

Swagger

Swagger Nedir ve Nasıl Kullanılır?

Swagger, SmartBear Yazılımından API geliştiricilerine yönelik bir araç paketidir ve OpenAPI tanımlamasının dayandığı eski bir spesifikasyondur.

Sıfırdan bir API oluşturduysanız, muhtemelen kullanıcının başarılı olması için ihtiyaç duyacağı tüm gerekli bilgileri ekleyip eklemediğinizi sorgulamışsınızdır. Swagger, yönergelerle uyduğunuzdan emin olmak için OpenAPI spesifikasyonunu çalıştırarak API belgeleri yazarken çok fazla zaman tasarrufu sağlar. OpenAPI ve Swagger tarafından yapılan otomasyonlar sayesinde, geliştiricilerin API dokümantasyon sürecini oluşturması ve sürdürmesi çok daha kolaydır.

Swagger API, programcıların REST API'lerini *geliştirmesine, tasarlamasına, belgelemesine* ve *kullanmasına* yardımcı olmak için oluşturulmuş bir dizi açık kaynaklı araçtır. Araç, OpenAPI spesifikasyonu etrafında oluşturulmuştur ve üç bileşen içerir:

- **Swagger Editor (Api'ları geliştirme):** Geliştiricilerin yeni API'ler için belgeler yazmasına, tasarlamasına ve açıklamasına ve mevcut API'leri düzenlemesine olanak tanır. Tarayıcı tabanlı düzenleyici, OpenAPI özelliklerini görsel olarak işler, hataları yönetir ve gerçek zamanlı geri bildirim sağlar. API'ler oluştururken, kodun kendisine dayalı olarak otomatik olarak bir Open API belgesi oluşturmak için Swagger araçları kullanılabilir. Bu, API açıklamasını bir projenin kaynak koduna yerleştirir ve gayri resmi olarak kod öncelikli veya aşağıdan yukarıya API geliştirme olarak adlandırılır.
- **Swagger UI(Api'leri dokümante etme):** Geliştiricilerin çeşitli platformlar için dokümantasyon oluşturmalarına yardımcı olan tamamen özelleştirilebilir bir araçtır. Her ortamda barındırılabilir. Bir OpenAPI belgesi tarafından tanımlandığında, Swagger açık kaynak araçları, Swagger UI (<https://github.com/swagger-api/swagger-ui>) aracılığıyla API ile doğrudan etkileşimde bulunmak için kullanılabilir. Bu proje, etkileşimli, HTML tabanlı bir kullanıcı arayüzü aracılığıyla doğrudan canlı API'lere bağlantı kurulmasına olanak tanır. İstekler doğrudan kullanıcı arayüzünden ve arayüz kullanıcısı tarafından keşfedilen seçeneklerden yapılabilir.
Bir örneğini görmek için https://petstore.swagger.io/?_ga=2.254799371.200266998.1715629421-139400268.1715629421 burayı inceleyebilirsiniz.
- **Swagger Codegen(API'lerle etkileşim):** Geliştiricilere farklı platformlar için istemci kitaplığı kodu ve SDK'lar oluşturma olanağı sağlar. Son kullanıcılar, doğrudan OpenAPI belgesinden istemci SDK'ları oluşturarak insan tarafından oluşturulan (human-generated) istemci kodu ihtiyacını azaltır.

Swagger'ın amacı, REST API'leri için standart, dilden bağımsız bir arayüz tanımlamaktır; bu arayüz, hem insanların hem de bilgisayarların, kaynak koduna, belgelere veya ağ trafiği incelemesine erişim olmadan hizmetin yeteneklerini keşfetmesine ve anlamasına olanak tanır. REST API ' da var olan kaynakların listesini ve bu kaynaklarda çağrılabilen işlemleri belirtir. Swagger belgesi ayrıca, parametrelerin adı ve tipi, parametrelerin zorunlu mu, yoksa isteğe bağlı mı olduğu ve bu parametrelere ilişkin kabul edilebilir değerlerle ilgili bilgiler de içinde

olmak üzere, bir işleme ilişkin parametrelerin listesini belirtir. Buna ek olarak, Swagger belgesi, REST API 'sindeki bir işleme gönderilen istek gövdesinin yapısını açıklayan JSON Şeması içerebilir ve JSON şeması, bir işlemten döndürülen yanıt gövdelerinin yapısını açıklar.

Swagger belgeleri, düzenlemelerin daha kolay yapılabilmesi için.json dosya uzantısıyla JSON biçiminde ya da .yaml ya da .yml dosya uzantısıyla YAML biçiminde olmalıdır.

Swagger araçlarının çoğu, bir Swagger belgesini, HTTP sunucusunda bulunan bir Swagger belgesini gösteren bir URL olarak kabul eder.

JWT, Basic Auth Gibi Kimlik Doğrulama Yapıları Swagger ile Nasıl Test Edilir?

Swagger, API'ler için kapsamlı testler oluşturulmasına olanak tanır. Bu testler, API'lerin işlevselliğini ve güvenliğini doğrulanmasına yardımcı olur.

JWT(Json Web Token)

API'lerde kullanıcıların kimliklerini doğrulamak için yaygın olarak kullanılan bir kimlik doğrulama yöntemidir. OpenAPI için JWT kimlik doğrulamasını operation, class veya global levele göre yapılandırabiliriz.

1- Swagger Editor'de OpenAPI tanımları yazma: Tüm proje konfigürasyonlarını içeren bir klasöre sahip olmak idealdir. Konfigürasyon klasörünüzde OpenAPI spesifikasyonlarınızı tanımlayacağınız bir config dosyası oluşturabilirsiniz.

2- JWT belirteçleri için güvenlik şemalarını (securitySchemes) tanımlama:

```
components:
  securitySchemes:
    bearerAuth: # arbitrary name for the security scheme
    type: http
    scheme: bearer
    bearerFormat: JWT # optional, arbitrary value for documentation purposes
```

Güvenlik şemasının HTTP kimlik doğrulaması kullandığı belirtiliyor

3- JWT tokenı oluşturmak için endpoint oluşturma: JWT tokenlarını oluşturmak için bir endpoint oluşturmak genellikle sizin kontrolünüzde olan bir işlemdir ve OpenAPI spesifikasyonuna doğrudan dahil edilmez. Bununla birlikte, API'nizin nasıl yetkilendirileceğini ve JWT tokenlarının nasıl alınacağını belirtmek için OpenAPI spesifikasyonunu kullanabilirsiniz. Bu genellikle "securityDefinitions" veya "security" bölümünde yapılır ve yetkilendirme yönteminizi (örneğin, JWT) ve yetkilendirme için gerekli bilgileri belirtir.

```
security:
  - ApiKeyAuth: []
  - OAuth2:
    - read
    - write
# The syntax is:
# - scheme name:
#   - scope 1
#   - scope 2
```

4- JWT Test Örneği Oluşturma: Swagger Editor'da, JWT kimlik doğrulama tokenıyla bir istek oluşturmak için bir test örneği oluşturun. Bu test örneği, isteğin gövdesini, başlıklarını ve kimlik doğrulama bilgilerini içermelidir.

5- Testi Çalıştırma: Swagger Editor'u kullanarak testi çalıştırın ve sonucu inceleyin. Test başarılıysa, API'niz JWT kimlik doğrulamasını doğru şekilde kullanıyor demektir.

Basic Auth

Basic Auth, API'lerinizde kullanıcıları kimlik doğrulamak için isminden de anlaşıldığı gibi daha basit bir yöntemdir. JWT'den farklı olarak securitySchemes nesnesi Basic Auth kimlik doğrulamasının türünü (basic) ve kimlik doğrulama parametrelerini (in, name) şeklinde belirtmelidir.

```
components:
  securitySchemes:
    BasicAuth:
      type: http
      scheme: basic
```

Swagger Nasıl Test Edilir?

Swagger ile API'lerinizi test etmenin iki ana yöntemi vardır:

1. Swagger UI ile Manuel Test:

Swagger UI, API'nizin etkileşimli dokümantasyonunu görüntülemenizi sağlayan bir araçtır. Bu dokümantasyon, API'nizin tüm uç noktalarını, parametrelerini, istek/yanıt yapısını ve kimlik doğrulama gereksinimlerini gösterir. Swagger UI'yi kullanarak manuel olarak test yapabilirsiniz:

- **İstek Gönderme:** Her bir uç noktası için farklı istek yöntemlerini (GET, POST, PUT, DELETE) seçebilir ve gerekli parametreleri girebilirsiniz. Swagger UI, isteği API'nize gönderecek ve yanıtı görüntüleyecektir.
- **Yanıt Doğrulama:** Yanıtın beklenen HTTP durum kodunu (örneğin, 200 başarılı, 401 yetkisiz) ve veri yapısını karşılayıp karşılamadığını kontrol edebilirsiniz.
- **Kimlik Doğrulama Testleri:** Basic Auth veya JWT gibi kimlik doğrulama mekanizmalarını kullanıyorsanız, uygun kimlik bilgilerini Swagger UI'ye girebilir ve API'nin yetkisiz erişim girişimlerini doğru şekilde reddetmesini doğrulayabilirsiniz.

2. Swagger Editor ile Otomatik Testler:

Swagger Editor, API'nizin OpenAPI spesifikasyonunu yazmanıza ve düzenlemenize olanak tanıyan bir araçtır. Bu spesifikasyon, API'nizin yapısı ve davranışı hakkında ayrıntılı bilgi içerir. Swagger Editor, otomatik testler oluşturmak için çeşitli araç ve kütüphanelerle birlikte kullanılabilir. Bu araçlar, OpenAPI spesifikasyonunu temel alarak API'nizin tüm uç noktalarını ve işlevlerini otomatik olarak test edebilir.

3- Postman ile test etme:

Postman, popüler bir API istemci aracıdır. Swagger spesifikasyonlarını içe aktarabilir ve otomatik test koleksiyonları oluşturmak için kullanabilirsiniz.

4- Swagger Inspector: Swagger ekibi tarafından geliştirilen bir araçtır. Swagger spesifikasyonlarını doğrulamak ve otomatik testler oluşturmak için kullanılabilir.

Bengisu Şahin

Referanslar

[https://en.wikipedia.org/wiki/Swagger_\(software\)](https://en.wikipedia.org/wiki/Swagger_(software))

<https://blog.hubspot.com/website/what-is-swagger>

<https://www.techtarget.com/searchapparchitecture/definition/Swagger>

<https://swagger.io/docs/open-source-tools/swagger-editor/>

<https://swagger.io/docs/specification/authentication/>

<https://github.com/swagger-api/swagger-ui>