

## Swagger

Swagger, RESTful API'lerin geliştirilmesi ve tanımlanması için bir dizi kural, spesifikasyon ve araçlar içeren açık kaynaklı bir sistemdir. Swagger çerçevesi, geliştiricilere, etkileşimli, makine ve insan tarafından okunabilir API belgeleri oluşturmak için olanak sağlar.

API spesifikasyonları genellikle desteklenen işlemler, parametreler ve çıktılar, yetkilendirme gereksinimleri, gerekli lisanslar gibi bilgileri içerir. Swagger, API'nin işaretleme dilinden belge dosyası döndürerek bu bilgileri otomatik olarak kaynaktan oluşturabilir.

Swagger, kullanıcıların RESTful web hizmetleri oluşturmaya, belgelemesine, test etmesine ve tüketmesine yardımcı olur. Hem üstten aşağıya hem de alttan yukarıya API geliştirme yaklaşımı ile kullanılabilir. Üstten aşağıya veya tasarım-önce yöntemde, Swagger kod yazılmadan önce bir API'yi tasarlamak için kullanılabilir. Altan yukarıya veya kod-önce yöntemde ise, Swagger bir API için yazılan kodu alır ve belgelemeyi oluşturur.

## Swagger Bileşenleri

- **Swagger Editor** → Geliştiricilere yeni API'ler için belge yazma, tasarlama ve tanımlama yeteneği ve aynı zamanda mevcut olanları düzenleme yeteneği sağlar. Tarayıcı tabanlı editör, OpenAPI spesifikasyonlarını görsel olarak işler, hataları yönetir ve gerçek zamanlı geri bildirim sağlar.
- **Swagger Codegen** → Geliştiricilere farklı platformlar için istemci kütüphanesi kodu ve SDK'larını oluşturma yeteneği verir.
- **Swagger User Interface** → Çeşitli platformlar için belge oluşturmaya yardımcı olan tamamen özelleştirilebilir bir araçtır. Herhangi bir ortamda barındırılabilir.
- **Swagger Inspector** → API belgeleri için bir test aracıdır. API'ler kolayca doğrulanabilir ve sonuçlar otomatik olarak bulutta kaydedilir ve erişilir.

## Swagger'in Yararları

- API'lerin planını haritalandıran kullanıcı dostu bir arayüze sahiptir.
- Dokümantasyon hem geliştiriciler hem de müşteriler veya proje yöneticileri gibi geliştirici olmayanlar için anlaşılabilir.
- Spesifikasyonlar insan ve makine tarafından okunabilir.
- Etkileşimli, kolayca test edilebilir belgeler oluşturur.
- 40'tan fazla dilde API kitaplıklarının oluşturulmasını destekler.
- Düzenlemelerin daha kolay yapılabilmesi için JSON ve YAML formatı kabul edilebilir.
- API ile ilgili süreçlerin otomatikleştirilmesine yardımcı olur.

## 1. JWT Yetkilendirme Yapısının Swagger ile Test Edilmesi:

- JWT ile korunan bir API'yi test etmek için Swagger'da "Authorize" veya "Security" bölümünü kullanabilirsiniz.
- Swagger UI'da, sayfanın üst kısmında genellikle "Authorize" adında bir düğme bulunur. Bu düğmeye tıkladığınızda bir modül açılır.
- Bu modülde, genellikle bir "Token" alanı vardır. JWT'ninizi buraya ekleyebilirsiniz. Ardından, bu JWT ile isteklerinizin yapılacağı API'yi test edebilirsiniz.
- Bazı durumlarda, JWT'yi elle girmek yerine "Bearer" token tipi seçeneği altında JWT'yi eklemeniz gerekebilir.

## 2. Basic Auth Yapısının Swagger ile Test Edilmesi:

- Basic Auth kullanarak korunan bir API'yi test etmek için yine Swagger'da "Authorize" veya "Security" bölümünü kullanabilirsiniz.
- "Authorize" düğmesine tıkladığınızda, genellikle bir dropdown menü açılır. Burada "Basic Auth" seçeneğini bulabilirsiniz.
- Basic Auth seçeneğini seçtikten sonra, kullanıcı adı ve şifrenizi girmeniz için alanlar açılacaktır. Kullanıcı adı ve şifrenizi girip "Authorize" veya "Submit" düğmesine tıklayarak yetkilendirmeyi gerçekleştirebilirsiniz.
- Daha sonra API isteklerinizi yaparken bu yetkilendirme bilgileri otomatik olarak her istekte gönderilecektir.

## Swagger nasıl test edilir?

1. API'nin Swagger belgelerini bulun.
2. API'nin Swagger UI'sini açın. Bu size API'nin kullanımını ve test edilmesini sağlar.
3. API'nin sunduğu end point'leri ve bu end point'lerin operasyonlarını gözden geçirin (GET, POST, vb.).
4. Her operasyonu test edin, parametreleri doldurun ve çağrıları yapın.
5. Her çağrının cevabını kontrol edin ve API'nin beklendiği gibi çalışıp çalışmadığını görün.
6. API'nin nasıl hata işlediğini test edin.

## Swagger Örnek: Post İşlemi

POST /user createUser

Parameters

Try it out

Name	Description
<b>user</b> * required (body)	USER

Example Value | Model

```
{  "id": 0,  "name": "string"}
```

Parameter content type  
application/json

Responses  
Response content type \*/\*

Post işleminin 1.adımı[2]

POST /user createUser

Parameters

Cancel

Name	Description
<b>user</b> * required (body)	USER

Example Value | Model

```
{  "id": 0,  "name": "DenemeString!"}
```

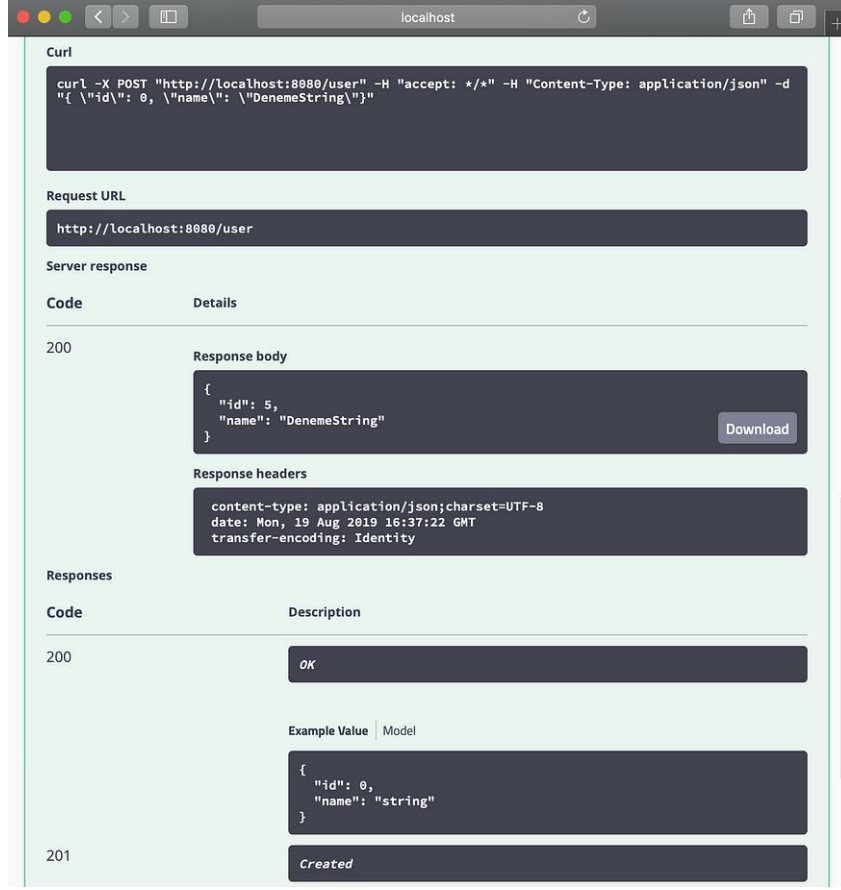
Cancel

Parameter content type  
application/json

Execute

Responses  
Response content type \*/\*

Post işleminin 2. Adımı[2]



Post işleminin 3.adımı[2]

TAHA GÖKPINAR

## Kaynakça

[1] <https://www.techtarget.com/searchapparchitecture/definition/Swagger#>

[2] <https://medium.com/android-t%C3%BCrkiye/swagger-nedir-ne-i%C5%9Feyarar-e8c12a9e9e7f>

[3] Chat-GPT