

SWAGGER

Swagger, API'lerin tasarlanması, test edilmesi ve dokümanının hazırlanması konusunda tercih edilen bir araçtır. API'leri daha anlaşılır bir şekilde sunar, geliştiricilerin API'leri kullanma sürecini hızlandırır, otomatik belgelendirme ve test edilebilmesini sağlar.

API'lerin olumlu ve olumsuz durumlardaki istek ve yanıtlarını kolay bir şekilde görülebilmesi ve bu dokümanın otomatik bir şekilde oluşturulması API'lerin kullanıma geçme hızını artırır ve belge oluşturma yükünü ortadan kaldırır.

Swagger Bileşenleri ve Kullanım Alanları:

Swagger Editor: API'nizin OpenAPI Specification (OAS) tanımını oluşturmak için Swagger Editor'ü kullanabilirsiniz. API'nizin tüm endpoint'lerini, endpoint'lerin ne yaptığını ve hangi parametreleri aldığını belirtmenize olanak sağlar.

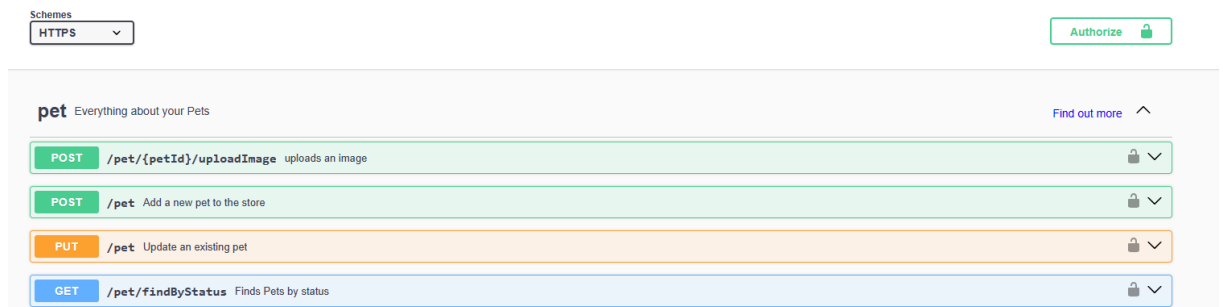
Swagger Codegen: OAS tanımınızı oluşturduktan sonra, Swagger Codegen'i kullanarak sunucu uygulamasını oluşturabilirsiniz. API'nizin belirli bir dilde otomatik olarak oluşturulmasını sağlar.

Swagger UI: API'nizin belgelerini görselleştirilmiş bir şekilde oluşturmak ve bir ekip olarak geliştirmek için Swagger UI'yi kullanabilirsiniz.

API Testi: Swagger UI, API testi için de kullanılabilir. API'nizin belgelerini oluşturduktan sonra, Swagger UI'yi kullanarak API'nizin her bir endpoint'ini test edebilir ve sonuçları görebilirsiniz.

Swagger ile JWT ve Basic Auth Testleri

- Swagger UI'ı Başlatın:** İlk olarak, Swagger UI'yi başlatın. Bu genellikle tarayıcınızda bir URL'yi ziyaret ederek yapılır.
- Yetkilendirme Anahtarlarını Ekleyin:** Swagger UI, herhangi bir yetkilendirme anahtarını eklemek için bir alan sağlar. JWT veya Basic Auth yapısını test etmek için bu alanı kullanın.
- API İsteklerini Yapın:** Auth yapılarını test etmek için doğru endpointlere sahip olan bir API seçin. Ardından, gerekli yetkilendirme anahtarlarını ekleyin ve istekleri gönderin.
- İstek Yanıtlarını İnceleyin:** Gönderdiğiniz isteklerin yanıtlarını alın ve doğruluğunu kontrol edin. Yetkilendirme başarılıysa, isteğinizin beklediğiniz gibi yanıt verdiğinden emin olun.



api_key (apiKey)

Name: api_key

In: header

Value:

Authorize

Close

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes.

API requires the following scopes. Select which ones you want to grant to Swagger UI.

petstore_auth (OAuth2, implicit)

Authorization URL: <https://petstore.swagger.io/oauth/authorize>

Flow: implicit

client_id:

Scopes: [select all](#) [select none](#)

☒ *read:pets*
read your pets

☐ *write:pets*
modify pets in your account

Authorize

Close

SWAGGER TESTLERİ

Swagger UI ile oluşturulmuş dokümantasyonlar aşağıdaki gibi görselleştirilir. Bu, API'nin dokümantasyonudur. Burada API isteği atabilir, dönen HTTP durum kodunu görebilir ve dönen sonuçları gözlemleyebilirsiniz.

user Operations about user		^
POST	/user Create user	▼
POST	/user/createWithList Creates list of users with given input array	▼
GET	/user/login Logs user into the system	▼
GET	/user/logout Logs out current logged in user session	▼
GET	/user/{username} Get user by user name	▼
PUT	/user/{username} Update user	▼
DELETE	/user/{username} Delete user	▼

Burada istenilen türde API isteği atılabilir. Atılan API request parametreleri, request body'si gözlemlenebilir.

Parameters

Cancel

No parameters

Request body

application/json

Created user object

```
{  "id": 10,  "username": "theUser",  "firstName": "John",  "lastName": "James",  "email": "john@email.com",  "password": "12345",  "phone": "12345",  "userStatus": 1}
```

Execute

Clear

Dönen sonucun HTTP durum kodu, response body'si, response header'ı, örnek Json nesnesi gözlemlenebilir. Bu işlemler tüm istek türlerinde yapılabilir.

Code


Details

500

Error: response status is 500

Response body

```
{
  "code": 500,
  "message": "There was an error processing your request. It has been logged (ID: 921e04dae292166d)"
}
```

 [Download](#)

Response headers

```
content-length: 110
content-type: application/json
```

Responses

Code	Description	Links
------	-------------	-------

default	successful operation	No links
---------	----------------------	----------

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "id": 10,
  "username": "theUser",
  "firstName": "John",
  "lastName": "James",
  "email": "john@email.com",
  "password": "12345",
  "phone": "12345",
  "userStatus": 1
}
```