



# Asp.Net ile Web Programlama

Doç. Dr. Zafer CÖMERT



**Bölüm  
9**

**Web API'ye Giriş**

# Giriş

## İçerik

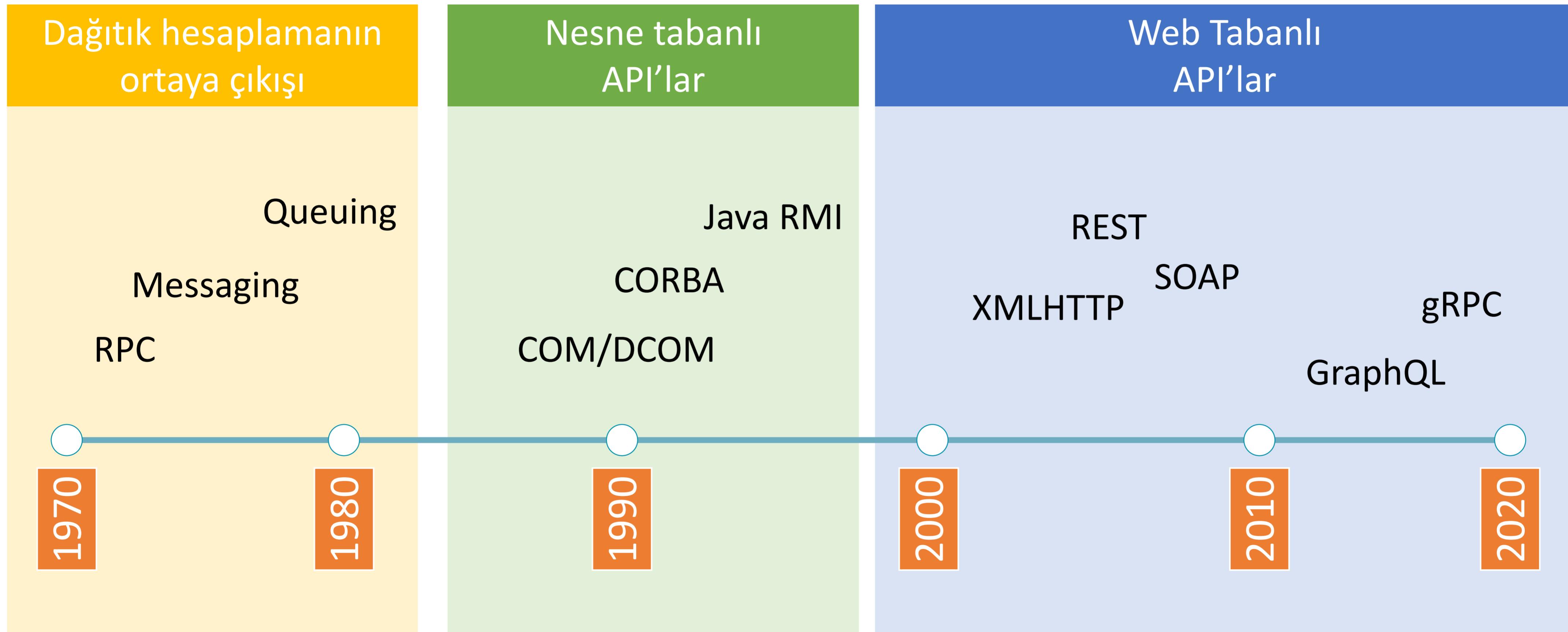
- API
- HTTP
  - Request
    - Http Verbs
    - Headers
    - Body
  - Response
    - StatusCode
    - Headers
    - Body
- REST
- RESTful API Tasarımı
- curl
- OpenAPI ve Swagger
- NewsArticle (CRUD)

# ASP.NET Core Web API

## API

- Application Programming Interface (API)'ler, tümleşik (integrated) web uygulamaları geliştirmek ve kurmak için bir dizi tanımlar ve protokollerden oluşan bir uygulama programlama arayüzüdür.
- API sizin ürün ya da servisinizin, uygulama detayları ve teknoloji tercihlerinden bağımsız olarak başka ürün ya da servislere ile iletişim kurmasına olanak tanır.

# Dağıtık API'ların Tarihi



# Representational State Transfer\*

curl

- Client URL.
- Unix bazlı sistemde mevcut olan bir komuttur.
- Curl komutları URL'lerin bağlanabilirliğini kontrol etmek ve veri transferi için harika bir araç olarak kullanılmak için üretilmiştir.
- Curl, istemci tarafından ücretsiz bir URL transfer kütüphanesidir.
- HTTP, HTTPS, FTP, FTPS, IMAP, IMAPS, SFTP, TELNET, GOPHER, LDAP, LDAPS, SMTP, SMTPS gibi protokollerini destekler.

# Representational State Transfer\*

curl

- Client URL.
- Unix bazlı sistemde mevcut olan bir komuttur.
- Curl komutları URL'lerin bağlanabilirliğini kontrol etmek ve veri transferi için harika bir araç olarak kullanılmak için üretilmiştir.
- Curl, istemci tarafından ücretsiz bir URL transfer kütüphanesidir.
- HTTP, HTTPS, FTP, FTPS, IMAP, IMAPS, SFTP, TELNET, GOPHER, LDAP, LDAPS, SMTP, SMTPS gibi protokollerini destekler.

# Representational State Transfer\*

curl

- Client URL.
- Unix bazlı sistemde mevcut olan bir komuttur.
- Curl komutları URL'lerin bağlanabilirliğini kontrol etmek ve veri transferi için harika bir araç olarak kullanılmak için üretilmiştir.
- Curl, istemci tarafından ücretsiz bir URL transfer kütüphanesidir.
- HTTP, HTTPS, FTP, FTPS, IMAP, IMAPS, SFTP, TELNET, GOPHER, LDAP, LDAPS, SMTP, SMTPS gibi protokollerini destekler.

# Representational State Transfer\*

curl

- Client URL.
- Unix bazlı sistemde mevcut olan bir komuttur.
- Curl komutları URL'lerin bağlanabilirliğini kontrol etmek ve veri transferi için harika bir araç olarak kullanılmak için üretilmiştir.
- **Curl, istemci tarafından ücretsiz bir URL transfer kütüphanesidir.**
- HTTP, HTTPS, FTP, FTPS, IMAP, IMAPS, SFTP, TELNET, GOPHER, LDAP, LDAPS, SMTP, SMTPS gibi protokollerini destekler.

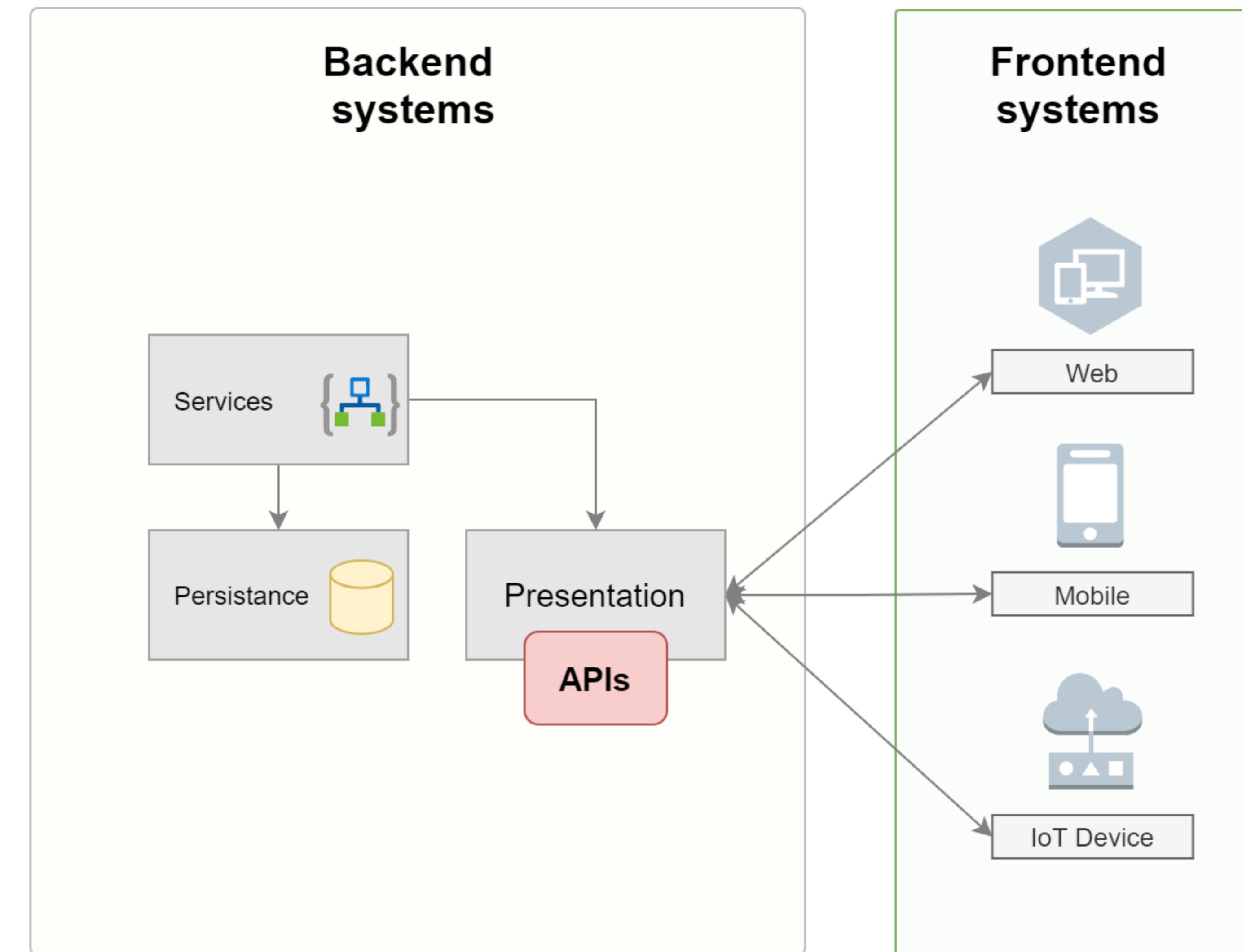
# Representational State Transfer\*

curl

- Client URL.
- Unix bazlı sistemde mevcut olan bir komuttur.
- Curl komutları URL'lerin bağlanabilirliğini kontrol etmek ve veri transferi için harika bir araç olarak kullanılmak için üretilmiştir.
- Curl, istemci tarafından ücretsiz bir URL transfer kütüphanesidir.
- HTTP, HTTPS, FTP, FTPS, IMAP, IMAPS, SFTP, TELNET, GOPHER, LDAP, LDAPS, SMTP, SMTPS gibi protokollerini destekler.

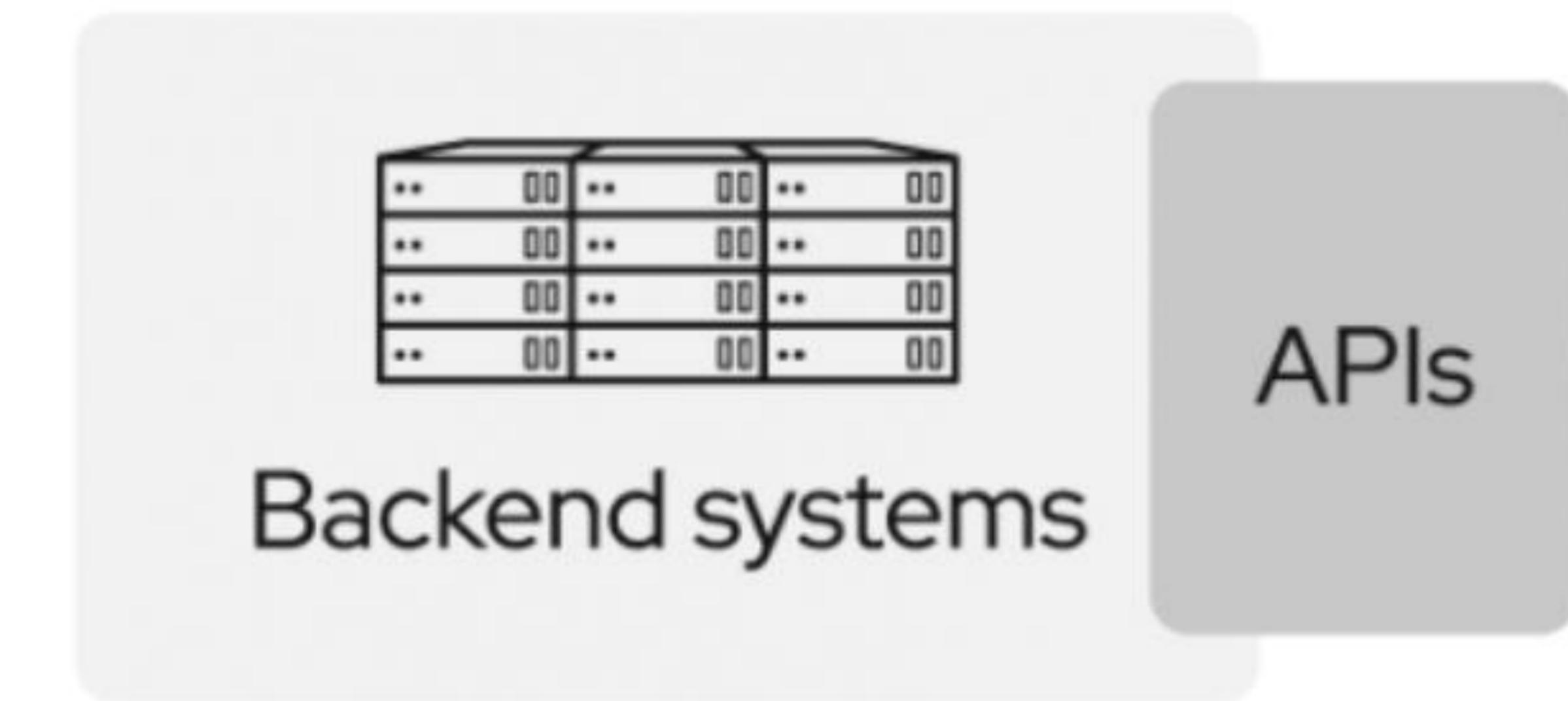
# ASP.NET Core Web API

API



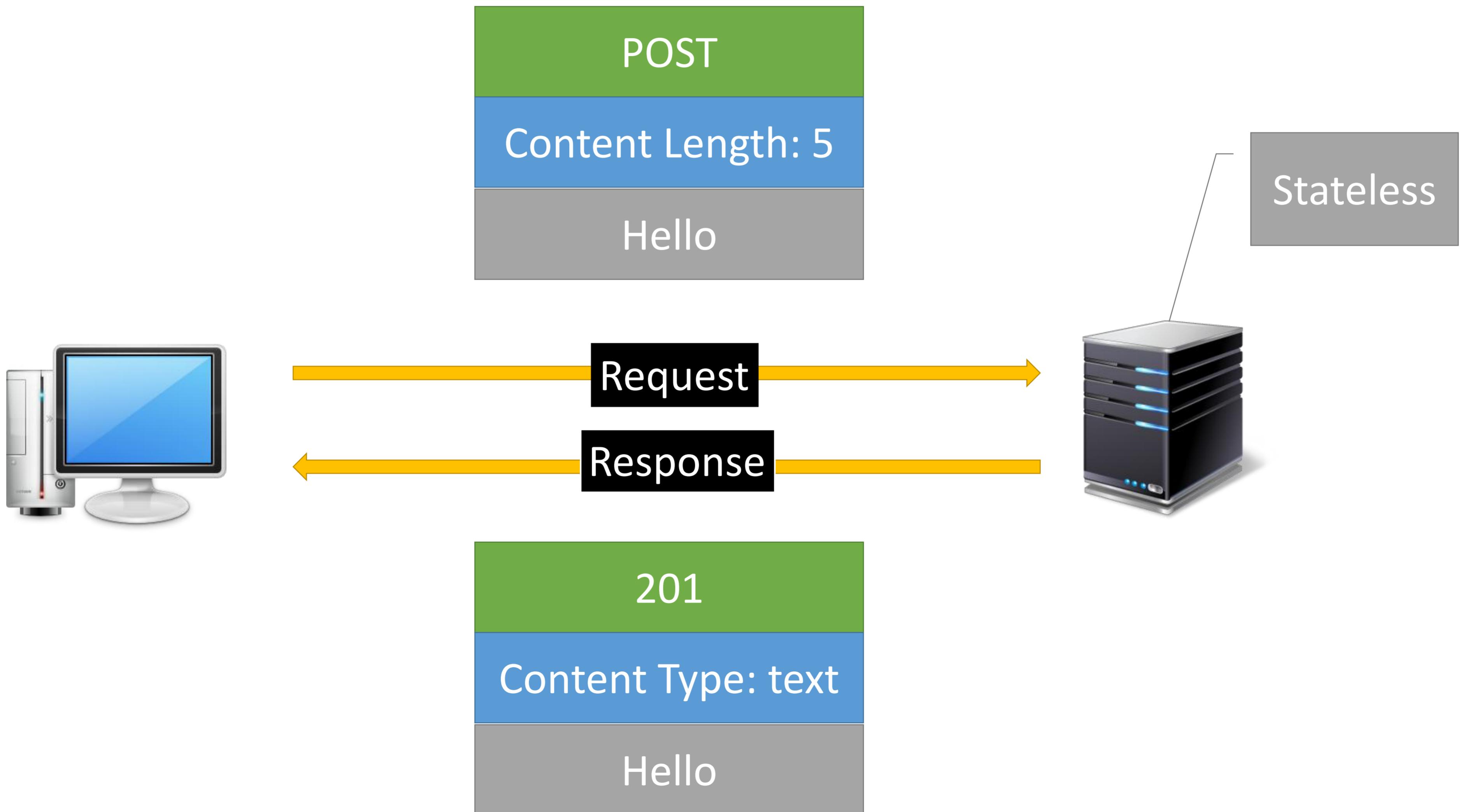
# ASP.NET Core Web API

API



- Backend as a service (BaaS)
- Backend as a function (BaaF)

# HTTP Nasıl Çalışır?



# REST API Bileşenleri

VERB

URI (Query String)

Headers

Request Body

# REST API Bileşenleri

Status Code

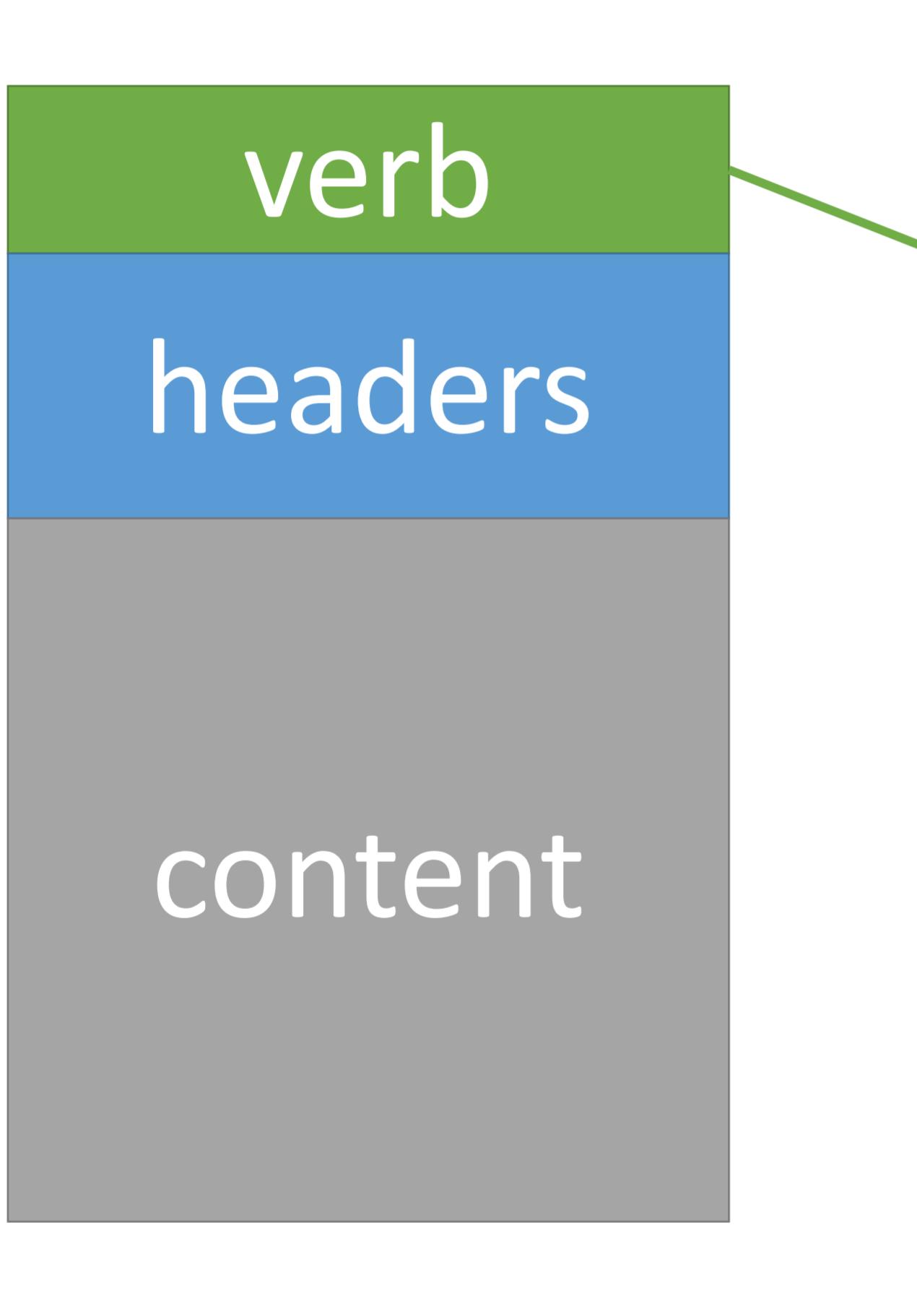
Headers

Response Body

# HTTP Status Codes

Kod Aralığı	Kategori	Örnek Kodlar	Anlamı	Açıklama / Kullanım Senaryosu
1xx	Bilgilendirme (Informational)	100 Continue, 101 Switching Protocols	İstek alındı, işlem devam ediyor	Genellikle tarayıcı veya proxy tarafından iç iletişimde kullanılır; son kullanıcıya gösterilmez.
2xx	Başarılı (Success)	200 OK, 201 Created, 202 Accepted, 204 No Content	İstek başarıyla işlendi	- <b>200 OK</b> → GET isteği başarıyla sonuçlandı. - <b>201 Created</b> → Yeni kaynak oluşturuldu (POST sonrası). - <b>204 No Content</b> → Başarılı fakat donecek veri yok (DELETE veya PUT sonrası).
3xx	Yönlendirme (Redirection)	301 Moved Permanently, 302 Found, 304 Not Modified	Kaynak başka bir yere yönlendiriliyor	- <b>301</b> → Kalıcı yönlendirme. - <b>302</b> → Geçici yönlendirme. - <b>304</b> → İçerik değişmedi, önbellekten kullanılabilir.
4xx	İstemci Hatası (Client Error)	400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, 409 Conflict, 422 Unprocessable Entity	İstek hatalı veya yetkisiz	- <b>400</b> → Geçersiz istek (yanlış veri, biçim). - <b>401</b> → Kimlik doğrulaması gereklidir. - <b>403</b> → Erişim yetkisi yok. - <b>404</b> → Kaynak bulunamadı. - <b>409</b> → Çakışma (ör. aynı kayıt iki kez gönderildi). - <b>422</b> → Doğrulama hatası.
5xx	Sunucu Hatası (Server Error)	500 Internal Server Error, 502 Bad Gateway, 503 Service Unavailable, 504 Gateway Timeout	Sunucuda hata oluştu	- <b>500</b> → Genel sunucu hatası. - <b>502</b> → Geçersiz ara sunucu yanıtı. - <b>503</b> → Sunucu geçici olarak hizmet dışı. - <b>504</b> → Sunucu yanıt süresi aşındırıldı.

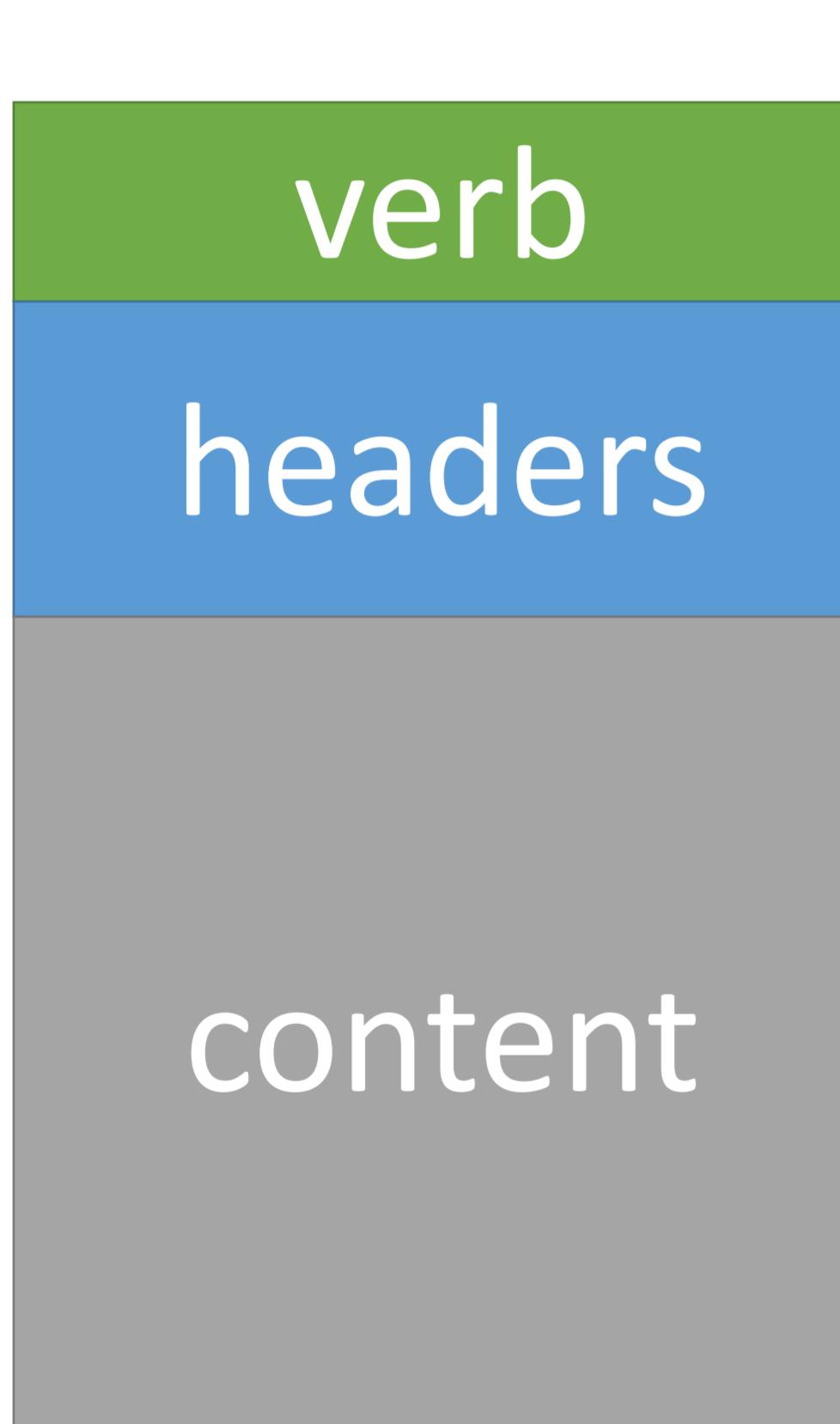
# İsteğin (Request) Yapısı



## Sunucuda gerçekleştirilen eylemler:

- **GET**
  - Kaynak isteme
- **POST**
  - Kaynak oluşturma
- **PUT**
  - Kaynak güncelleme
- **PATCH**
  - Kismi kaynak güncellemesi
- **DELETE**
  - Kaynak silme
- ve daha fazla eylem...

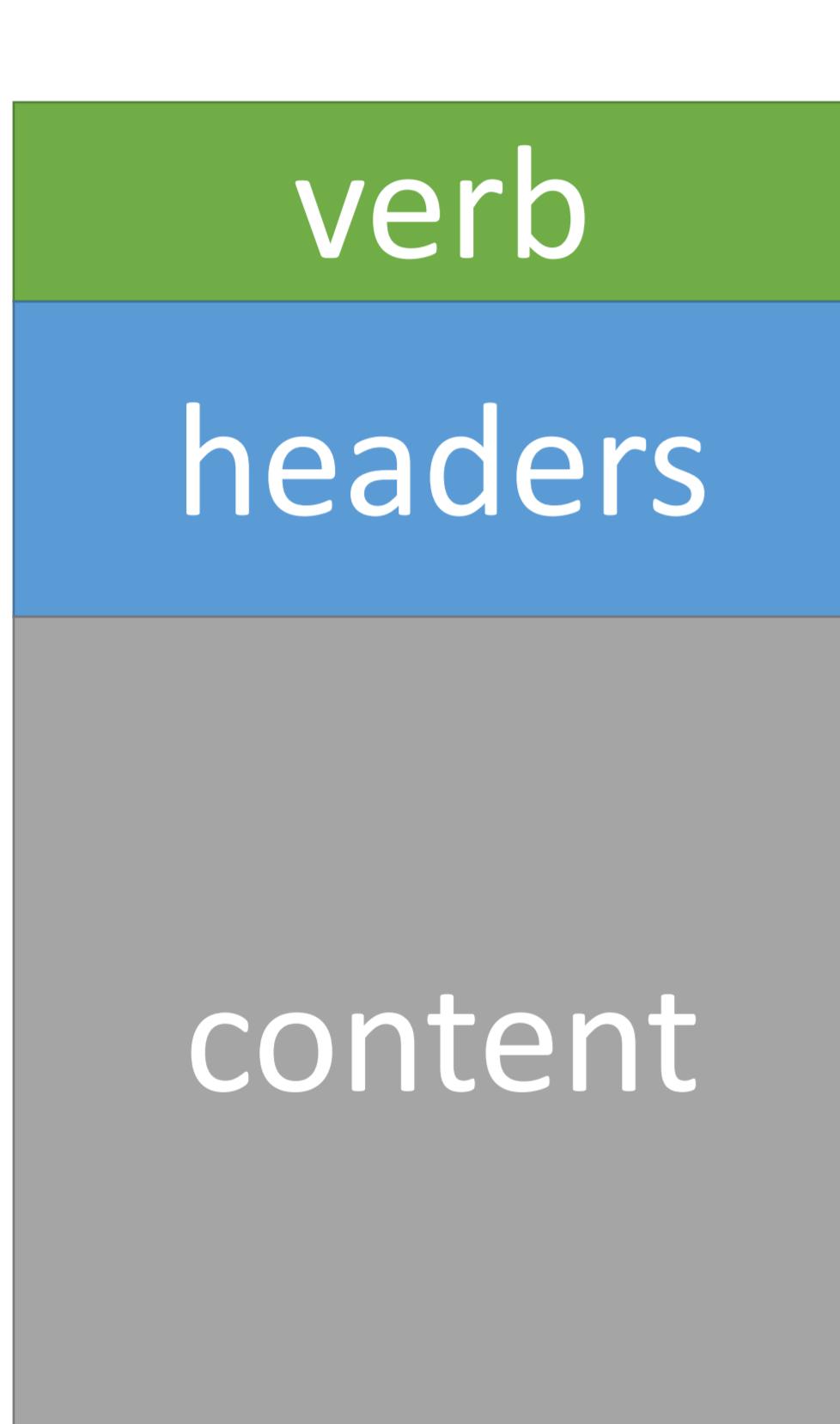
# İsteğin (Request) Yapısı



## İstek hakkında üst (meta) bilgiler:

- **Content Type**
  - İçeriğin formatı
- **Content Length**
  - İçeriğin boyutu
- **Authorization**
  - İsteği yapanın kimliği

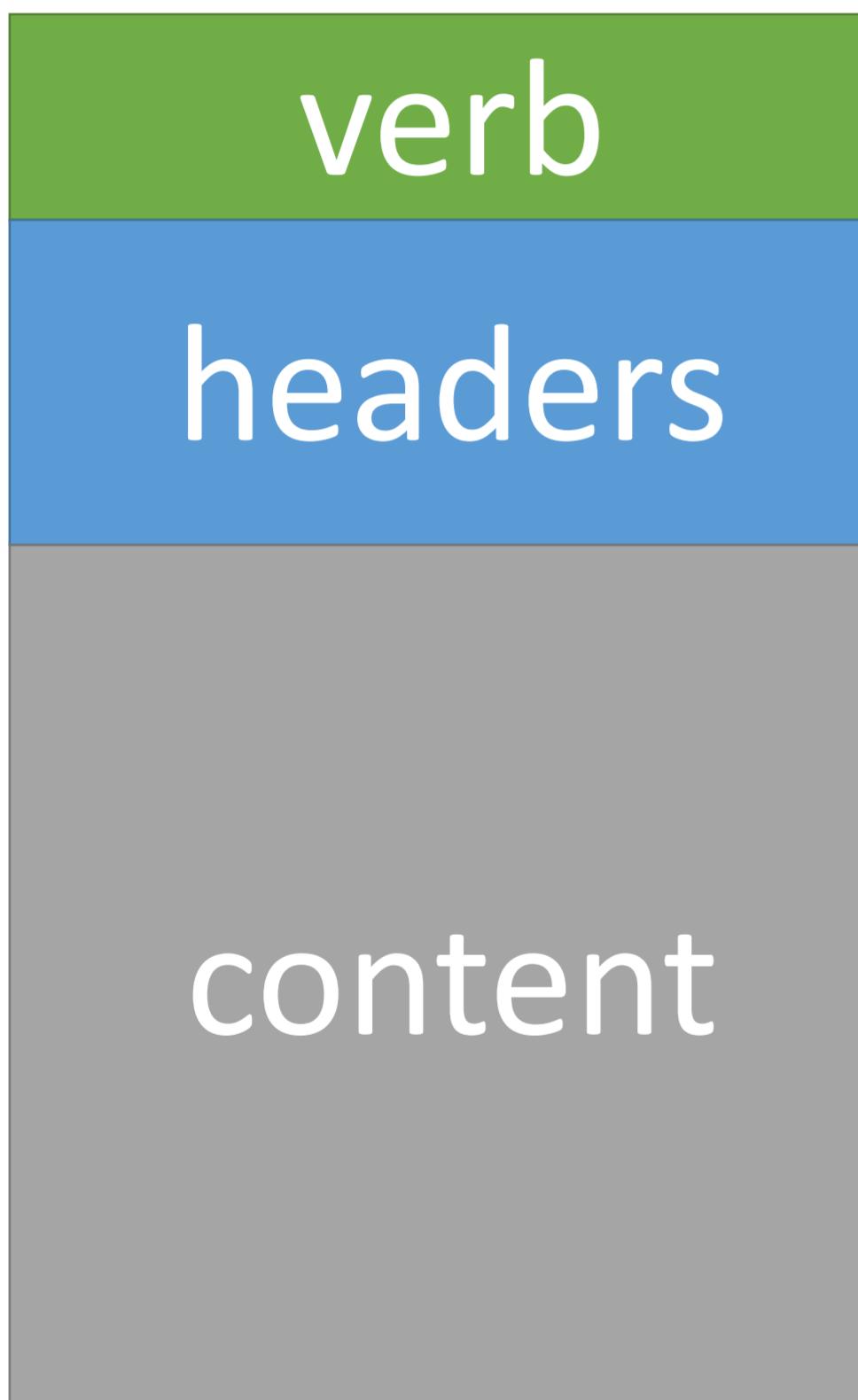
# İsteğin (Request) Yapısı



## İstek hakkında üst (meta) bilgiler:

- **Accept**
  - Kabul edilen tipler
- **Cookies**
  - İstek içindeki veriler
- ve daha fazlası...

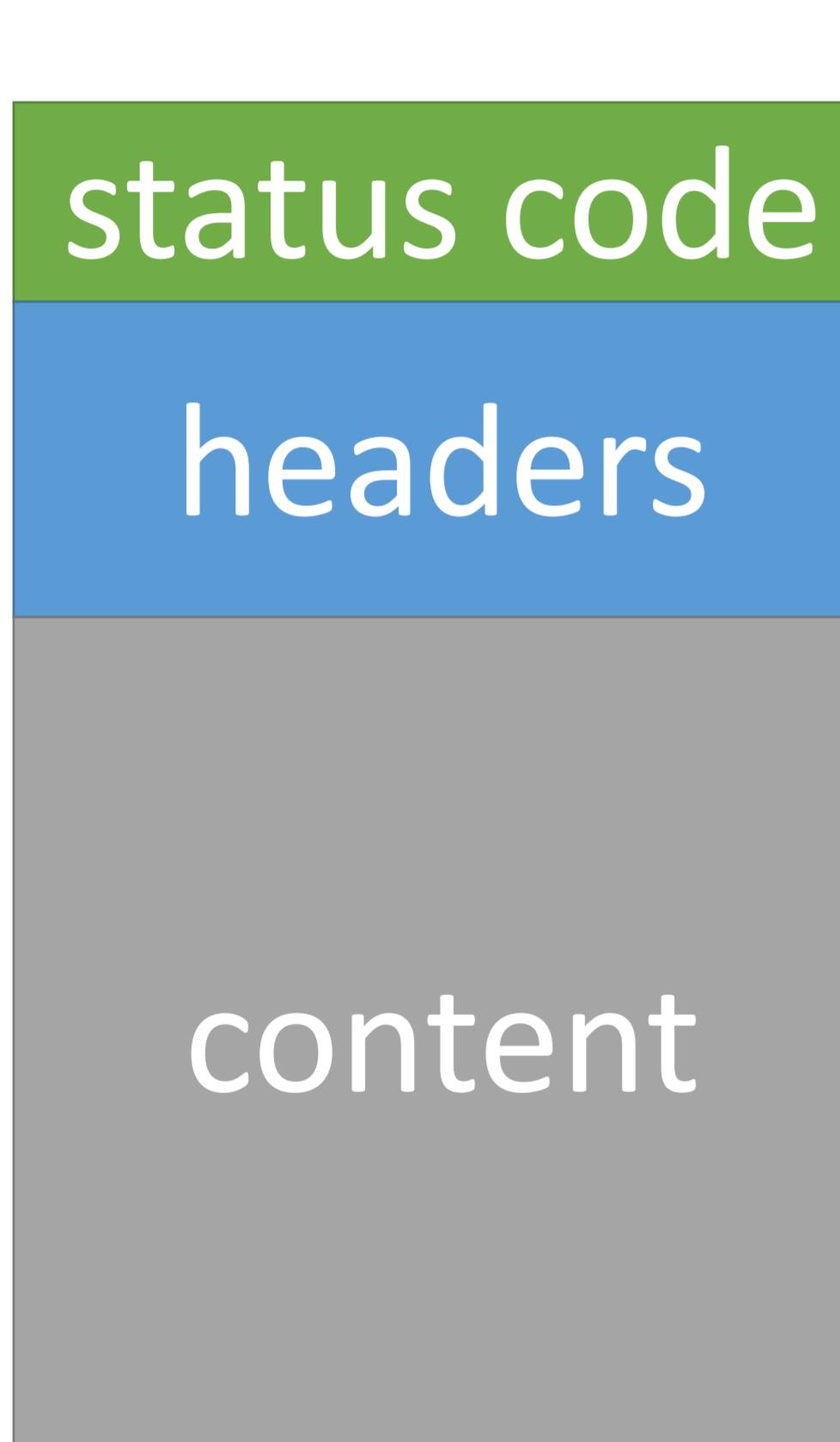
# İsteğin (Request) Yapısı



## İstek ile ilgili içerik:

- HMTL, CSS, JavaScript, XML, JSON
- Bazı eylemler ile geçerli olmayan içerik
- İsteği gerçekleştirmeye yardımcı olmak için bilgiler
- Binary ve blobs common (.jpg gibi)

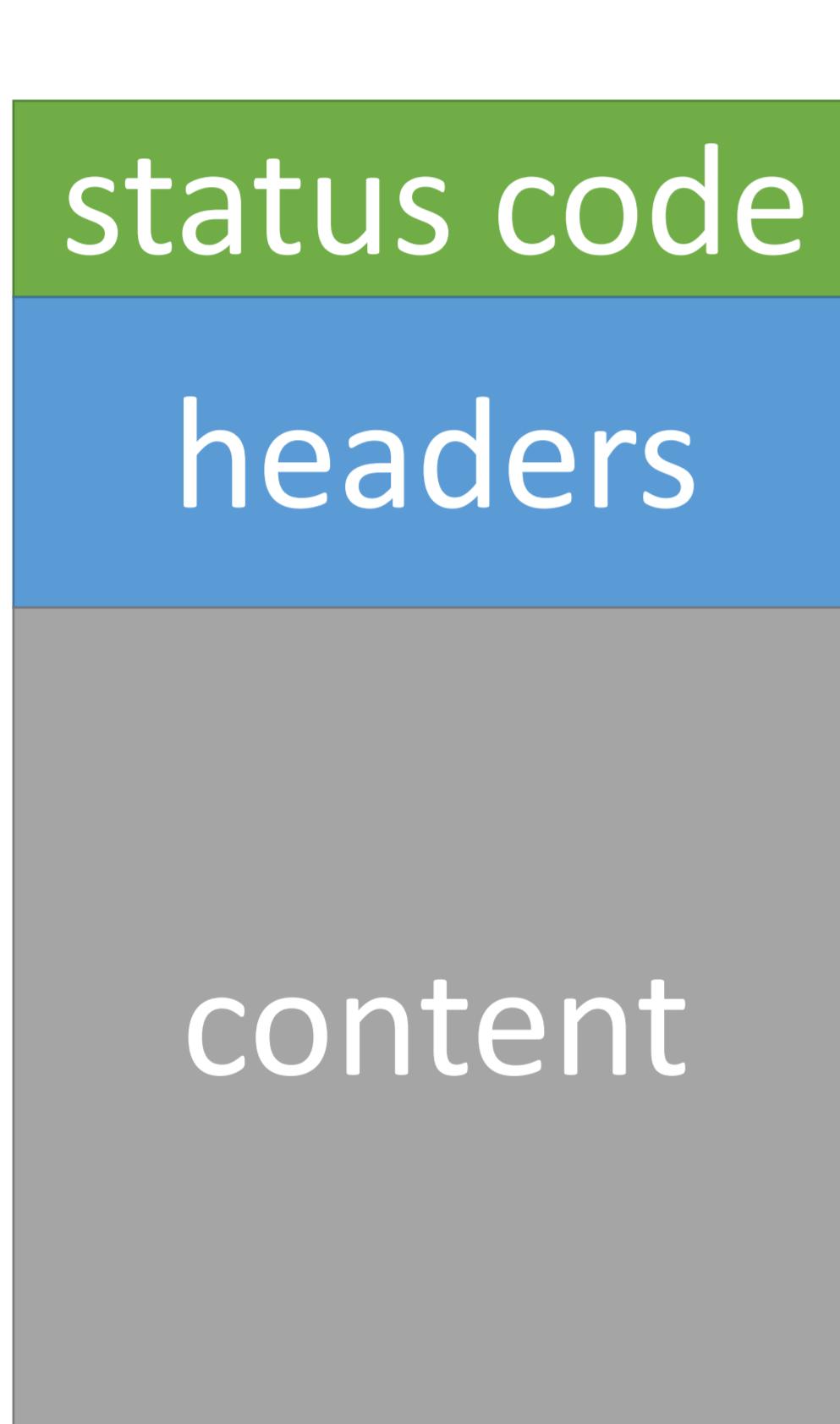
# Cevabin (Response) Yapısı



## Operasyon Durumları

- 100 – 199
  - Bilgi (Information)
- 200 – 299
  - Başarı (Success)
- 300 – 399
  - Yeniden yönlendirme (Redirection)
- 400 – 499
  - İstemci hataları (Client errors)
- 500 – 599
  - Sunucu hataları (Server errors)

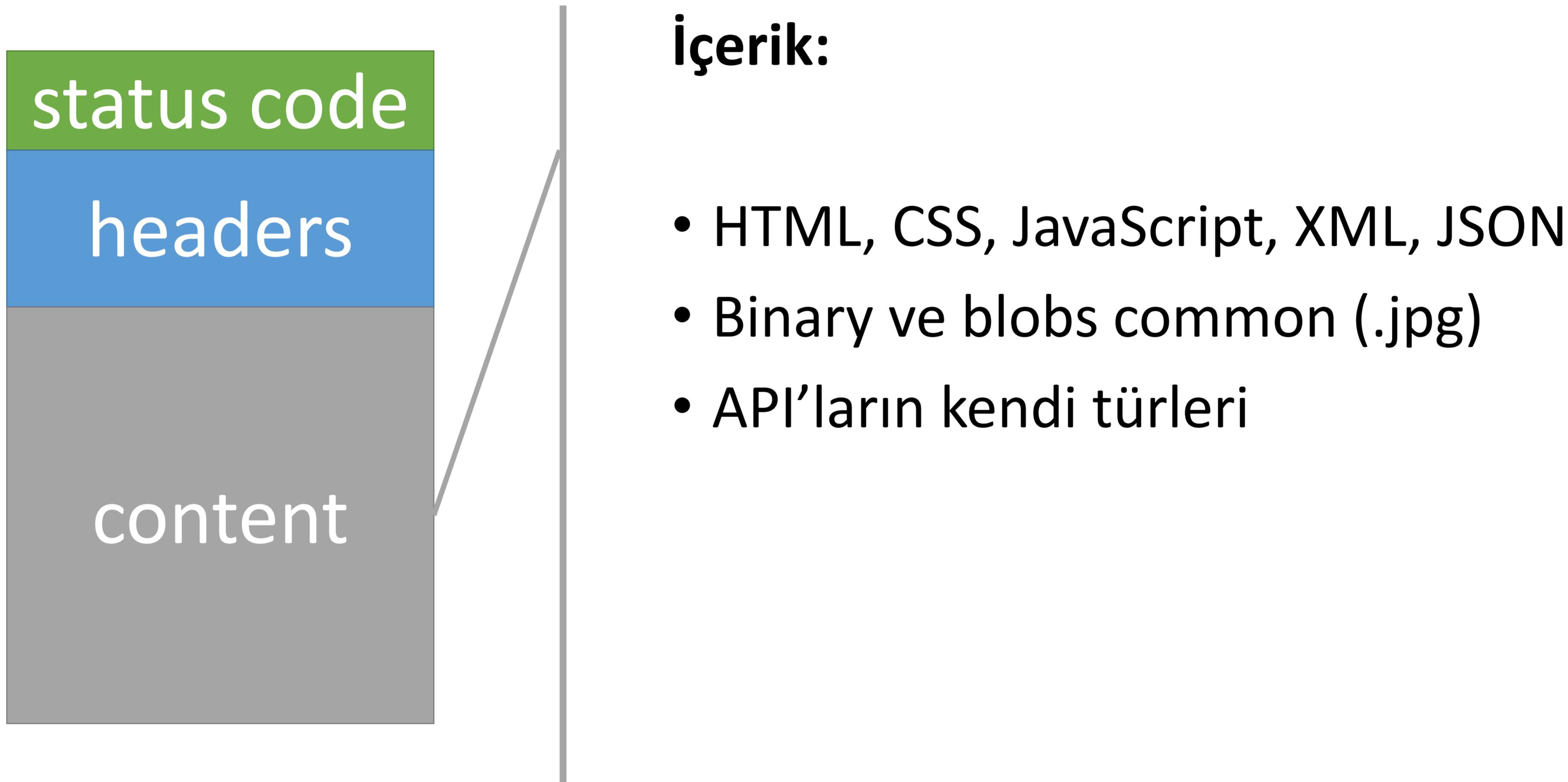
# Cevabin (Response) Yapısı



## Cevap hakkında üst (meta) bilgiler:

- **Content Type**
  - İçeriğin formatı
- **Content Length**
  - İçeriğin boyutu
- **Expires**
  - Ne zaman geçersiz sayılır

# Cevabin (Response) Yapısı



# Idempotent

*Sonucu değiştirmeden defalarca uygulanabilen işlemi ifade eder.*

# REST mimarisinden Idempotency

## Idempotency

- İşlem sonuçları GET, PUT, PATCH ve DELETE eylemlerinde aynıdır.
- POST idempotent değildir.

# RESTful API Tasarımı

## Sonuçların Tasarımı

### **Formata karar verin:**

- Kullanacağı formata karar verin. Genellikle JSON tercih edilir.
- Format için Query String kullanmayın. Headers bölümünü tercih edin.
- JSON
  - application/json
- XML
  - text/xml

# RESTful API Tasarımı

## En iyi pratikler

- Sonuçların kendini tanımlamasını sağlayınız.
- Programlı gezinmeye izin veriniz.
- Sayfalama, sıralama, filtreleme ve arama desteği sağlayınız.
- Önbellek desteği sununuz.
- Sorgu sınırı getiriniz.
- Veri şekillendirilmesini sağlayınız.
- Versiyonlama yapınız.
- Belgelendirmeye özen gösteriniz.

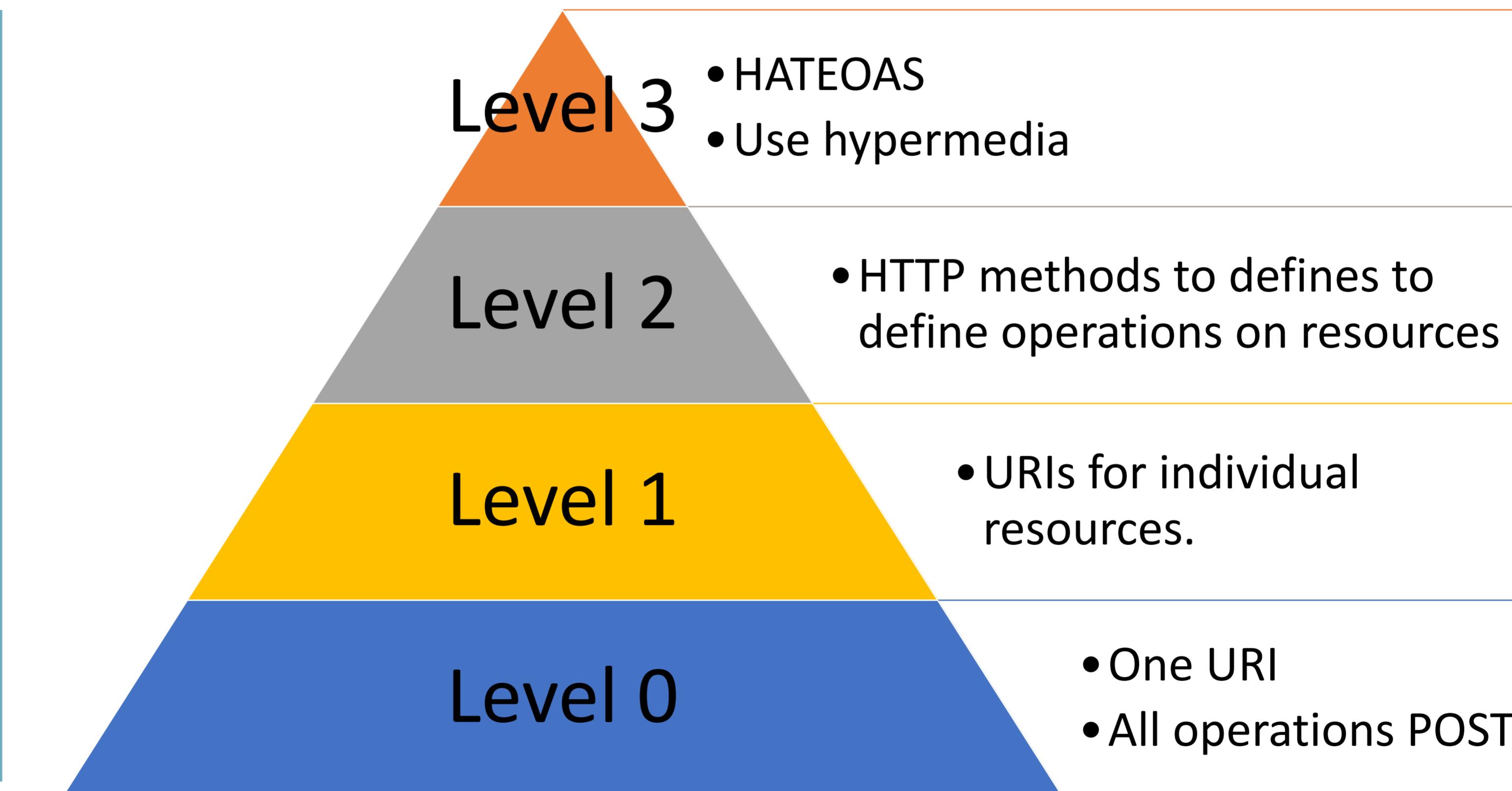
# Hypermedia

*Hipermedya yardımcı olabilir.*

*Ancak pragmatizm, çoğu projenin buna ihtiyaç duymadığı anlamına gelir.*

# RESTful API Tasarımı

Olgunlaşma  
Seviyesi

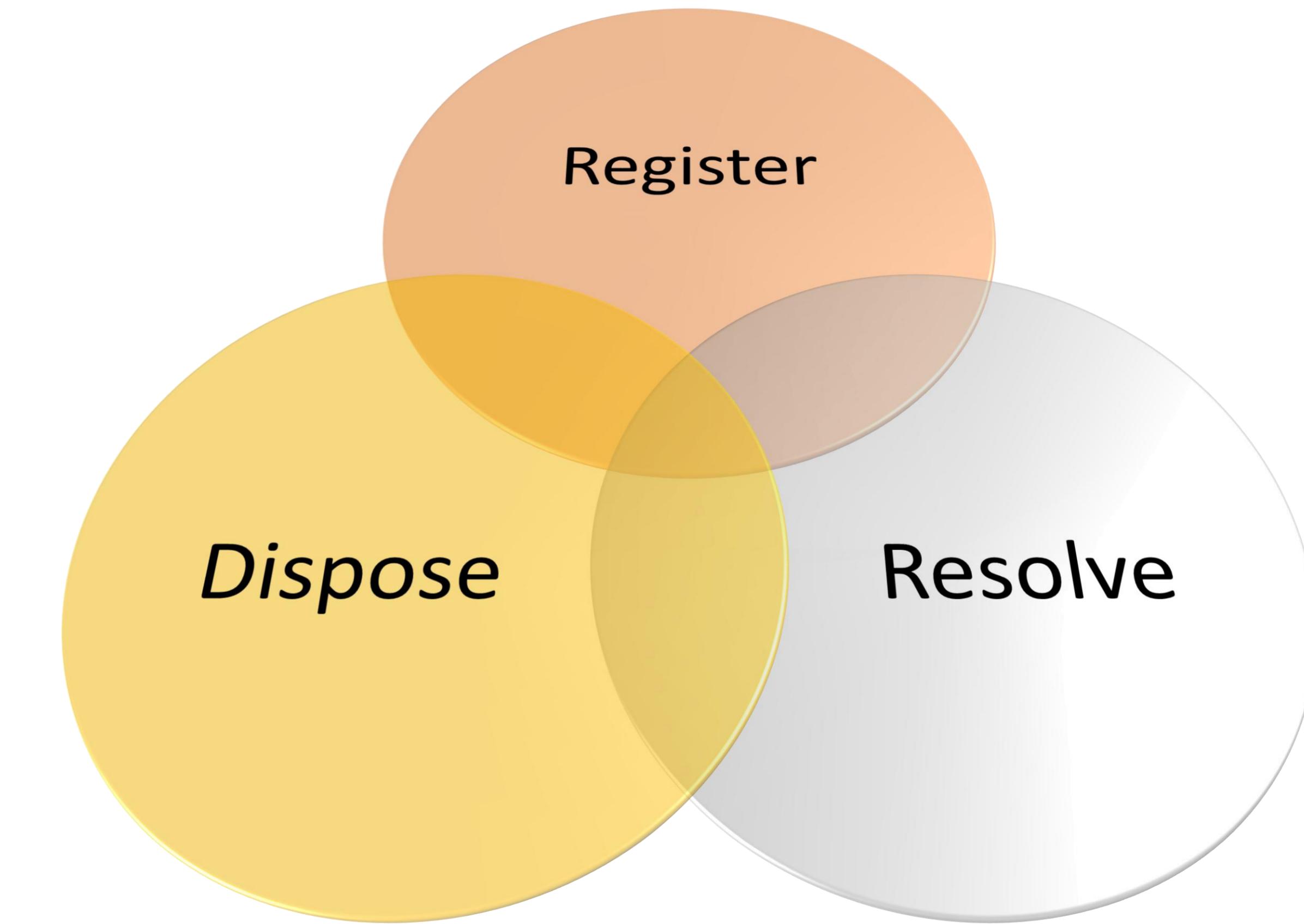


# RESTFul API Tasarımı

## İlk API Tasarımı

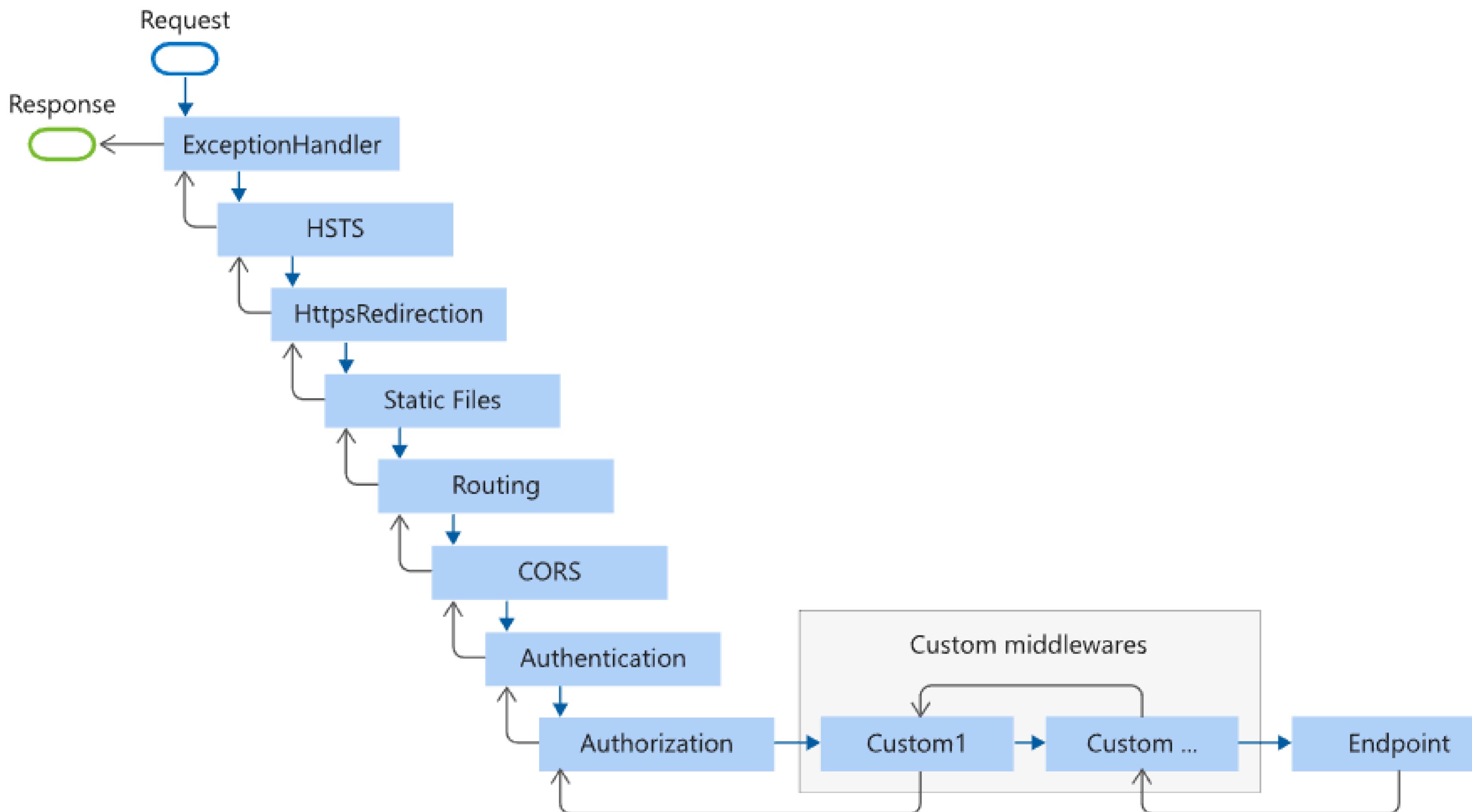
- API yayınladıktan sonra değiştirmeyin.
- İyi tasarlamış bir API olgunlaşabilir.

## Dependency Injection



<https://learn.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection?view=aspnetcore-7.0>

# Pipeline

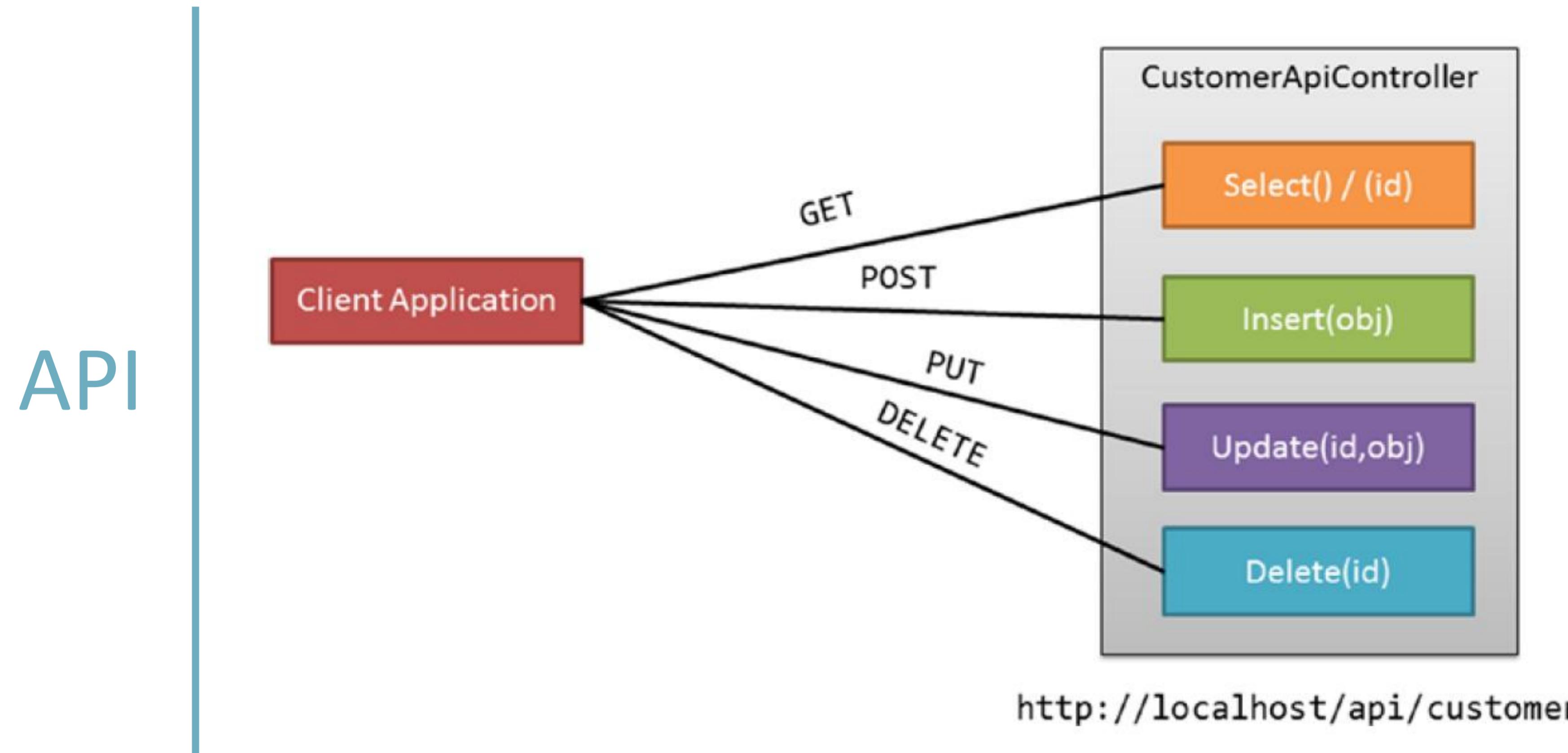


# Built-in Middlewares

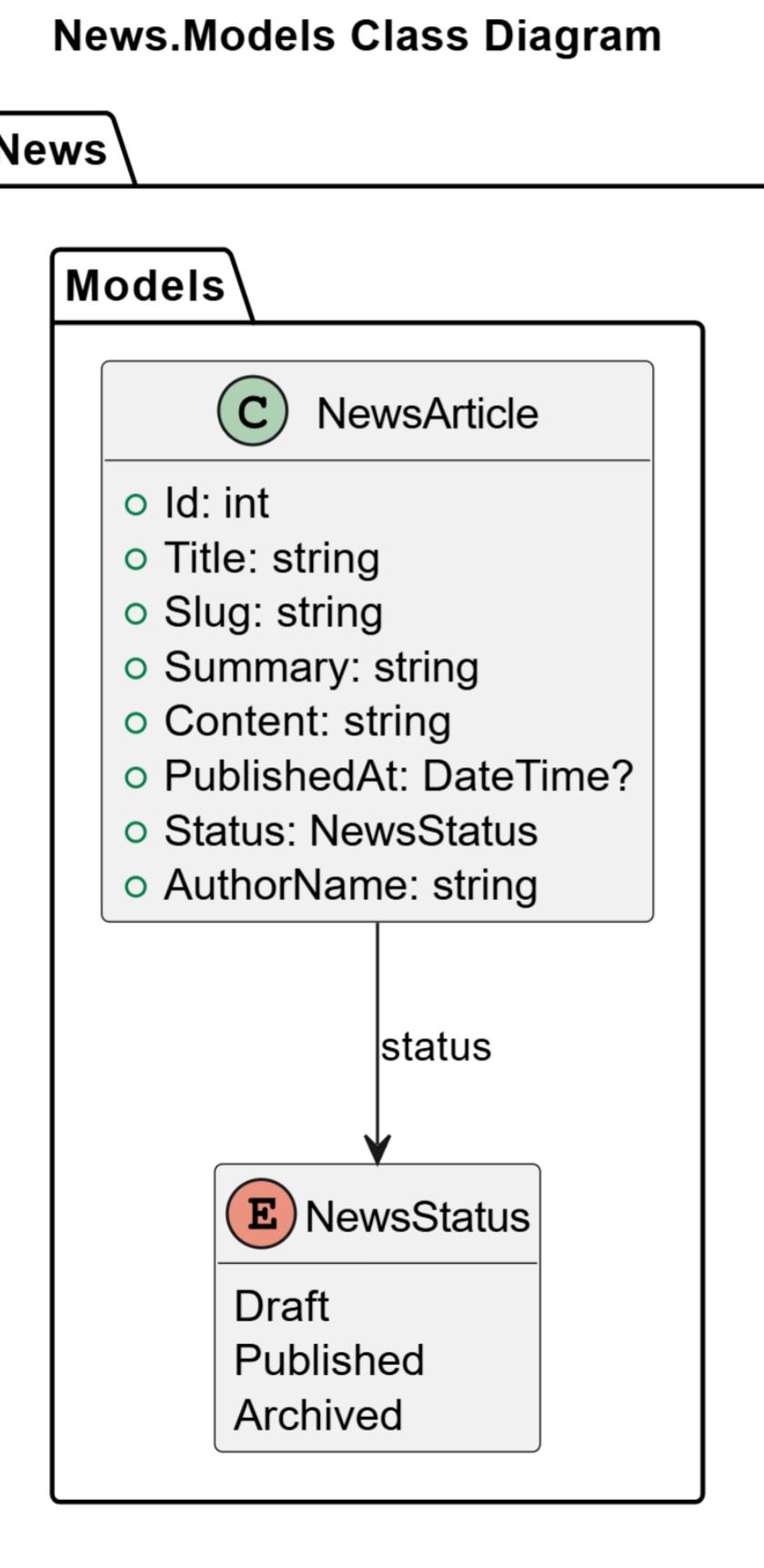
- Authentication
- Authorization
- Cookie Policy
- CORS
- DeveloperExceptionPage
- Diagnostic
- Forwarded Headers
- Health Check
- Header Propagation
- HTTP Logging
- HTTP Method Override
- HTTPS Redirection
- MVC
- Static Files

<https://learn.microsoft.com/en-us/aspnet/core/fundamentals/middleware>

# Web API

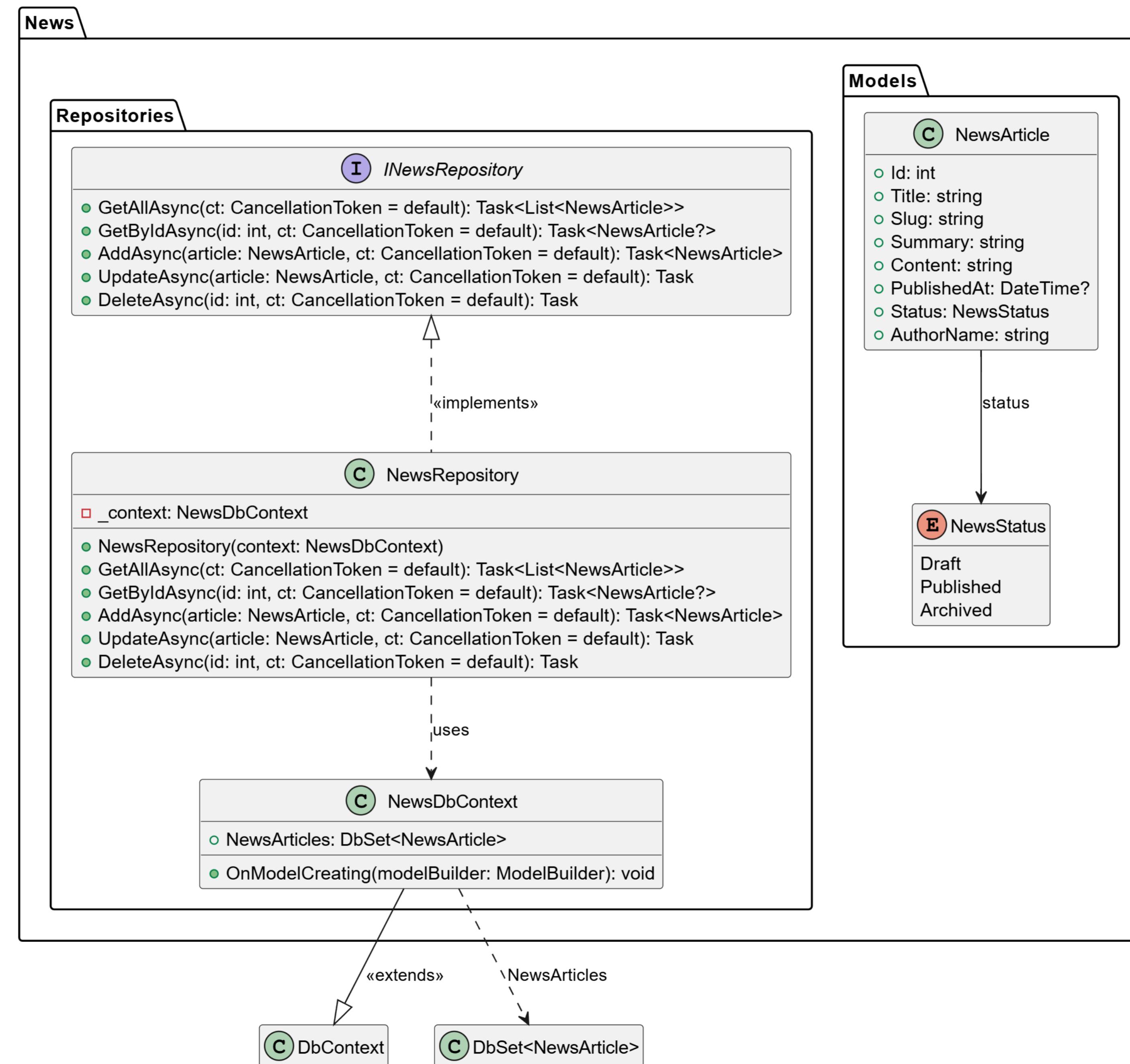


# Model



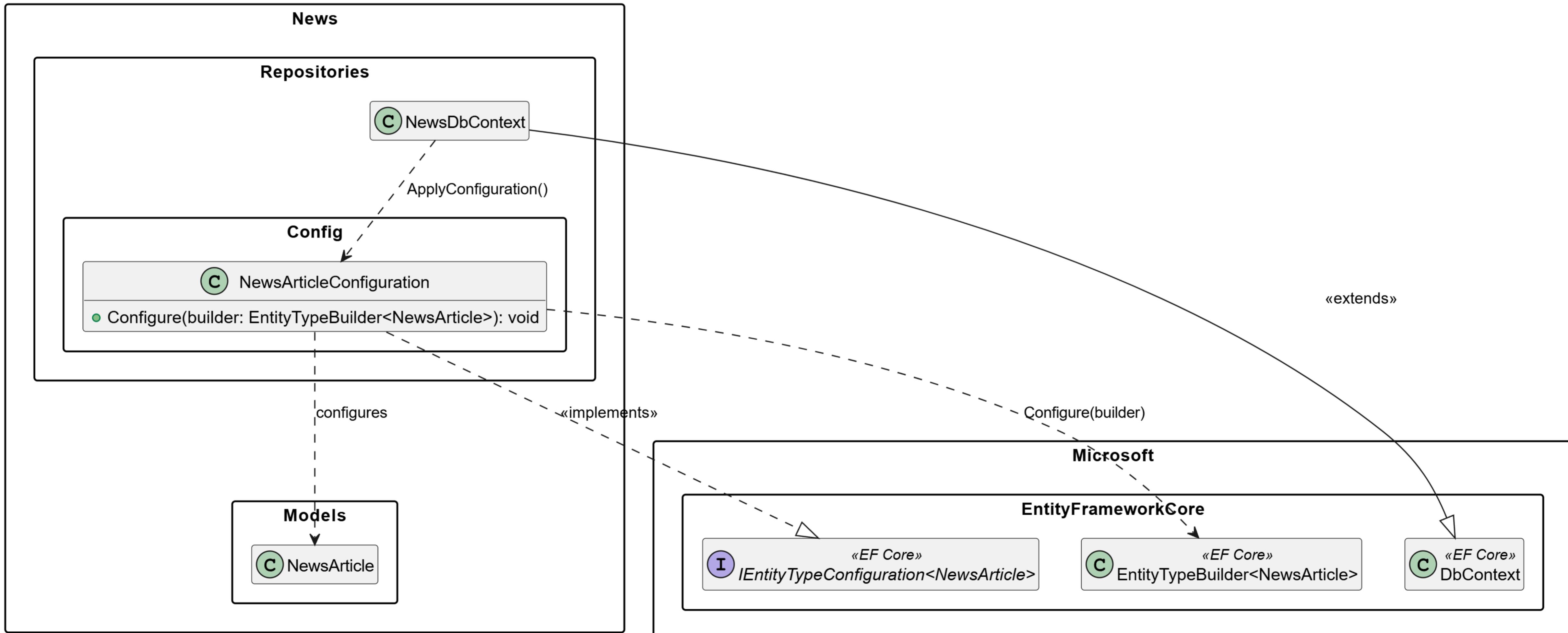
- **Id:** Benzersiz kimlik numarası.
- **Title:** Haber başlığı.
- **Slug:** URL dostu kısa ad.
- **Summary:** Haber özeti.
- **Content:** Tam içerik metni.
- **PublishedAt:** Yayımlanma tarihi.
- **Status:** Yayın durumu.
- **AuthorName:** Yazar adı.

## News Models and Repositories

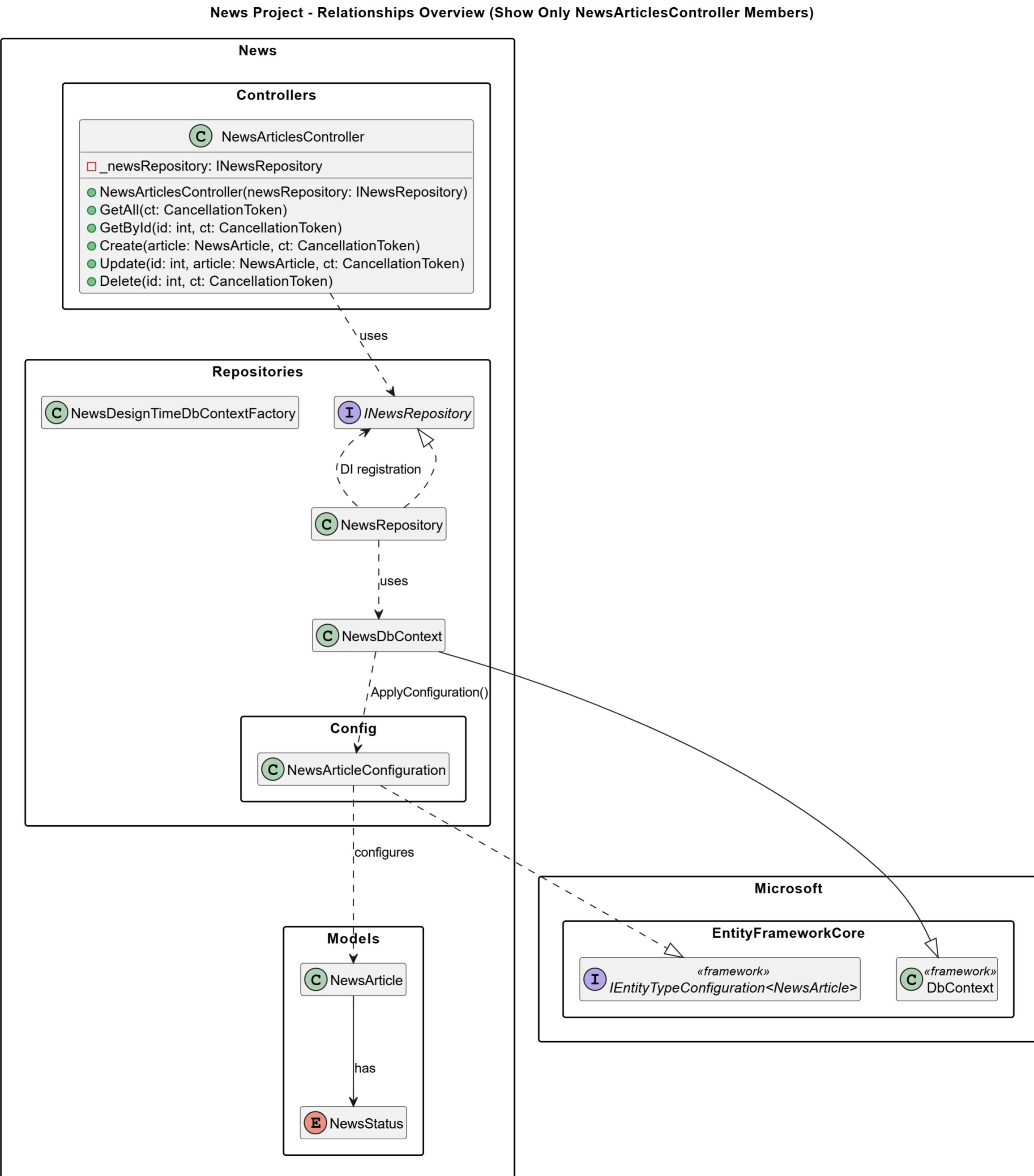


# Tip Yapılandırması

NewsArticle Configuration Diagram



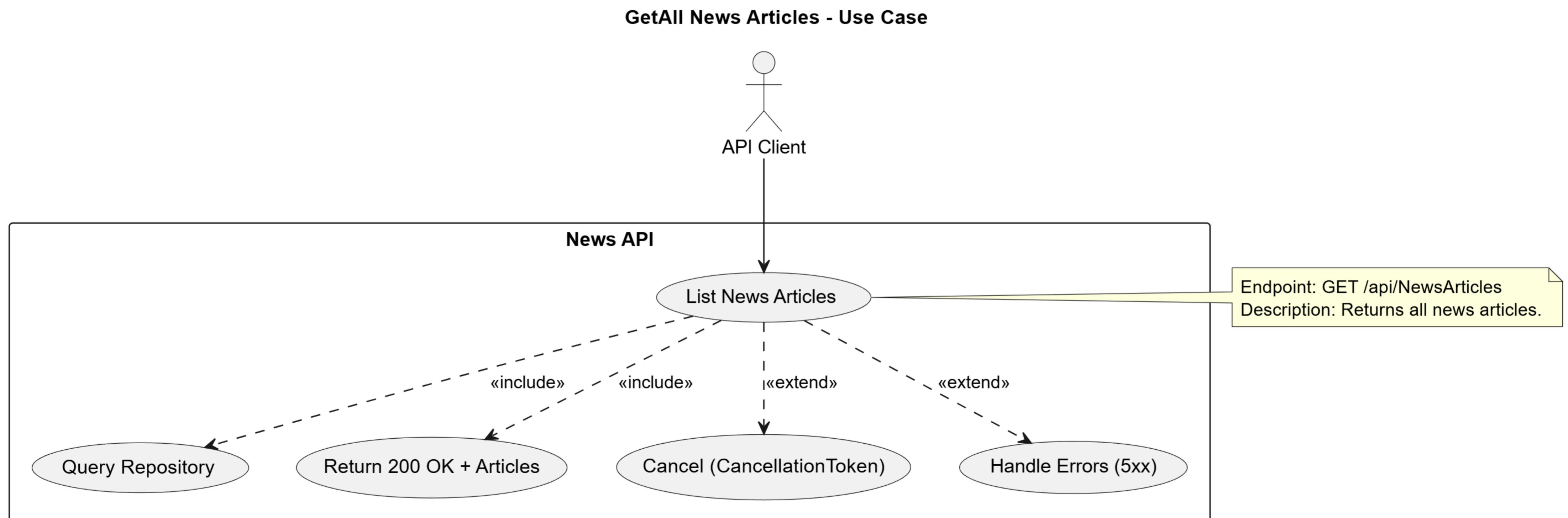
# Proje Sınıf Diyagramı



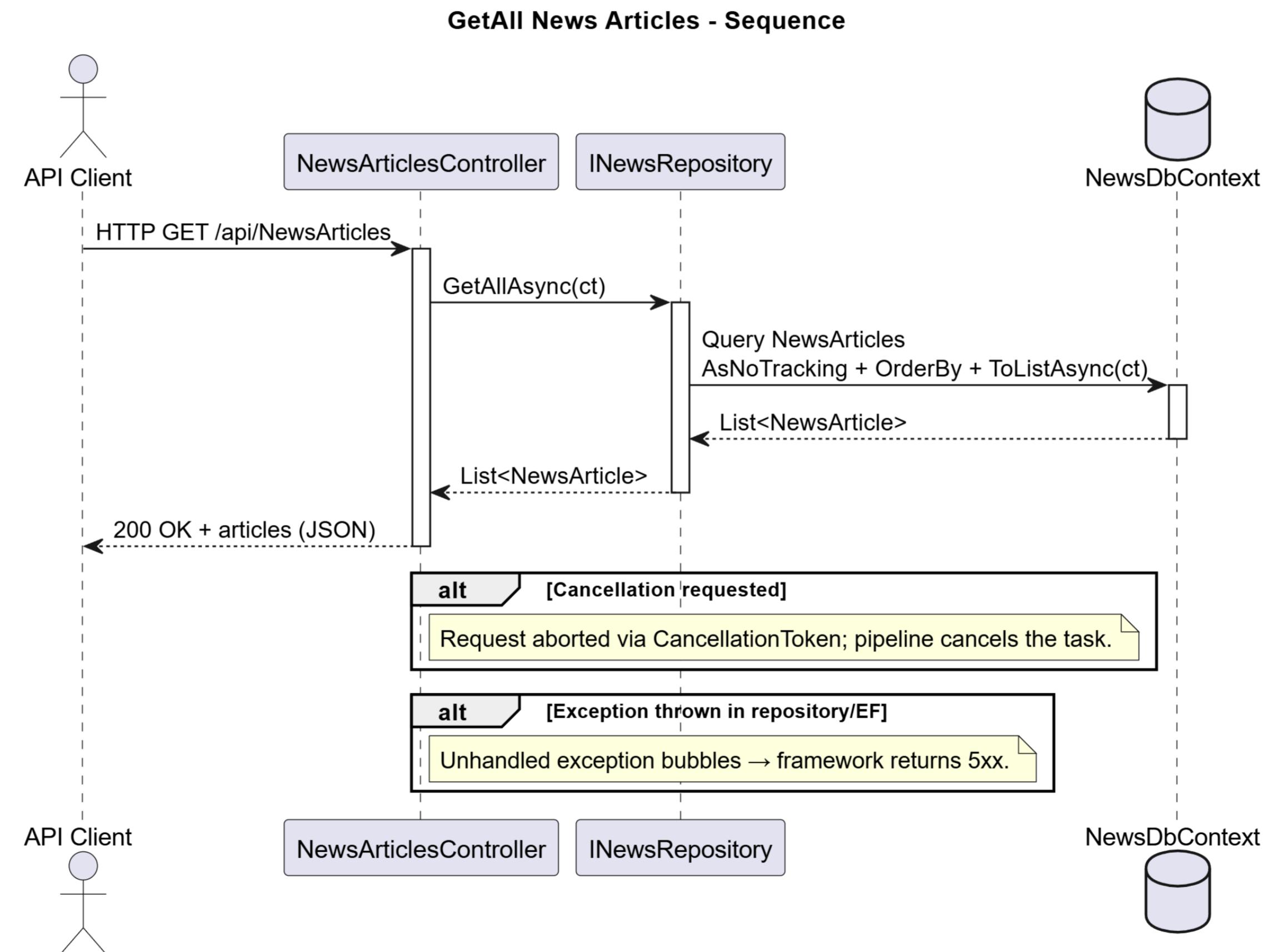
# Endpoints

HTTP Metodu	Endpoint	Açıklama	Yanıt
<b>GET</b>	api/NewsArticles	Tüm haberleri listeler.	200 OK
<b>GET</b>	api/NewsArticles/{id:int}	Belirli haberi (id) döner.	200 OK / 404 Not Found
<b>POST</b>	api/NewsArticles	Yeni haber oluşturur. (body: NewsArticle JSON)	201 Created + Location
<b>PUT</b>	api/NewsArticles/{id:int}	Haberi günceller. (body: NewsArticle JSON; route id = body id)	204 No Content
<b>DELETE</b>	api/NewsArticles/{id:int}	Haberi siler.	204 No Content

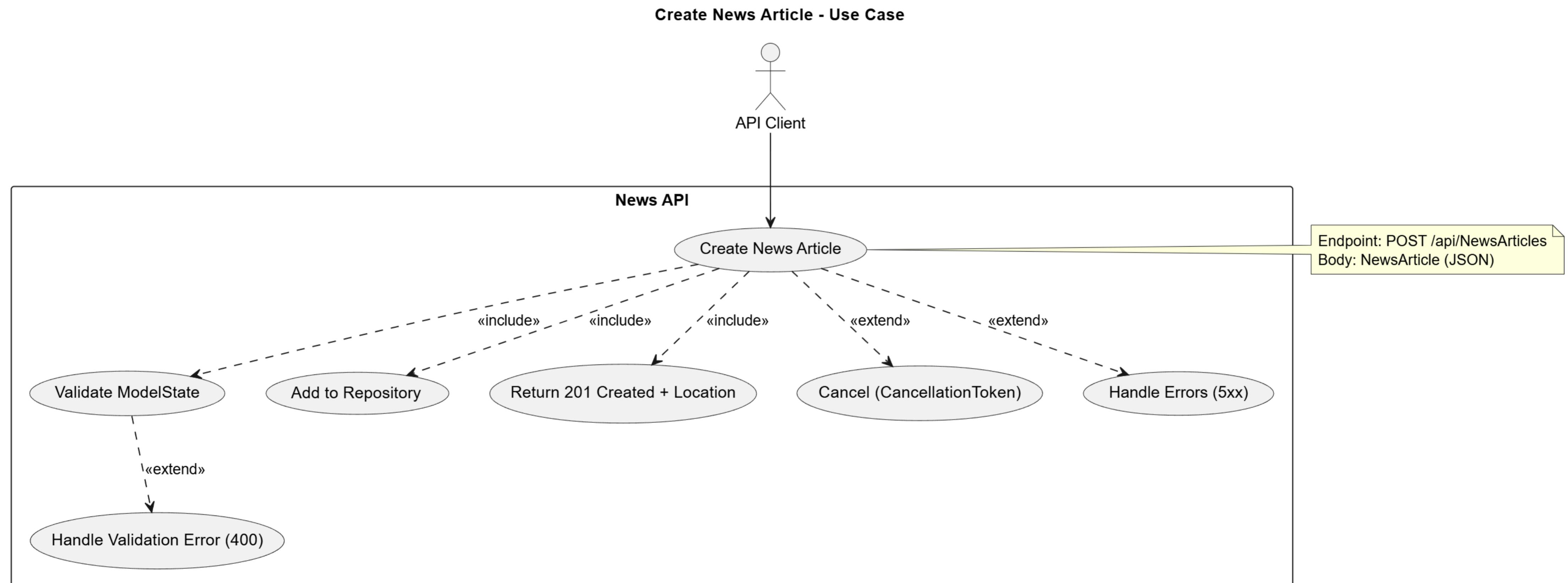
# Kullanım Senaryosu (Use Case) Diyagramı



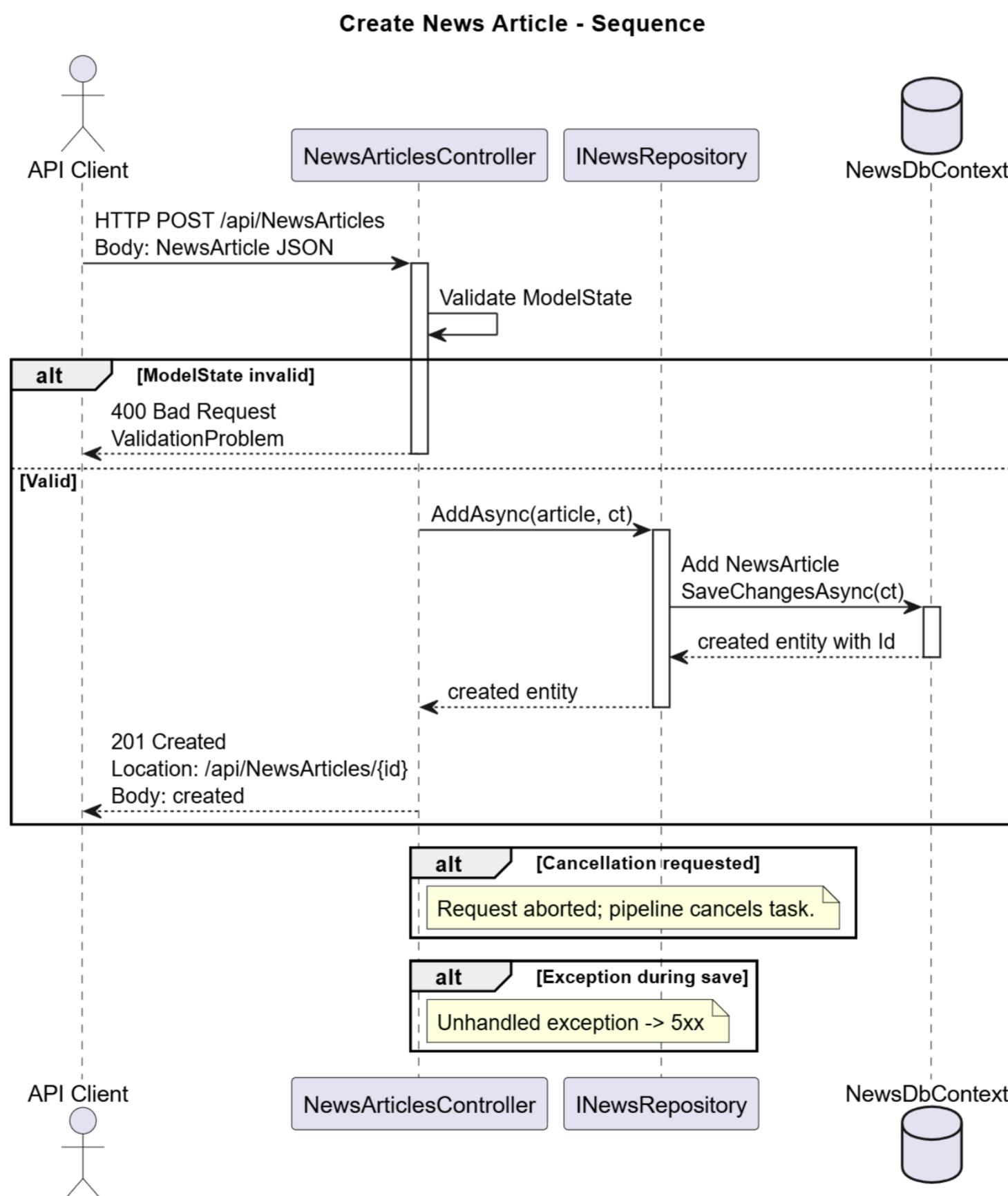
# Sıra (Sequence) Diyagramı



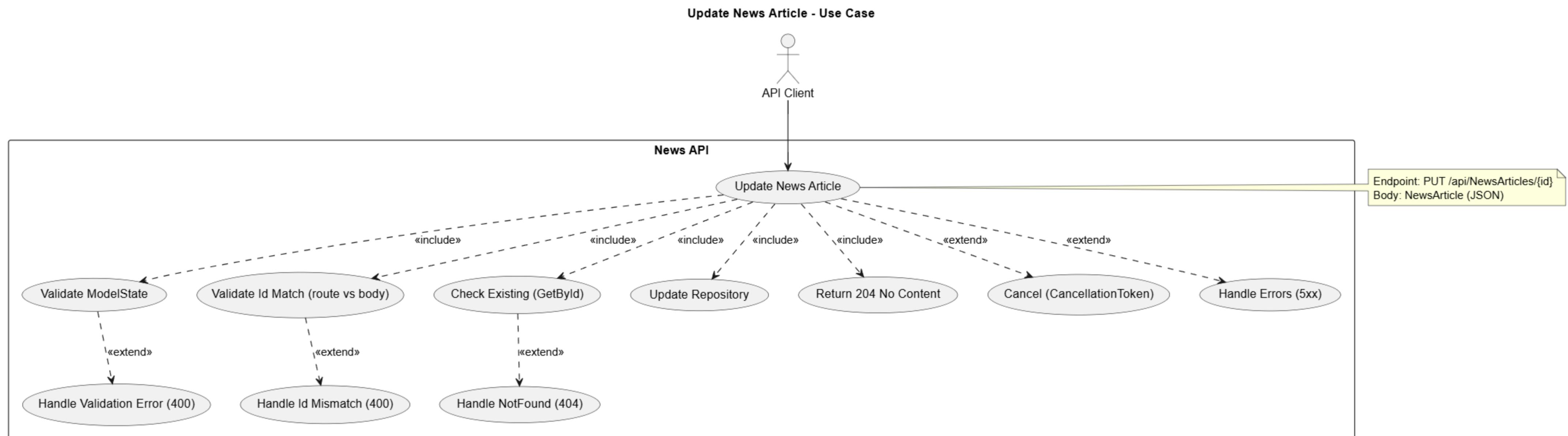
# Kullanım Senaryosu (Use Case) Diyagramı



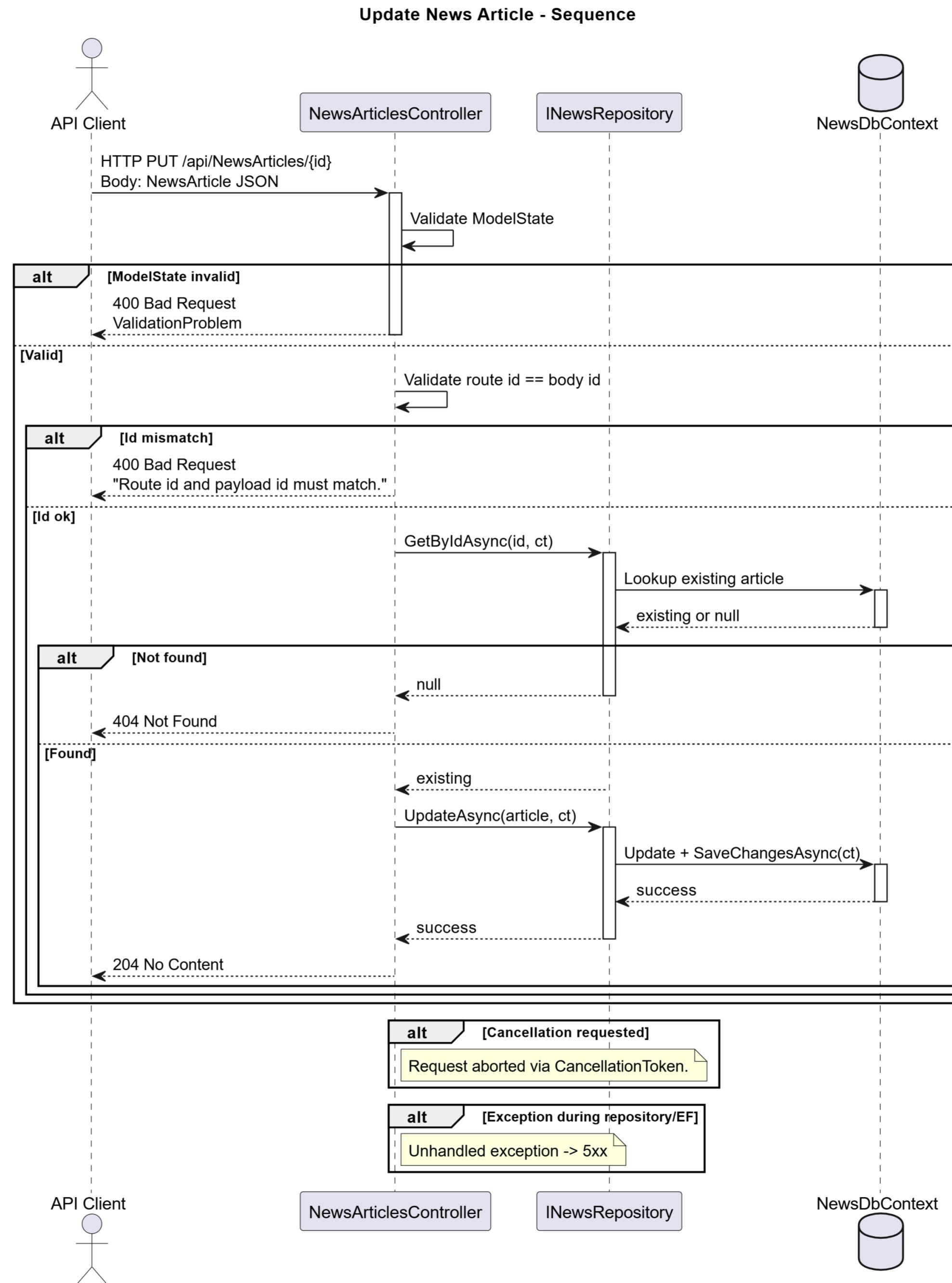
# Sıra (Sequence) Diyagramı



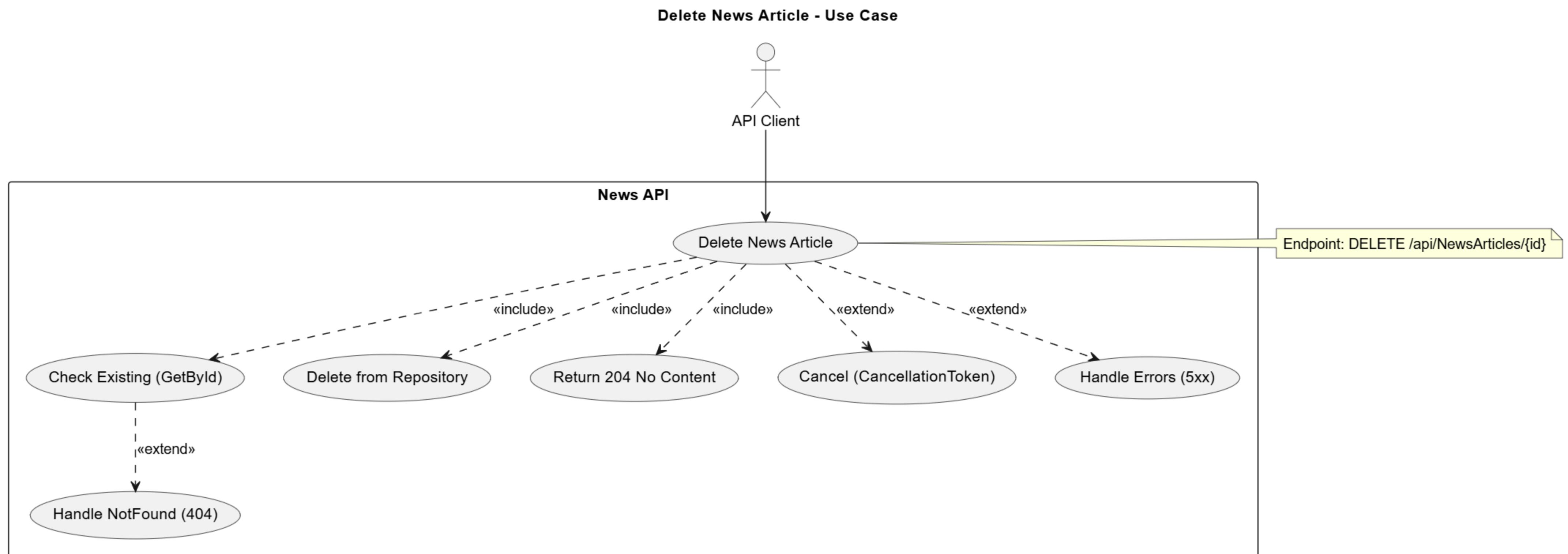
# Kullanım Senaryosu (Use Case) Diyagramı



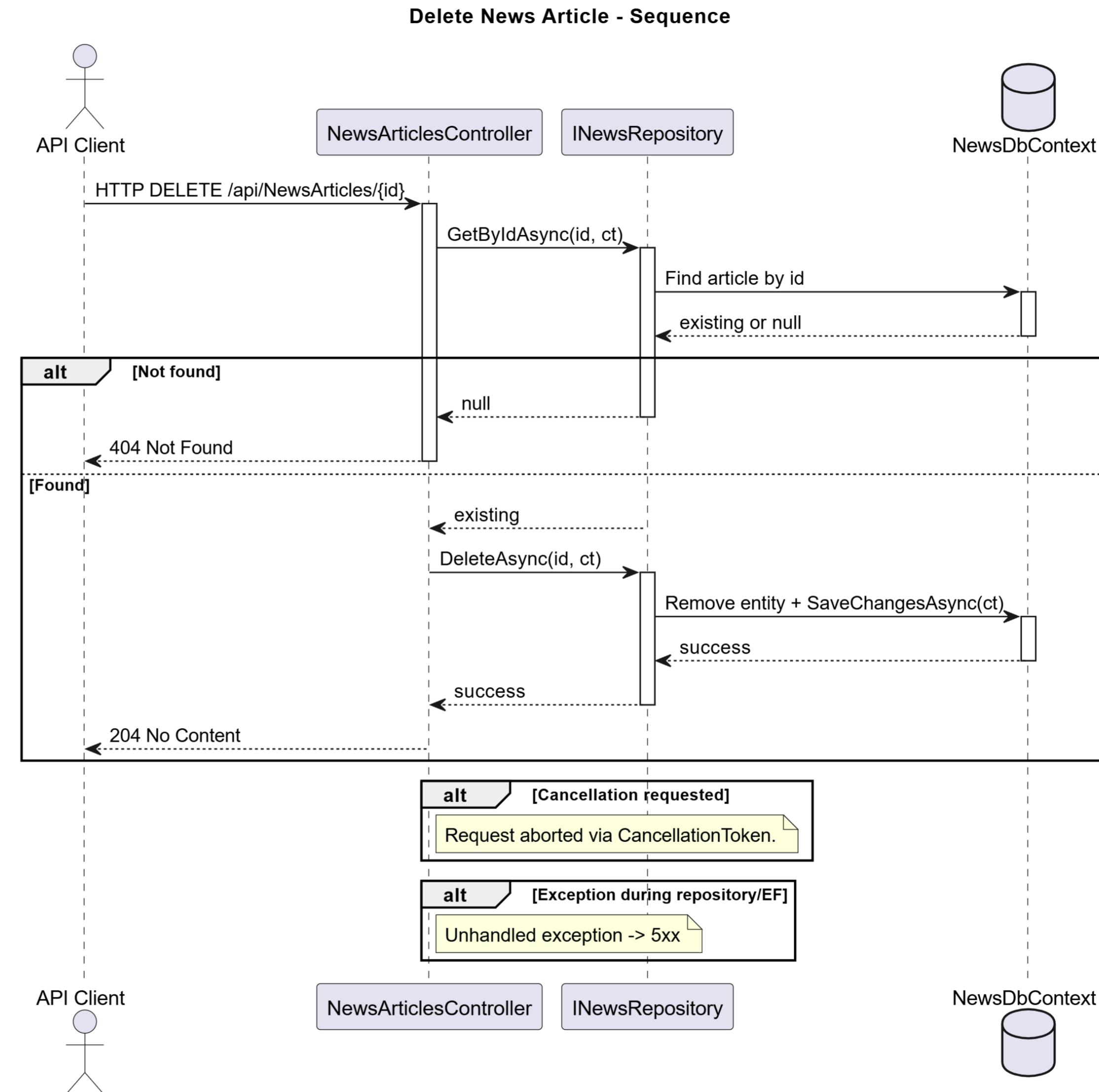
# Sira (Sequence) Diyagramı



# Kullanım Senaryosu (Use Case) Diyagramı



# Sira (Sequence) Diyagramı



# Teşekkürler

ZAFER CÖMERT  
Öğretim Üyesi