

Swagger

Swagger nedir?

Swagger OpenAPI Spesifikasyonu etrafında oluşturulmuş açık kaynaklı araçlardan oluşan REST API'lerini tasarlamaya, oluşturmaya, belgelemenize ve tüketmenize yardımcı olabilecek araçlar kümesidir.

Swagger, API'lerin planlanması, yapılandırılması, belgelenmesi ve test edilmesi için kullanılır. Swagger, API'lerin açık bir şekilde tanımlanmasını sağlar, bu da geliştiricilerin ve diğer ilgili tarafların API'leri daha kolay anlamasına, test etmesine ve kullanmasına olanak tanır.

Bazı **major Swagger araçları** şunlardır:

- ❑ **Swagger Editor:** Swagger Editor, API tanımlarını oluşturmak ve düzenlemek için kullanılan bir web tabanlı araçtır. Geliştiricilere, API'lerini kolayca tasarlama, düzenleme ve doğrulama imkanı sunar. Swagger Editor, JSON ve YAML formatlarında çalışabilir ve hataları tespit etmeyi kolaylaştırmak için anında geri bildirim sağlar.
- ❑ **Swagger UI:** Swagger UI, API'lerin interaktif ve görsel dokümantasyonunu sunan bir araçtır. Swagger veya OpenAPI spesifikasyonlarına dayanarak, API kullanıcıları ve geliştiricilerin API'leri daha iyi anlamalarına, test etmelerine ve kullanmalarına yardımcı olur. Swagger UI kullanarak, kullanıcıların API çağrılarını doğrudan tarayıcıda deneyebilecekleri etkileşimli API belgeleri oluşturulabilir. Swagger UI'nin kullanıldığı bir örnek için <https://petstore.swagger.io/> sitesini ziyaret edebilirsiniz.
- ❑ **Swagger Codegen:** Swagger Codegen, Swagger veya OpenAPI spesifikasyonlarına dayanarak istemci SDK'ları, sunucu iskeletleri ve API dokümantasyonu oluşturmak için kullanılan bir araçtır. Bu araç, farklı programlama dilleri ve çerçeveler için otomatik olarak kod üretebilir, böylece geliştiricilerin API entegrasyonlarını ve uygulama geliştirmelerini hızlandırmalarına yardımcı olur.
- ❑ **Swagger Core:** Swagger Core, Java tabanlı uygulamalar için Swagger/OpenAPI spesifikasyonlarını oluşturmak ve yönetmek amacıyla kullanılan bir kütüphanedir. RESTful API'leri Java koduyla tanımlamanıza ve bu tanımları Swagger/OpenAPI spesifikasyonları olarak dışa aktarmanıza olanak tanır. Bu kütüphane, API dokümantasyonunu otomatik olarak oluşturmak için **Java anotasyonlarını** kullanır.

JWT, Basic Auth yapıları swagger ile nasıl test edilir?

Swagger, API'lerinizin güvenliğini sağlamak için farklı kimlik doğrulama yöntemlerini destekler. JWT (JSON Web Token) ve Basic Auth, Swagger ile test edebileceğiniz iki yaygın kimlik doğrulama yöntemidir.

JWT(Json Web Token)

JWT (JSON Web Token), genellikle internet üzerinde bilgi alışverişi yaparken kullanılan bir açık standarttır. JSON formatında veri alışverişi yapmak için tasarlanmıştır ve token tabanlı kimlik doğrulama ve yetkilendirme işlemleri için kullanılır.

JWT, üç bölümden oluşur:

Header (Başlık): Algoritma ve token tipi bilgilerini içerir. Örneğin:

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

Payload (Yük): Token içinde taşınan asıl veriyi içerir. Bu veri, kullanıcı bilgileri, tokenın geçerlilik süresi gibi bilgileri kapsayabilir. Örneğin:

```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "admin": true  
}
```

Signature (İmza): Tokenın bütünlüğünü ve doğruluğunu sağlamak için kullanılır. İmza, header ve payload kısımlarının birleştirilip belirli bir algoritma ile şifrelenmesiyle oluşturulur. Örneğin, HMAC SHA256 algoritması ile:

```
HMACSHA256(  
  base64UrlEncode(header) + "." + base64UrlEncode(payload),  
  secret  
)
```

Bir JWT şu şekilde görünür:

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjOnRydWV9.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c

Bearer kimlik doğrulama (token kimlik doğrulaması olarak da adlandırılır), bearer token adı verilen güvenlik tokenlerini içeren bir HTTP kimlik doğrulama şemasıdır. "Bearer kimlik doğrulama" ismi, "bu tokenin taşıyıcısına erişim ver" olarak anlaşılabilir. Bearer token, genellikle bir giriş isteğine yanıt olarak sunucu tarafından oluşturulan kriptik bir dizidir. İstemci, korunan kaynaklara istek yaparken bu tokeni Authorization başlığında göndermelidir.

Bearer kimlik doğrulaması sadece HTTPS (SSL) üzerinden kullanılmalıdır.

JWT'yi Swagger ile test etmek için öncelikle Swagger şemanızı yapılandırmanız.

```
1. openapi: 3.0.0
2. ...
3.
4. # 1) Define the security scheme type (HTTP bearer)
5. components:
6.   securitySchemes:
7.     bearerAuth:      # arbitrary name for the security scheme
8.       type: http
9.       scheme: bearer
10.      bearerFormat: JWT # optional, arbitrary value for documentation purposes
11.
12. # 2) Apply the security globally to all operations
13. security:
14.   - bearerAuth: []      # use the same name as above
```

Bu görselde bir swagger şeması örneği verilmiştir.

Components bölümünde, güvenlik düzenleri için **bearerAuth** kullanılacağı ve bu işlemin **HTTP** protokolü kullanılarak gerçekleştirileceği belirtilmiştir. Ayrıca, formatın **JWT** olduğu ifade edilmiştir.

Security kısmında, Bearer kimlik doğrulamasının API'nin genelinde uygulandığı belirtilmektedir. Eğer Bearer kimlik doğrulamasını yalnızca birkaç işlem için uygulamak isterseniz, işlem düzeyinde güvenlik eklemelisiniz. Bu işlem aşağıdaki gibi yapılabilir.

```
1. paths:
2.   /something:
3.     get:
4.       security:
5.         - bearerAuth: []
```

JWT ayrıca diğer kimlik doğrulama metodlarıyla kombinlenerek kullanılabilir.

Basic Auth

Basic Authentication, HTTP protokolünde kimlik doğrulama yöntemlerinden biridir. Bu yöntem, kullanıcı adı ve şifre bilgilerini base64 ile kodlayarak HTTP başlıklarında gönderir. Basic Authentication, özellikle basit ve hızlı bir kimlik doğrulama yöntemi olarak bilinir. Ancak, veri şifrelenmediği için güvenlik açıkları bulunur ve **sadece HTTPS** (SSL/TLS) üzerinden kullanılması önemlidir.

```
1. openapi: 3.0.0
2. ...
3.
4. components:
5.   securitySchemes:
6.     basicAuth: # <-- arbitrary name for the security scheme
7.       type: http
8.       scheme: basic
9.
10.  security:
11.    - basicAuth: [] # <-- use the same name here
```

Bu görselde bir swagger şeması örneği verilmiştir.

Components bölümünde, güvenlik düzenleri için **basicAuth** kullanılacağı ve bu işlemin **HTTP** protokolü kullanılarak gerçekleştirileceği belirtilmiştir.

Security kısmında, basicAuth kimlik doğrulamasının API'nin genelinde uygulandığı belirtilmektedir. Eğer basicAuth kimlik doğrulamasını yalnızca birkaç işlem için uygulamak isterseniz, işlem düzeyinde güvenlik eklemelisiniz. Bu işlem aşağıdaki gibi yapılabilir.

```
1. paths:
2.   /something:
3.     get:
4.       security:
5.         - basicAuth: []
```

Basic auth ayrıca diğerkimlik doğrulama metodlarıyla kombinlenerek kullanılabilir.

Swagger Nasıl Test Edilir?

✓ Swagger UI :

Swagger UI kullanarak manuel test yapmak oldukça basit ve kullanıcı dostu bir süreçtir. Aşağıdaki adımları takip ederek Swagger UI kullanarak test yapabilirsiniz.

1. API Endpoint'lerini Görüntüleyin: Swagger UI, Swagger/OpenAPI belgenizi yükledikten sonra API'nizdeki tüm endpoint'leri listeleyecektir. Her bir endpoint, açıklamalar ve parametrelerle birlikte gösterilir.
2. "Try it out" Butonunu Kullanın: Test etmek istediğiniz endpoint'in üzerine gelin ve "Try it out" butonuna tıklayın.
3. Parametreleri Girin: Endpoint'in gerektirdiği parametreleri girin. Bu parametreler sorgu parametreleri, başlıklar veya gövde verileri olabilir. Örneğin, bir GET isteği için sorgu parametrelerini girmeniz gerekebilir. Bir POST isteği için ise gövde verilerini JSON formatında girmeniz gerekebilir.
4. İsteği Gönderin: Girdiğiniz parametrelerden sonra "Execute" butonuna tıklayın. Swagger UI, isteği gönderecek ve yanıtı gösterecektir.
5. Yanıtı İnceleyin: Swagger UI, API isteğinin yanıtını, HTTP durum kodunu, başlıkları ve yanıt gövdesini gösterecektir. Bu bilgileri inceleyerek API'nizin beklenen şekilde çalışıp çalışmadığını doğrulayabilirsiniz.

✓ Swagger Codegen:

Swagger Codegen, Swagger/OpenAPI belgelerinizden test kodları oluşturabilir. Adımlar:

1. Swagger Codegen'i kurun.
2. Swagger/OpenAPI belgenizi kullanarak test kodları oluşturun.
3. Oluşturulan test kodlarını çalıştırın ve sonuçları inceleyin.

✓ Postman ve Newman:

Postman, API testleri yazmak ve çalıştırmak için kullanımı kolay bir araçtır. Newman, Postman koleksiyonlarınızı komut satırından çalıştırmanızı sağlar.

Adımlar:

1. Swagger/OpenAPI belgenizi Postman'e import edin.
2. Postman'de test senaryolarınızı oluşturun.

3. Newman ile testleri otomatikleřtirin ve komut satırından alıřtırın.

Mustafa Koer

Kaynakça:

- <https://swagger.io/docs/specification/authentication/basic-authentication/>
- <https://swagger.io/docs/specification/about/>
- <https://swagger.io/docs/specification/authentication/bearer-authentication/>
- <https://petstore.swagger.io/>
- <https://bsseylcin.medium.com/api-swagger-nedi%CC%87r-5766c5c8c2d0>