

# Swagger

Swagger, API (Application Programming Interface - Uygulama Programlama Arayüzü) belirtme ve belgeler oluşturma aracıdır. API'ler, yazılım uygulamaları arasında veri alışverişi yapmanın standart yoludur. Swagger, bu alışverişi daha anlaşılır ve belgelenebilir hale getirir.

API'ler, yazılım geliştiricilerin diğer sistemlerle entegrasyon yapmasını sağlayan temel yapı taşlarıdır. Ancak, API'leri anlamak ve kullanmak, bazen karmaşık olabilir. Swagger, bu karmaşıklığı azaltır ve API'lerin kullanımını daha basit hale getirir.

Swagger belgeleri, API'lerin nasıl kullanılacağı, hangi isteklerin yapılabileceği ve hangi yanıtların alınabileceği gibi bilgileri açıkça belirtir. Bu belgeler, API'nin işlevselliğini ve kullanımını tanımlayan açık bir sözleşme gibidir. Bu da geliştiricilerin API'leri daha hızlı ve daha güvenli bir şekilde kullanmalarını sağlar.

Swagger ayrıca, belgeleri otomatik olarak oluşturmanıza ve API'nin farklı platformlarda nasıl kullanılacağını gösteren client ve server kodlarını otomatik olarak üretmenize olanak tanır. Bu da yazılım geliştirme sürecini hızlandırır ve hataları azaltır.

Ayrıca, Swagger UI adı verilen bir arayüzü vardır. Bu arayüz, Swagger belgelerini görsel olarak sunar ve API'yi test etmek için bir deneme ortamı sağlar. Bu ortamda, API'yi deneyebilir, istekler yapabilir ve yanıtları kontrol edebilirsiniz.

JWT (JSON Web Token) ve Basic Authentication gibi güvenlik yapıları da Swagger ile test edilebilir. Swagger UI, bu tür kimlik doğrulama mekanizmalarını destekler ve test edebilirsiniz. Böylece, API'nin güvenliğini ve doğruluğunu doğrulayabilirsiniz.

## Swagger ile JWT ve Basic Auth yapılarını test etmek

1. **Swagger Belgesi Oluşturma:** İlk adım, API'nin Swagger belgesini oluşturmak veya mevcut bir Swagger belgesini kullanmaktır. Swagger belgesi, API'nin endpoint'lerini, parametrelerini, istek ve yanıt türlerini vb. açıklar.
2. **Gerekli Kimlik Doğrulama Şemalarını Belirtme:** Swagger belgesinde, API'nin JWT veya Basic Auth gibi kimlik doğrulama yöntemlerini nasıl kullanacağını belirtmelisiniz. Bu, her endpoint için ayrı ayrı yapılabilir veya genel olarak tüm API için belirtilebilir.
3. **Swagger UI Kullanma:** Swagger belgesini görsel bir arayüzde sunan Swagger UI'yi kullanın. Bu arayüzde, API'yi test etmek için kullanabileceğiniz bir deneme ortamı bulunur. Swagger UI, belirlediğiniz kimlik doğrulama yöntemlerini kullanarak istek yapmanıza olanak tanır.
4. **JWT ile Kimlik Doğrulama:** Swagger UI üzerinden JWT ile kimlik doğrulama yapmak için, JWT token'ını Authorization başlığı altında göndermeniz gerekir. Swagger UI'de, isteği yaparken Authorization başlığına "Bearer {JWT Token}" formatında token'ı ekleyebilirsiniz. Bu şekilde, API'ye istek yaparken doğru kimlik doğrulamasını sağlayabilirsiniz.

5. **Basic Auth ile Kimlik Doğrulama:** Basic Auth ile kimlik doğrulama yapmak için, Swagger UI'de isteği yaparken kullanıcı adı ve parolayı Authorization başlığı altında Base64 kodlanmış olarak göndermeniz gerekir. Authorization başlığına "Basic {Base64(KullanıcıAdı:Parola)}" formatında bilgileri ekleyerek isteği gönderebilirsiniz.
6. **İstek Yapma ve Yanıtı Kontrol Etme:** Belirlediğiniz kimlik doğrulama yöntemini kullanarak API'ye istek yapın ve gelen yanıtı kontrol edin. Doğru kimlik doğrulaması yapıldığında beklenen yanıtı almalısınız. Eğer yanıt beklenenden farklıysa veya hata varsa, API'nin kimlik doğrulama mekanizmasını tekrar gözden geçirin ve hatayı gidermeye çalışın.

```
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
```

```
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
```