

# HTTP Rehberi

## 1. HTTP Protokolüne Giriş

HTTP (Hypertext Transfer Protocol), istemciler (web tarayıcıları, mobil uygulamalar, diğer sunucular) ile sunucular arasında veri alışverişini sağlayan bir uygulama katmanı protokolüdür. Web API'leri geliştirirken, istemci ve sunucu arasındaki iletişimin temelini HTTP oluşturur.

### Temel Özellikleri:

- **İstek-Yanıt Mekanizması:** İletişim, istemcinin bir HTTP isteği göndermesiyle başlar ve sunucunun bir HTTP yanıtı dönmesiyle tamamlanır.
- **Durumsuz (Stateless):** Her HTTP isteği, önceki isteklerden bağımsız olarak işlenir. Sunucu, istekler arasında herhangi bir durumu hatırlamaz. Bu, ölçeklenebilirliği artırırken, oturum yönetimi gibi durum bilgileri için çerezler, token'lar veya veritabanı gibi harici mekanizmaların kullanılmasını gerektirir.
- **Metin Tabanlı:** Protokol mesajları genellikle insan tarafından okunabilir metin formatında iletilir, bu da hata ayıklama ve izleme süreçlerini kolaylaştırır.

## 2. HTTP Mesaj Yapısı: İstek ve Yanıt

HTTP iletişimi, bir istemci tarafından gönderilen **HTTP İstek Mesajı** ve sunucu tarafından geri dönülen **HTTP Yanıt Mesajı** olmak üzere iki ana mesaj türünden oluşur.

### 2.1. HTTP İstek Mesajı (HTTP Request Message)

Bir HTTP isteği temel olarak şu bölümlerden oluşur:

- **İstek Satırı (Request Line):**
  - **HTTP Metodu:** İstemcinin kaynak üzerinde gerçekleştirmek istediği eylemi tanımlar (örn: GET, POST, PUT, DELETE).
  - **İstek URL'si (Request-URI):** Hedef kaynağın sunucu üzerindeki yolu.
  - **HTTP Protokol Sürümü:** Kullanılan HTTP sürümü (örn: HTTP/1.1, HTTP/2.0).
  - **Örnek:** GET /api/urunler/1 HTTP/1.1
- **İstek Başlıkları (Request Headers):** İstemci, istek veya kaynak hakkında ek meta veriler sağlar. Anahtar-değer çiftlerinden oluşur.
  - **Sık Kullanılan Başlıklar:**
    - **Host:** Hedef sunucunun ana bilgisayar adı ve isteğe bağlı olarak port numarası.
    - **User-Agent:** İsteği yapan istemci yazılımının bilgisi (örn: tarayıcı tipi).
    - **Accept:** İstemcinin kabul edebileceği medya türleri (örn: application/json, text/html).
    - **Content-Type:** İstek gövdesinin medya türü (örn: application/json bir JSON gövdesi gönderildiğinde).
    - **Content-Length:** İstek gövdesinin bayt cinsinden uzunluğu.

Bu rehber Acunmedya Akademi 13.Dönem Backend Sınıfı için [Engin Niyazi Ergül](#) tarafından hazırlanmıştır.

- Authorization: Kimlik doğrulama veya yetkilendirme bilgileri (örn: Bearer <token>).
- Örnek:  
Host: api.example.com
- User-Agent: PostmanRuntime/7.26.8
- Accept: application/json
- Authorization: Bearer eyJhbGci...
- 
- **Boş Satır:** Başlıklar bölümünü isteğe bağlı istek gövdesinden ayırır.
- **İstek Gövdesi (Request Body - Opsiyonel):** Özellikle POST, PUT, ve PATCH metodlarında sunucuya gönderilecek veriyi (payload) içerir. Veri formatı genellikle Content-Type başlığı ile belirtilir (örn: JSON, XML, form verisi).
  - Örnek (POST isteği için JSON gövde):

```
{  
  "urunAdi": "Akıllı Telefon",  
  "fiyat": 15000,  
  "stokAdedi": 50  
}
```
  -

## 2.2. HTTP Yanıt Mesajı (HTTP Response Message)

Bir HTTP yanıtı da benzer şekilde şu bölümlerden oluşur:

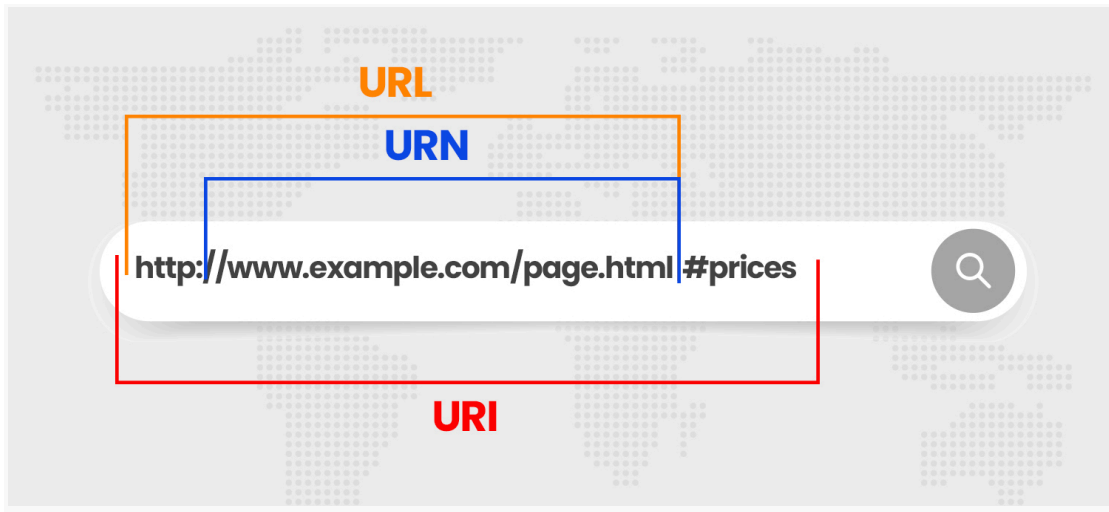
- **Durum Satırı (Status Line):**
  - **HTTP Protokol Sürümü:** Kullanılan HTTP sürümü.
  - **Durum Kodu (Status Code):** İsteğin sonucunu (başarılı, istemci hatası, sunucu hatası vb.) belirten 3 haneli sayısal kod.
  - **Durum Metni (Reason Phrase):** Durum kodunun kısa, insan tarafından okunabilir metinsel açıklaması.
  - Örnek: HTTP/1.1 200 OK
- **Yanıt Başlıkları (Response Headers):** Sunucu, yanıt veya kaynak hakkında ek meta veriler sağlar.
  - **Sık Kullanılan Başlıklar:**
    - Content-Type: Yanıt gövdesinin medya türü.
    - Content-Length: Yanıt gövdesinin bayt cinsinden uzunluğu.
    - Server: Yanıtı üreten sunucu yazılımı bilgisi.
    - Date: Yanıtın oluşturulduğu tarih ve saat.
    - Cache-Control: Önbellekleme davranışını kontrol eder.
  - Örnek:  
Content-Type: application/json; charset=utf-8
  - Content-Length: 125
  - Date: Tue, 05 Nov 2023 10:30:00 GMT
  - Server: Kestrel
  -
- **Boş Satır:** Başlıklar bölümünü isteğe bağlı yanıt gövdesinden ayırır.

Bu rehber Acunmedya Akademi 13.Dönem Backend Sınıfı için [Engin Niyazi Ergül](#) tarafından hazırlanmıştır.

- **Yanıt Gövdesi (Response Body - Opsiyonel):** İstemci tarafından istenen veriyi (örn: JSON, XML, HTML belge) veya hata detaylarını içerir.
  - Örnek (GET isteği için JSON gövde):

```
{  
  "id": 1,  
  "urunAdi": "Akıllı Telefon",  
  "fiyat": 15000,  
  "stokAdedi": 50  
}
```
  - 
  - 
  - 
  - 
  - 
  -

\*\*



## ASP.NET Core Web API HTTP Rehberi

### HTTP İstek Mesajı (HTTP Request Message)

#### İstek Satırı

```
GET /api/urunler/1 HTTP/1.1  
Method: GET  
URI: /urunler/1  
Version: HTTP/1.1
```

#### Başlıklar (Headers)

```
Host: api.example.com  
User-Agent: Postman...  
Accept: application.json  
Authorization: Bearer <token>  
Content-Type: applrtion
```

#### İstek Gövdei (Body)

```
{  
  "urunAdi":  
    "Akıllı Telfon,  
    fiyat: 15000,  
    stokAdedi": 50)  
}
```

### HTTP Yanıt Mesajı (Response Message)

#### Durum Satırı

```
HTP/1.1 200 OK  
Version: HTP/1.1  
Status Code: 200  
Reason: OK
```

#### Başlıklar Headers)

```
Content-Type application  
Content-Length: json  
Content-Length: 125  
Date: Tue, 05 Nov 2023...  
Server: Kestrel  
Cache-Control: public, max-age,600
```

#### Yanıt Gövdei (Body)

```
{  
  "id: 1  
  "urunlu Telfon,  
  fiyat: 15000,  
  stokAdedi" 50)  
}
```



### 3. En Çok Kullanılan HTTP Metotları (Verbs)

HTTP metotları, istemcinin belirli bir kaynak üzerinde gerçekleştirmek istediği eylemi belirtir. RESTful API tasarımının temelini oluştururlar.

- **GET:** Bir kaynaktan veri **almak** için kullanılır. Yanıt gövdesinde istenen kaynak veya kaynaklar döndürülür. GET istekleri idempotent (aynı istek defalarca yapıldığında sunucu tarafında bir değişikliğe yol açmaz) ve güvenlidir (sadece veri okur, değiştirmez).
  - *Kullanım:* /api/urunler (Tüm ürünleri listele), /api/urunler/5 (ID'si 5 olan ürünü getir).
  - *Örnek Senaryo:* Bir blog gönderisini veya kullanıcı profilini görüntülemek istediğinizde.
- **POST:** Sunucuda **yeni bir kaynak oluşturmak** için kullanılır. İstek gövdesinde yeni kaynağın verileri bulunur.
  - *Kullanım:* /api/urunler (Yeni bir ürün oluştur).
  - *Örnek Senaryo:* Yeni bir kullanıcı kaydı oluşturmak, bir form göndermek veya bir makaleyi yayınlamak.
- **PUT:** Mevcut bir kaynağı **tamamen güncellemek** veya belirtilen URI'de bir kaynak yoksa **yeni bir kaynak oluşturmak** için kullanılır. İstek gövdesinde kaynağın tüm yeni verileri bulunur. PUT idempotenttir.
  - *Kullanım:* /api/urunler/5 (ID'si 5 olan ürünü tamamen güncelle).
  - *Örnek Senaryo:* Bir kullanıcının tüm profil bilgilerini güncellemek veya bir ürünün tüm detaylarını değiştirmek.
- **PATCH:** Mevcut bir kaynağın **kısmi olarak güncellemek** için kullanılır. İstek gövdesinde sadece güncellenecek alanlar bulunur. PATCH idempotent değildir.
  - *Kullanım:* /api/urunler/5 (ID'si 5 olan ürünün sadece fiyatını güncelle).
  - *Örnek Senaryo:* Bir kullanıcının sadece e-posta adresini veya bir ürünün sadece stok miktarını değiştirmek.
- **DELETE:** Belirtilen bir kaynağı sunucudan **silmek** için kullanılır. DELETE idempotenttir.
  - *Kullanım:* /api/urunler/5 (ID'si 5 olan ürünü sil).
  - *Örnek Senaryo:* Bir yorumu silmek, bir kullanıcı hesabını kapatmak.

### 4. En Sık Kullanılan HTTP Durum Kodları

HTTP durum kodları, bir sunucunun belirli bir isteğe nasıl yanıt verdiğini gösteren 3 haneli sayılardır. Her kod belirli bir anlamı temsil eder ve 5 ana kategoriye ayrılır:

#### 4.1. 1xx - Bilgilendirme Yanıtları (Informational Responses)

İsteğin alındığını ve işlemin devam ettiğini belirtir. Genellikle geçici yanıtlardır ve nadiren doğrudan API geliştirmede kullanılır.

#### 4.2. 2xx - Başarılı Yanıtlar (Successful Responses)

İsteğin başarıyla alındığını, anlaşıldığını ve işlendiğini belirtir.

Bu rehber Acunmedya Akademi 13.Dönem Backend Sınıfı için [Engin Niyazi Ergül](#) tarafından hazırlanmıştır.

- **200 OK:** İstek başarıyla tamamlandı. GET, PUT, PATCH, DELETE gibi çoğu başarılı istek için varsayılan durum kodudur. Yanıt gövdesi, istenen veriyi veya işlemin sonucunu içerir.
  - *Kullanım Durumu:* Bir kaynaktan veri başarıyla çekildiğinde, bir kaynak başarıyla güncellendiğinde veya silindiğinde.
  - *Örnek:* "Veritabanından ürün listesi başarıyla getirildi."
- **201 Created:** İstek sonucunda sunucuda **yeni bir kaynak başarıyla oluşturuldu**. Yanıt gövdesinde genellikle oluşturulan kaynağın kendisi ve Location başlığında yeni kaynağın URI'si bulunur.
  - *Kullanım Durumu:* POST metodu ile yeni bir kaynak oluşturulduğunda.
  - *Örnek:* "Yeni kullanıcı hesabı başarıyla oluşturuldu."
- **204 No Content:** İstek başarıyla işlendi, ancak yanıt gövdesinde **geri dönecek herhangi bir içerik yok**. Genellikle PUT, PATCH veya DELETE gibi işlemlerden sonra kullanılır.
  - *Kullanım Durumu:* Bir kaynak başarıyla güncellendi veya silindi ve istemcinin ek bir veri beklemediği durumlarda.
  - *Örnek:* "Ürün başarıyla silindi, geri dönecek içerik yok."

#### 4.3. 3xx - Yönlendirmeler (Redirection Messages)

İstemcinin isteği tamamlamak için başka bir konuma yönlendirilmesi gerektiğini belirtir.

- **301 Moved Permanently:** Kaynak kalıcı olarak **yeni bir URI'ye taşındı**. İstemci, gelecekteki isteklerini yeni URI'ye göndermelidir.
  - *Kullanım Durumu:* Bir API endpoint'inin kalıcı olarak değiştirilmesi veya eski bir URI'nin kullanımdan kaldırılması.
  - *Örnek:* "Eski ürünler API'si yeni bir adrese taşındı, lütfen güncel adresi kullanın."
- **302 Found (Geçici Yönlendirme):** Kaynak geçici olarak başka bir URI'de bulunuyor. İstemci, sonraki istekleri için orijinal URI'yi kullanmaya devam etmelidir.
  - *Kullanım Durumu:* Geçici bakım, yük dengeleme veya A/B testi gibi durumlarda.
  - *Örnek:* "Bakım nedeniyle sayfa geçici olarak başka bir adrese yönlendirildi."

#### 4.4. 4xx - İstemci Hataları (Client Error Responses)

İstemci tarafından hatalı bir istek yapıldığını belirtir.

- **400 Bad Request:** Sunucu, istemci tarafından gönderilen isteği **anlayamadı veya işleyemedi** çünkü istek formatı geçersizdi, eksik bilgi içeriyordu veya anlamsızdı.
  - *Kullanım Durumu:* Geçersiz JSON, zorunlu bir alanın eksik olması, parametre formatının yanlış olması.
  - *Örnek:* "Geçersiz bir ürün adı gönderdiniz, lütfen tekrar kontrol edin."
- **401 Unauthorized:** İstemci, istenen kaynağa erişim için **kimlik doğrulamasını yapmadı veya başarısız oldu**. İstemcinin geçerli kimlik bilgileri sağlaması gerekir.
  - *Kullanım Durumu:* Geçersiz token, eksik Authorization başlığı.
  - *Örnek:* "Bu kaynağa erişmek için giriş yapmalısınız (geçersiz veya eksik kimlik bilgisi)."
- **403 Forbidden:** İstemcinin **kimlik doğrulaması yapmasına rağmen**, istenen kaynağa erişim yetkisi **yok**.
  - *Kullanım Durumu:* Bir kullanıcının yönetici yetkisi gerektiren bir işlemi yapmaya çalışması.

Bu rehber Acunmedya Akademi 13.Dönem Backend Sınıfı için [Engin Niyazi Ergül](#) tarafından hazırlanmıştır.

- Örnek: "Bu işlemi gerçekleştirmek için yeterli yetkiye sahip değilsiniz."
- **404 Not Found:** İstenen kaynak **sunucuda bulunamadı**.
  - *Kullanım Durumu:* Geçersiz bir URL'ye istek atılması, silinmiş bir kaynağa erişilmeye çalışılması.
  - Örnek: "Aradığınız ürün (ID: 10) bulunamadı."
- **405 Method Not Allowed:** Kaynak için belirtilen **HTTP metodu desteklenmiyor**. Örneğin, bir kaynağa POST metodu ile erişmeye çalışılması ancak o kaynağın sadece GET ve PUT metotlarını desteklemesi.
  - *Kullanım Durumu:* Bir API endpoint'ine yanlış HTTP metoduyla istek gönderilmesi.
  - Örnek: "Bu kaynak için 'DELETE' metodu desteklenmiyor."
- **409 Conflict:** İstek, **kaynağın güncel durumuyla çelişiyor**. Genellikle aynı anda birden fazla istemcinin aynı kaynağı güncellemeye çalıştığı durumlarda (çakışma) kullanılır.
  - *Kullanım Durumu:* Bir veritabanı kaydını güncellemeye çalışırken, bu kaydın başka bir işlem tarafından zaten değiştirilmiş olması.
  - Örnek: "Bu kaydı güncelleyemezsiniz çünkü başka bir kullanıcı tarafından değiştirildi."
- **429 Too Many Requests:** Belirli bir zaman dilimi içinde **çok fazla istek gönderildi**. API hız sınırlayıcıları (rate limiting) tarafından döndürülür.
  - *Kullanım Durumu:* Bir API'ye belirlenen limitin üzerinde istek atıldığında.
  - Örnek: "Çok fazla istek gönderdiniz, lütfen bir süre bekleyip tekrar deneyin."

#### 4.5. 5xx - Sunucu Hataları (Server Error Responses)

Sunucunun geçerli bir isteği işlerken bir hata ile karşılaştığını belirtir.

- **500 Internal Server Error:** Sunucu, isteği yerine getirirken **beklenmedik bir hata** ile karşılaştı. Bu genellikle sunucu tarafındaki bir kod hatası, veritabanı bağlantı sorunları gibi durumlarda meydana gelir.
  - *Kullanım Durumu:* Sunucu kodunda işlenmemiş bir istisna oluştuğunda.
  - Örnek: "Sunucumuzda beklenmedik bir hata oluştu, lütfen daha sonra tekrar deneyin."
- **503 Service Unavailable:** Sunucu şu anda isteği işleyemiyor, genellikle **aşırı yüklenme veya bakım nedeniyle geçici olarak kullanılamıyor**.
  - *Kullanım Durumu:* Sunucunun bakıma alınması, trafik yoğunluğu nedeniyle kaynak yetersizliği.
  - Örnek: "Hizmetimiz şu anda geçici olarak kullanılamıyor, lütfen kısa bir süre sonra tekrar deneyin."