

SWAGGER

Swagger Nedir?

Swagger, Restful Api'leri kullanım aracıdır diyebiliriz. Bu kullanım nasıl olur diye düşünersek, backend servisinde yazılan url'lerin front end veya mobil geliştiricilerin, Swagger kullanarak tasarlanması , belgelenmesi veya test edilmesi şeklinde bir kullanımı söz konusudur. Bu tool, api url'lerin verdiği cevaplar veya aldığı parametreleri görselleştirerek geliştiricilere kolaylık sağlamaktadır.

Swagger Neden Kullanılır?

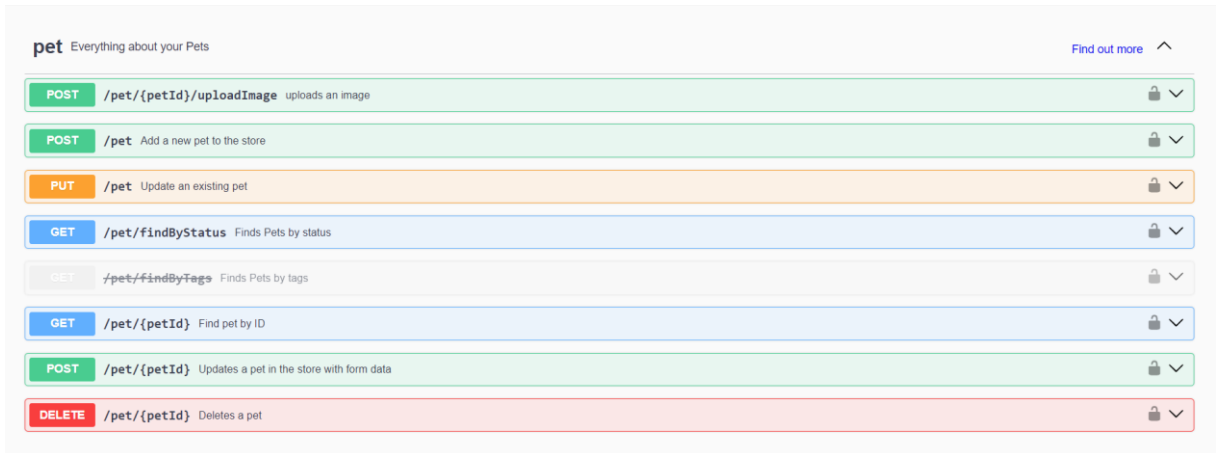
Servis işlemlerinde uygulamanın arka planı için sunucudan alacağımız veriler veya yükleyeceğimiz veriler için işlem yapan servisçi, biz mobil uygulama geliştiricilerine url gönderir ve bu url ile http istekleri atarız. Peki bu http isteklerinin post mu get mi delete mi vs... hangisiyle atacağımızı hangi parametreleri göndereceğimizi nasıl bileceğiz. İşte burda Swagger kullanmalıyız.

Swagger Nasıl Kullanılır?

Diyelim ki elimizde <https://petstore.swagger.io/> şöyle bir base url var bu base url'in pathlerini swaggerde belirtiyoruz ve get'mi post'mu diye bunu kullanacak geliştiriciye yardımcı oluyoruz. Bu linki açtığınızda karşınıza bi ekran gelecek o ekranı en baştan en aşağıya inceleyelim.



Ekranın en üstünde url'nin verdiği json yapısını görmemiz için bağlantı bulunur bu bağlantıya tıkladığımızda json yapısını basit düzeyde görürüz.



Burda gördüğümüz kısım ise en önemli kısımlardan birisidir. Base url'nin gidebileceği path'lere göre modüller bulunmaktadır. Bu modüllerden pet modülü üzerinde konuşmaya başlayalım. Bu modülde yapabileceğimiz 8 ayrı http isteği vardır. Bu http isteklerinin türü solda belirtilmiştir. Geçersiz olan url'nin üzeri

çizilmiş olup gri renkle boyanmıştır. Şimdi bu isteklerden Post isteğinin üzerine tıklayalım ve daha detayına inelim.

Name	Description
petId * required integer(\$int64) (path)	ID of pet to update <input type="text" value="petId"/>
additionalMetadata string (formData)	Additional data to pass to server <input type="text" value="additionalMetadata"/>
file file (formData)	file to upload <input type="button" value="Dosya Seç"/> Dosya seçilmedi

Responses

Response content type: application/json

Code	Description
200	successful operation

Example Value | Model

```
{  "code": 0,  "type": "string",  "message": "string"}
```

İlk sıradaki Post isteğini açınca şöyle bir ekranla karşılaşırız Bu ekranda url'nin ihtiyaç duyduğu parametreleri Swagger bize belirtir. Bu parametrelerden zorunlu olan required ile belirtilmiştir ve integer bir değer istediğini bize gösteriyor. Artı olarak url'nin amacı belirli bir id'ye sahip hayvanın resmini yüklemektir ve file ve metadata bilgisi istiyor.Swagger'in gelişmiş görselleştirme ve işlevlerini burda hissediyoruz.

GET /pet/findByStatus Finds Pets by status

Multiple status values can be provided with comma separated strings

Parameters

Try it out

Name	Description
status * required array[string] (query)	Status values that need to be considered for filter Available values : available, pending, sold <input type="text" value="available"/> pending sold

Burda ise get isteğini görüyoruz sağdaki try it out butonuna basınca parametreler ekranı işlevli oluyor ve bu parametre ekranında servişçinin belirttiği 3 durumdan birini seçip duruma göre bulunan hayvanların bilgisi aşağıdaki gibi dönüyor.

```
Code Details
200
Response body
[
  {
    "id": 9223372016900205000,
    "category": {
      "id": 0,
      "name": "string"
    },
    "name": "doggie",
    "photoUrls": [
      "string"
    ],
    "tags": [
      {
        "id": 0,
        "name": "string"
      }
    ],
    "status": "available"
  },
  {
    "id": 9223372016900205000,
    "category": {
      "id": 0,
      "name": "string"
    },
    "name": "Puff",
    "photoUrls": [
      "string"
    ],
    "tags": [
      {
        "id": 0,
        "name": "string"
      }
    ],
    "status": "available"
  }
]
Response headers
access-control-allow-headers: Content-Type,api_key,Authorization
access-control-allow-methods: GET,POST,DELETE,PUT
access-control-allow-origin: *
content-type: application/json
date: Tue, 14 May 2024 13:05:40 GMT
server: Jetty(9.2.9.v20150224)
```

Bu şekilde base url'nin gidebileceği url ve pathleri test edip deneyebiliriz ve herhangi bir hata durumunda hatayı servişçiye ya da api'mizi biz yaptıysak kendimiz düzeltebiliriz. Bu şekilde birçok müdiül vardır belirttiğim linkten erişebilirsiniz.

JWT VE BASİC AUTH YAPISI TESTİ

Swagger'de jwt tabanlı auth işlemlerinde authorization kısmında bir jwt token girilir ve bu taşıyıcıyla login ve logout işlemleri yapılır. Swagger belgesini düzenlerken securityDefinitions kısmında jwt için bir tanım yapması gerekecektir.

Basic auth işlemleri için yine Swagger üzerinde authorization kısmında;

Available authorizations



Basic authorization ←

Username:

Password:

Authorize

Close

Böyle bir ekran açılacaktır ve burdan username ve password bilgileri girilerek login testi yapılabilir. Bu şekilde Swagger bize birçok şeyi kolaylıkla test etmemizi sağlar.