

Uygulamalı Laboratuvar - JPetStore'u Mikroservislerle Modernize Et

Tahmini Süre: 30 dakika

Bu laboratuvara, Swagger UI kullanımıyla tanışacaksınız. Swagger UI, OpenAPI (Swagger) spesifikasyonu ile tanımlanan bir API'nin belgelerini görsel olarak sunmak için kullanılan açık kaynaklı bir projedir. REST API'leri, bir iletişim kanalının bir ucunu temsil eder. Bir API başka bir sistemle etkileşime girdiğinde, bu iletişimın temas noktaları uç noktalar olarak kabul edilir. API'ler için bir uç nokta, bir sunucu veya hizmetin URL'sini içerebilir.

Öğrenme Hedefleri:

Bu alıştırmayı tamamladıktan sonra aşağıdaki görevleri yerine getirebilmelisiniz:

- Swagger UI'yi kullanım ve çeşitli bileşenleri anlayın.
- Swagger UI aracılığıyla uç noktalara erişin.

Ön Koşullar

- Docker uygulamaları ve komutları hakkında bilgi sahibi olmalısınız.
- REST API hakkında iyi bir anlayışa sahip olmalısınız.

Görev 1 - Swagger UI'yi Alma

- IDE'deki üst menüyü kullanarak bir terminal penceresi açın: **Terminal > Yeni Terminal**.
- Terminalde, Swagger için docker imajını çekmek üzere aşağıdaki komutu çalıştırın.

```
docker pull swaggerapi/swagger-ui
```

- Swagger uygulamasını çalıştırmak ve arayüze erişmek için aşağıdaki komutu çalıştırın.

```
docker run -dp 8080:8080 swaggerapi/swagger-ui
```

Not - Size bir hex kodu donecek. Bu, sunucunun başarıyla başlatıldığı anlamına gelir.

4. Aşağıdaki butona tıklayın veya **Skills Network** simgesine tıklayın ve menüden **Uygulamayı Başlat** seçeneğini seçin ve port numarasını *8080* girin.

Uygulamayı Başlat

5. Tarayıcı açılır ve Swagger uygulamasının çalıştığı port 8080'e bağlanır.

Swagger UI, varsayılan, önceden yapılandırılmış örnek [PetStore JSON](#) ile bağlanır. Bu JSON dosyasını yeni bir sayfada açarak kontrol edebilirsiniz.

Sayfa aşağıda gösterildiği gibi görünecek. Aşağıdakileri listeler:

1. Kullanılan JSON
2. Protokol şeması
3. REST isteklerinin türü ile birlikte uç noktalar - GET, POST, PUT, UPDATE, DELETE
4. Ayrıca uygulamada kullanılan Modelleri listeler

Görev 2 - Swagger UI üzerinden uç noktalara erişim

Bu görevin bir parçası olarak -

1. Bir evcil hayvan ekle
2. Evcil hayvanın detaylarını id ile al
3. Evcil hayvanın durumunu **satıldı** olarak güncelle
4. Durumun güncellenip güncellenmediğini görmek için evcil hayvanın detaylarını id ile al
5. Evcil hayvanı sil
6. Evcil hayvanın mevcut olmadığını görmek için evcil hayvanın detaylarını id ile al

POST, GET ve DELETE uç noktalarını deneyeceksiniz.

1. **POST /pet** uç noktasının yanındaki açılır menüye tıklayın.
2. Bir evcil hayvan eklemek için **Try it out** seçeneğine tıklayın.
3. Bu, değerleri düzenlemenizi sağlayacaktır. Önceden doldurulmuş JSON'u aşağıdaki ile değiştirin:

```
{  
    "id": 10,  
    "category": {  
        "id": 1,  
        "name": "dogs"  
    "name": "Hershey",  
    "photoUrls": [  
        "https://i.natgeofe.com/n/4f5aaece-3300-41a4-b2a8-ed2708a0a27c/domestic-dog_thumb_3x2.jpg"  
    "tags": [  
        {  
            "id": 1,  
            "name": "Friendly"  
    "status": "available"
```

}

4. **Execute** butonuna tıklayarak evcil hayvanı ekleyin. Başarılı bir POST isteği olduğunu gösteren 200 Kodlu bir **Sunucu Yanıtı** görelmisiniz.
5. **POST /pet** açılır menüsünü kapatın.
6. Şimdi eklediğiniz evcil hayvanın detaylarını alacaksınız. **GET /pet/{petid}** yanındaki açılır menüye tıklayın.
7. **Try it out** butonuna tıklayın ve detaylarını almak istediğiniz evcil hayvanın id'sini girin. Bizim evcil hayvanımızın id'si **10**. **Execute** butonuna tıklayın.
8. Elde edilen Sunucu Yanıtında, isteğin başarılı olduğunu ve evcil hayvanın detaylarını gösteren **200** göreceksiniz.
9. **GET /pet/{petid}** açılır menüsünü kapatın.
10. Şimdi evcil hayvanın durumunu **satıldı** olarak güncelleyeceksiniz. Bunu yapmak için **POST /pet/{petid}** yanındaki açılır menüye tıklayın.
11. Şimdi **10** id'li evcil hayvanın durumunu **satıldı** olarak değiştireceksiniz. **Deneyin** butonuna tıklayın ve aşağıda gösterildiği gibi değişiklikleri yapın, ardından **Çalıştır** butonuna tıklayın.
12. Güncellemeye başarılı bir şekilde çalıştıysa, **200** kodu ile bir sunucu yanıtı alırsınız.
13. **POST /pet/{petid}** açılır menüsünü kapatın.
14. Güncellediğiniz evcil hayvanın **satıldı** olarak yansiyip yansımadığını kontrol edebilirsiniz. **GET /pet/{petid}** yanındaki açılır menüye tıklayın.
15. Detaylarını almak istediğiniz evcil hayvanın id'sini girin. Güncellediğiniz evcil hayvanın id'si **10**. **Çalıştır** butonuna tıklayın.
16. Aldığınız Sunucu Yanıtında, isteğin başarılı olduğunu ve evcil hayvanın detaylarını gösteren **200** kodunu göreceksiniz.
17. **GET /pet/{petid}** açılır menüsünü kapatın.
18. Artık evcil hayvan satıldığından, onu sisteminizden silebilirsiniz. **DELETE /pet/{petid}** yanındaki açılır menüye tıklayın. **Try it out** butonuna tıklayın ve **petid** olarak 10 girin.

19. **Execute** butonuna tıklayın. Silme işlemi başarılı bir şekilde gerçekleştiyse, **200** kodu ile bir sunucu yanıtı alırsınız.
20. Yeni sildiğiniz evcil hayvanın sistemden kaldırılıp kaldırılmadığını kontrol edebilirsiniz. **GET /pet/{petid}** yanındaki açılır menüye tıklayın.
21. Detaylarını almak istediğiniz evcil hayvanın id'sini girin. Sildiğiniz evcil hayvanın id'si **10. Execute** butonuna tıklayın.
22. Aldığınız Sunucu Yanıtı'nda **404** göreceksiniz, bu da isteğin hatalı olduğunu ve böyle bir evcil hayvanın bulunmadığını gösterir.

Tebrikler! Görevi başarıyla tamamladınız.

Eğitim döaylorı

Yazar: Lavanaya T S

Katkıda Bulunanlar: Pallavi Rai

© IBM Corporation. Tüm hakları saklıdır.