



# RESTful API

## Neden kullanılır?

- Yazılım alt yapısını dışarı açmak
  - Yetkili veya yetkisiz
- Güçlü donanım gerektiren işlemleri istemci yerine sunucularda gerçekleştirmek için.
- Teknoloji bağımsız bir alt yapı oluşturmak için
  - İşletim sistemleri
  - Programlama dilleri
- Farklı uygulamaların birbiriyle ortak veri yapısı ile iletişim sağlamak için.
- İstemcilerin teknolojik bağımsızlığını sağlamak için.
  - Masaüstü
  - Mobil
  - Web
  - Konsol
  - Embedded vb.
- Farklı istemcilerin ihtiyaçlarına farklı endpoint'ler ile cevap verebilmek için.
  - XML
  - JSON
  - gRPC vb.
- Veri Güvenliği
  - Veri koruma, yedekleme vb.
  - Hacking saldırılarına karşı merkezi koruma

## Terminoloji

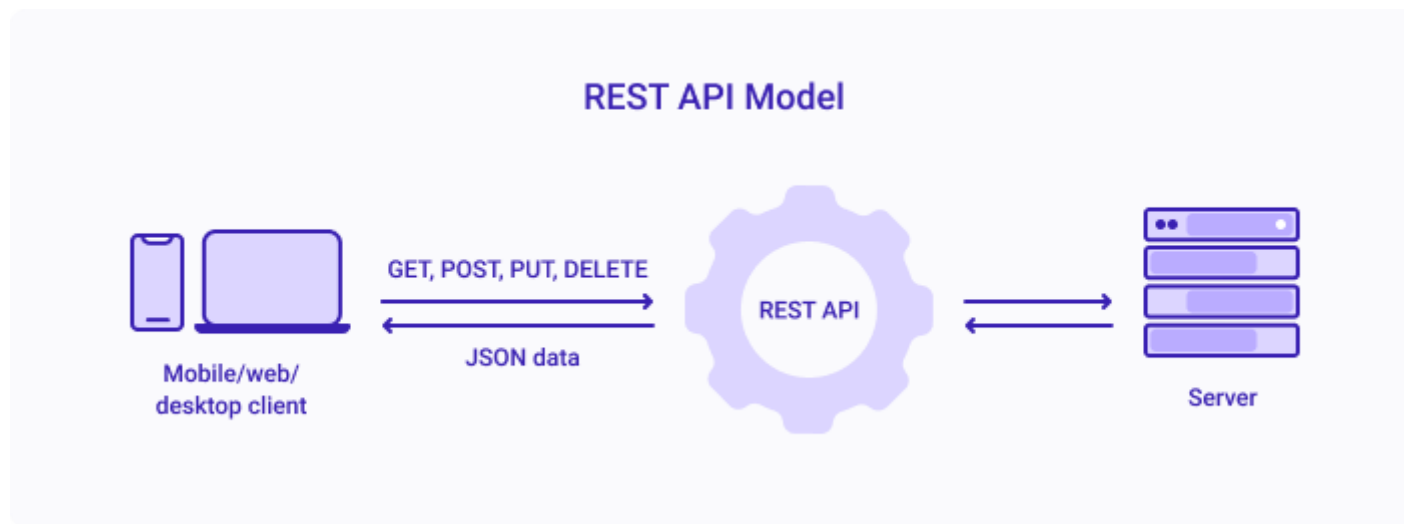
API: Application Program Interface

REST: Respresentational State Trasfer

Resource: API'den elde edilen bir veri parçasına denir.

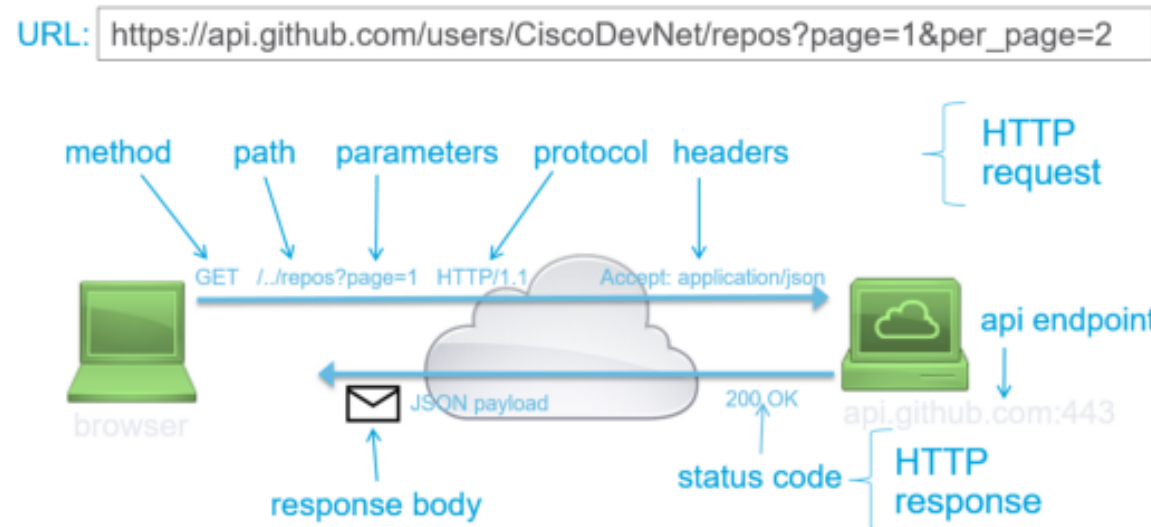
Request: Bir API'ye yapılan çağrıya denir. (POST, GET, PUT, DELETE)

Response: API'ye yapılan request sonucunda API'den dönen sonuca denir. (JSON, XML)



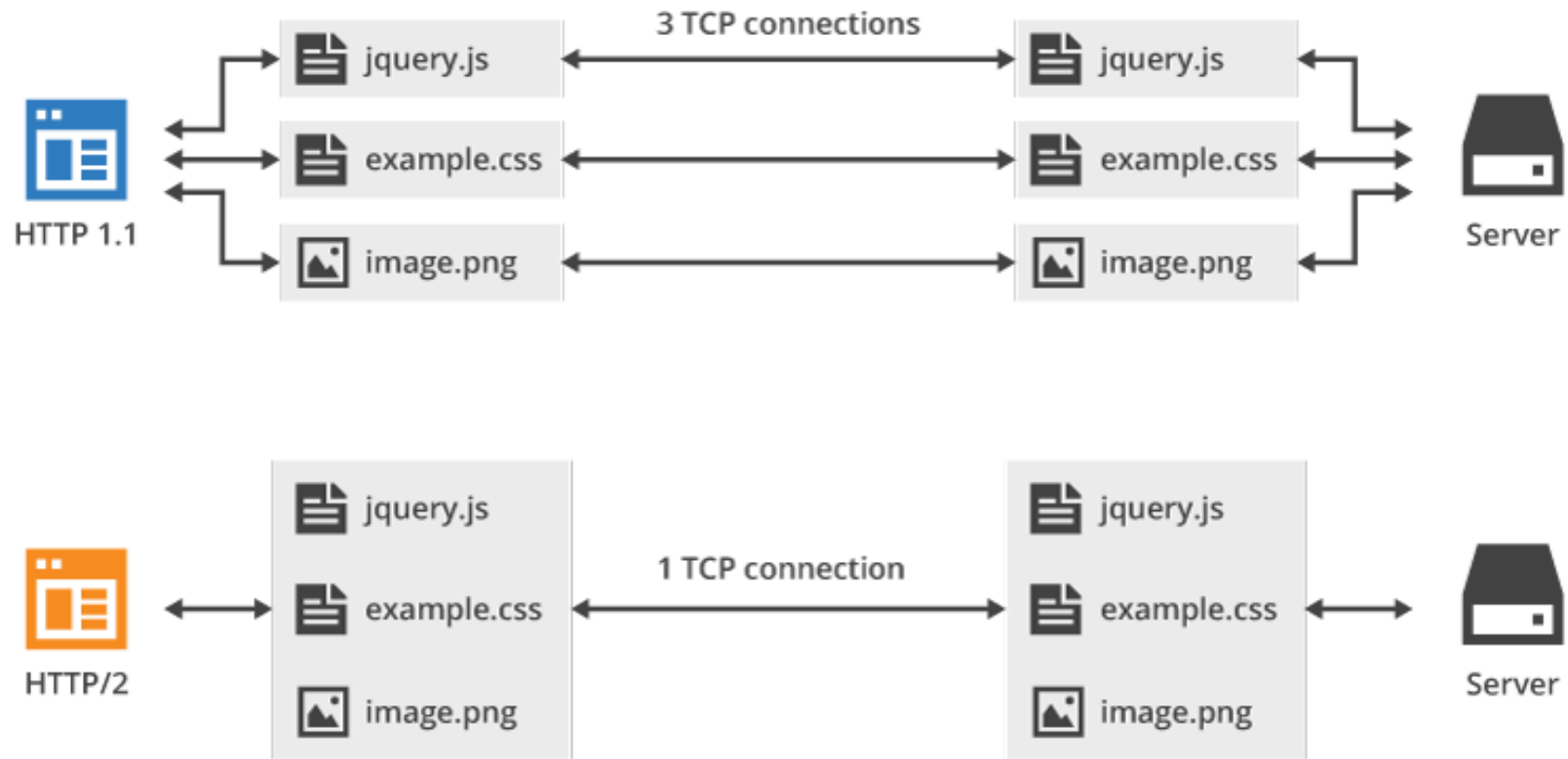
## Bir Request'in Anatomisi

### Anatomy of a REST API query



Notes:

## Multiplexing



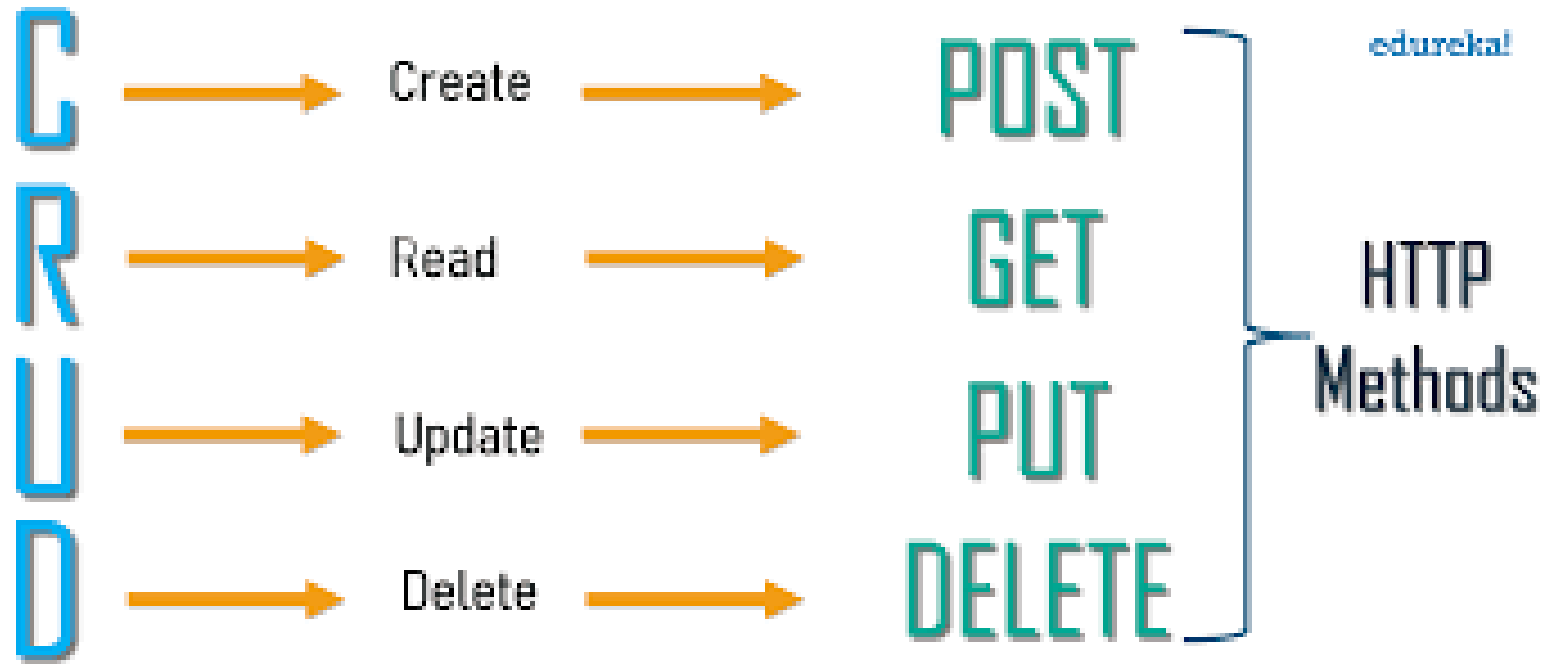
## Endpoint

- Sorgu oluşturabilmek için kullanılan URL'lerdir. Diğer adı route'dur.
- API'nin temel(base) URL'ine root-endpoint denir. Web ya da API gözetmeksizin base url de kullanılabilir.
- Path: API'den talep edilen kaynağı belirler. Endpoint içerisindeki bir URL parçasıdır.
  - Root-Endpoint: <https://api.github.com>
  - Path: /users/:username/repos
    - : path üzerindeki herhangi bir değişkeni belirtir.
- Sorgu parametreleri: Request'i düzenlemek için çeşitli seçenekler sunun parametredir.
  - Örnek Sorgu Parametre: ?query1=value1&query2=value2
  - Sorgu parametreleri teknik olarak REST mimarisinin bir parçası değildir. Ancak çok işlevseldir ve sık kullanılır.
  - Sorgu işarete ile başlar.
  - Birden fazla parametre ardarda kullanılabilir ve her parametre diğerinden ampersand(&) ile ayrılır.
  - Github API Örneği: <https://api.sedakayademir.com/users/kayademirs/repos?sort=pushed>

## HTTP Metotları

HTTP metotları gerçekleştirilen HTTP çağrısının tipini sunucuya bağlamak için kullanılır.

- GET
- POST
- PUT - PATCH
- DELETE



Diğer komutlar

- HEAD: GET ile aynı işlemi yapar. Farkı sadece status ve header bölümlerini aktarır.
- CONNECT: Verilen URI tarafından tanımlanan sunucuya bir tünel(tunnel) oluşturur.
- OPTIONS: Hedef kaynak için iletişim seçeneklerin açıklar.
- TRACE: Hedef kaynağa giden yol boyunca message loop-back testi gerçekleştirir.

API mimarisi her isteğin hangi HTTP metodunu kullandığını tutar ve bildirir.

## HTTP Header

- HTTP başlıkları(headers) hem istemci hem de sunucuya bilgi sağlamak için kullanılır.

method	path	protocol
GET	/tutorials/other/top-20-mysql-best-practices/	HTTP/1.1

```
Host: net.tutsplus.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: PHPSESSID=r2t5uvjq435r4q7ib3vtdjq120
Pragma: no-cache
Cache-Control: no-cache
```

HTTP headers as Name: Value

## Data (Body, Message)

- Data(bazen body veya message kullanılabilir) sunucuya gönderilmek istenen bilgileri içerir.
- Bu seçenek sadece POST, PUT, PATCH ya da delete istekleriyle kullanılır.

## REST Servisleri için Tasarım İlkeleri


- Basit tutmak
- İsimleri kullan, fiilleri değil
- Doğru HTTP metodunu seçmek
- Tekil değil, çoğul isimler kullanmak
- Parametreleri kullanmak
- Doğru HTTP kodları hayat kurtarır

- Versiyonlama kullanmak
- Sayfalama kullanmak
- Desteklenen formatlar kritik öneme sahiptir
- Veri ve hata mesajları standart olmalı

## HTTP Durum Kodları

### HTTP response status codes - HTTP | MDN

This interim response indicates that the client should continue the request or ignore the response if the request is already finished. 101 Switching Protocols This code is sent in response to an request header from the client and indicates the protocol the server is switching to.

 <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>

