyor.

```
bash ./bin/setup.sh
```

## Laboratuvar Klasörüne Geçin

Sonraki adımda, laboratuvar dosyalarını içeren laboratuvar dizinine geçmeniz gerekiyor.

```
cd /home/project/duwjsx-tdd_bdd_PracticeCode/labs/11_generating_steps
```

## Python Bağımlılıklarını Yükle

Son hazırlık adımı, laboratuvar için gerekli Python paketlerini yüklemek üzere pip kullanmaktır:

```
pip install -r requirements.txt
```

Artık laboratuvara başlamak için hazırsınız.

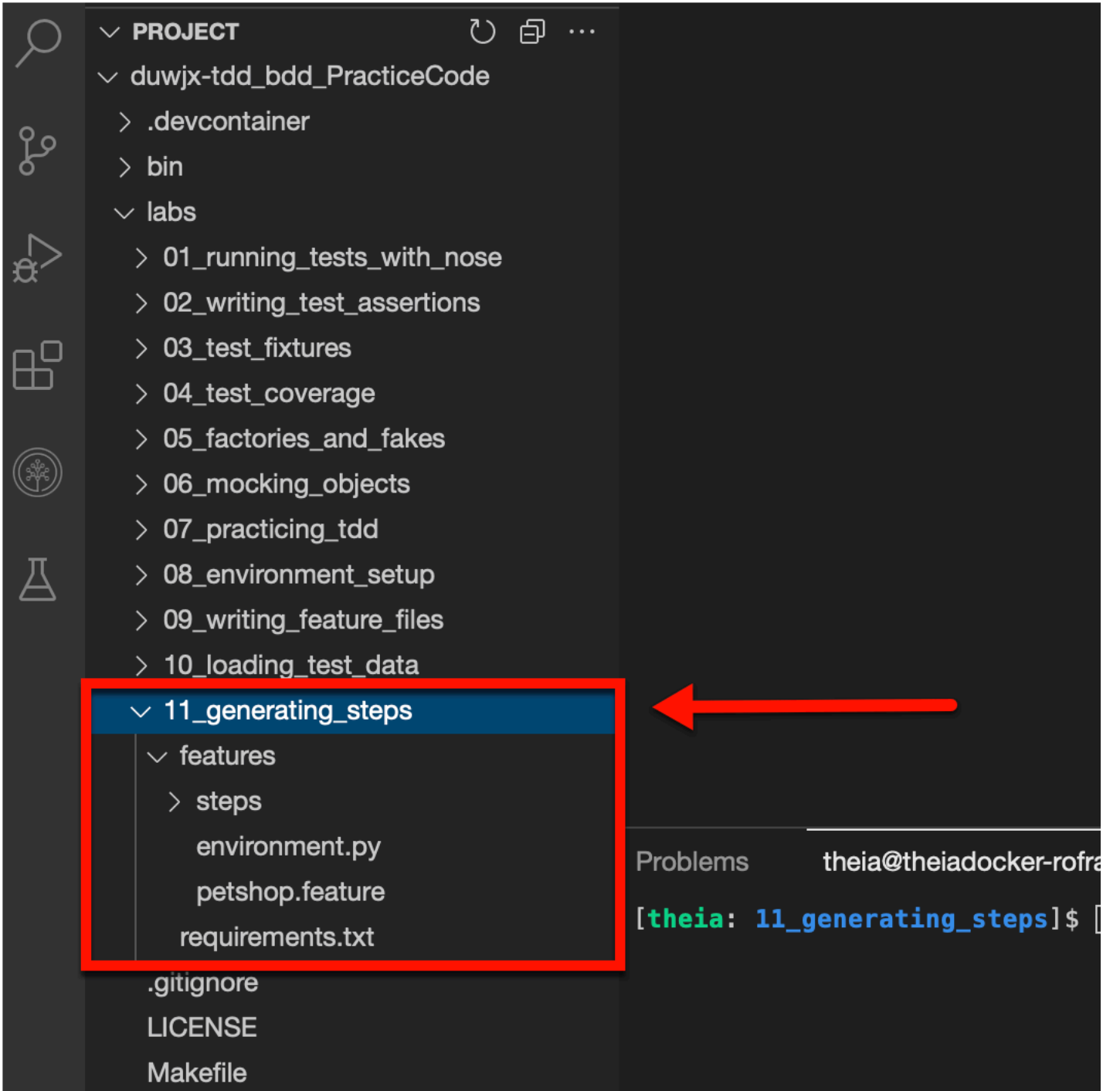
### Opsiyonel

Eğer terminalde çalışmak, komut isteminin uzunluğu nedeniyle zorlaşırsa, aşağıdaki komutu kullanarak istemi kısaltabilirsiniz:

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

## Koda Git

IDE'de duwjsx-tdd\_bdd\_PracticeCode/labs/11\_generating\_steps klasörüne gidin. Bu klasör, bu laboratuvar için gereken tüm kaynak kodunu içerir.



## Adım 1: Adım Önerilerini Oluştur

İlk kez behave komutunu çalıştırdığınızda, ./features klasöründe .feature uzantılı dosyaları arayacak ve bunları eşleştirmek için ./features/steps klasöründeki Python dosyalarında adımları bulmaya çalışacaktır. Herhangi bir adım eksikse, oluşturulması gereken adım tanımlarını önerir.

Özellik dosyasını ilk oluşturduğunuzda, çoğu adım eksik olacaktır. İlk adımlarınız için işlev tanım stub'larını oluşturabilirsiniz.

### Göreviniz

behave komutunu çalıştırın. Ardından ./features/steps/web\_steps.py dosyasını düzenleyin ve önerileri dosyaya gerekli bazı importlarla birlikte yapıştırın ve kaydedin.

IDE editöründe features/steps/web\_steps.py dosyasını açın. Laboratuvarın geri kalanında bu dosya üzerinde çalışacaksınız.

Open **web\_steps.py** in IDE

1. Bir Terminal kabuğu açın ve eksik olanları görmek için behave komutunu çalıştırın:

```
behave
```

Veri tablosundan sonraki ilk satırın sarı olduğunu fark edin çünkü eşleşen bir adım bulamadı.

```
Feature: Search for pets by category # features/petshop.
  As a pet shop customer
  I need to be able to search for a pet by category
  So that I only see the category of the pet I am interested in
  Background: # features/petshop.feature:7
```

```
Scenario: Search for dogs # features/petshop.feature:14
```

```
  Given the following pets # features/petshop.feature:15
```

	name	category	available	gender	birthday
	Fido	dog	True	MALE	2019-11-01
	Kitty	cat	True	FEMALE	2020-08-15
	Leo	lion	False	MALE	2021-04-01

```
    Given I am on the "Home Page" # None
```

```
    When I set the "Category" to "dog" # None
```

```
    And I click the "Search" button # None
```

```
    Then I should see the message "Success" # None
```

```
    And I should see "Fido" in the results # None
```

```
    But I should not see "Kitty" in the results # None
```

```
    And I should not see "Leo" in the results # None
```

```
Failing scenarios:
```

```
  features/petshop.feature:14 Search for dogs
```

```
0 features passed, 1 failed, 0 skipped
```

```
0 scenarios passed, 1 failed, 0 skipped
```

```
1 step passed, 0 failed, 0 skipped, 7 undefined
```

```
Took 0m0.055s
```

2. Çıktının en altındaki sarı renkteki kod tanımlarını kopyalayın ve web\_steps.py dosyasına yapıştırın:

Not: Adımları kopyalamak için Terminal içinde sağ tıklayıp **Kopyala** komutunu seçmeniz gerekebilir.

Failing scenarios:

features/petshop.feature:14 Search for dogs

0 features passed, 1 failed, 0 skipped

0 scenarios passed, 1 failed, 0 skipped

1 step passed, 0 failed, 0 skipped, 7 undefined

Took 0m0.066s

You can implement step definitions for undefined steps

```
@given(u'I am on the "Home Page"')
```

```
def step_impl(context):
```

```
    raise NotImplementedError(u'STEP: Given I am on the
```

```
@when(u'I set the "Category" to "dog"')
```

```
def step_impl(context):
```

```
    raise NotImplementedError(u'STEP: When I set the "C
```

```
@when(u'I click the "Search" button')
```

```
def step_impl(context):
```

```
    raise NotImplementedError(u'STEP: When I click the
```

```
@then(u'I should see the message "Success"')
```

```
def step_impl(context):
```

```
    raise NotImplementedError(u'STEP: Then I should see
```

```
@then(u'I should see "Fido" in the results')
```

```
def step_impl(context):
```

```
    raise NotImplementedError(u'STEP: Then I should see
```

```
@then(u'I should not see "Kitty" in the results')
```

```
def step_impl(context):
```

```
    raise NotImplementedError(u'STEP: Then I should not
```

```
@then(u'I should not see "Leo" in the results')
```

```
def step_impl(context):
```

```
def step_impl(context):  
    raise NotImplementedError(u'STEP: Then I should not
```

3. Bağımlılıkları çözmek için gerekli olan aşağıdaki importları web\_steps.py dosyasının en üstüne ekleyin.

```
from behave import given, when, then  
from selenium.webdriver.common.by import By
```

4. Dosyayı kaydedin.

## Çözüm

Aşağıda dosyayı doğru oluşturduğunuzu kontrol edebilirsiniz.

▼ Çözüm için buraya tıklayın.

Bu, başlangıç adımlarını oluşturma çözümüdür:

```
from behave import given, when, then  
from selenium.webdriver.common.by import By  
@given(u'I am on the "Home Page"')  
def step_impl(context):  
    raise NotImplementedError(u'STEP: Given I am on the "Home Page"')  
@when(u'I set the "Category" to "dog"')  
def step_impl(context):  
    raise NotImplementedError(u'STEP: When I set the "Category" to "dog"')  
@when(u'I click the "Search" button')  
def step_impl(context):  
    raise NotImplementedError(u'STEP: When I click the "Search" button')  
@then(u'I should see the message "Success"')  
def step_impl(context):  
    raise NotImplementedError(u'STEP: Then I should see the message "Success"')  
@then(u'I should see "Fido" in the results')  
def step_impl(context):  
    raise NotImplementedError(u'STEP: Then I should see "Fido" in the results')  
@then(u'I should not see "Kitty" in the results')  
def step_impl(context):  
    raise NotImplementedError(u'STEP: Then I should not see "Kitty" in the results')  
@then(u'I should not see "Leo" in the results')  
def step_impl(context):  
    raise NotImplementedError(u'STEP: Then I should not see "Leo" in the results')
```

▼ Details

```
::page{title="Adım 2: Behave'i Çalıştır"}
```

Artık oluşturduğunuz web adımlarını bulup bulmadığını görmek için behave komutunu çalıştırmaya hazırsınız.

## Göreviniz

1. Bir terminal kabuğunda behave komutunu çalıştırın

```
behave
```

Çıktı aşağıdaki resme benzer görünmelidir. İlk ifadenin şimdi kırmızı olduğunu fark edin; bu, adımı bulduğunu ancak adımın `NotImplementedError` hatası verdiğini gösteriyor, bu da sürpriz değil. Adımlar henüz uygulanmadığı için, adımı gerçekleştirmek için gerçek kodu ekleyene kadar hepsi bu hatayı verecektir.

```
Feature: Search for pets by category # features/petshop.feature
  As a pet shop customer
  I need to be able to search for a pet by category
  So that I only see the category of the pet I am interested
  Background: # features/petshop.feature:7
```

```
Scenario: Search for dogs # features/p
  Given the following pets # features/s
    | name | category | available | gender | birthday |
    | Fido | dog      | True      | MALE   | 2019-11-18 |
    | Kitty | cat      | True      | FEMALE | 2020-08-13 |
    | Leo   | lion     | False     | MALE   | 2021-04-01 |
  Given I am on the "Home Page" # features/s
```

```
Traceback (most recent call last):
  File "/home/theia/venv/lib/python3.9/site-packages/b
    match.run(runner.context)
  File "/home/theia/venv/lib/python3.9/site-packages/b
    self.func(context, *args, **kwargs)
  File "features/steps/web_steps.py", line 14, in step
    raise NotImplementedError(u'STEP: Given I am on th
NotImplementedError: STEP: Given I am on the "Home Pag
```

```
When I set the "Category" to "dog" # None
And I click the "Search" button # None
Then I should see the message "Success" # None
And I should see "Fido" in the results # None
But I should not see "Kitty" in the results # None
And I should not see "Leo" in the results # None
```

Failing scenarios:

```
features/petshop.feature:14 Search for dogs
```

```
0 features passed, 1 failed, 0 skipped
0 scenarios passed, 1 failed, 0 skipped
1 step passed, 1 failed, 6 skipped, 0 undefined
Took 0m0.064s
```

Artık bir sonraki laboratuvara geçmeye hazırsınız.

```
::page{title="Sonuç"}
```

Tebrikler! **Behave ile Adım Oluşturma** laboratuvarını tamamladınız. Artık bir BDD feature dosyası için testleri uygulamak üzere başlangıç adımlarını oluşturma becerisine sahip olmalısınız. Gerçek kod ile doldurabileceğiniz işlev stub'ları oluşturmak için behave aracını kullanabilirsiniz.

Bir sonraki laboratuvarınızda adımları yazmayı öğreneceksiniz.



**Author(s)**

[John J. Rofrano](#)

**Contributors**

© IBM Corporation. Tüm hakları saklıdır.