



# **A**pplication **P**rogramming **I**nterface

**API**

**(Uygulama Programlama Arayüzü)**

**26/02/2022**



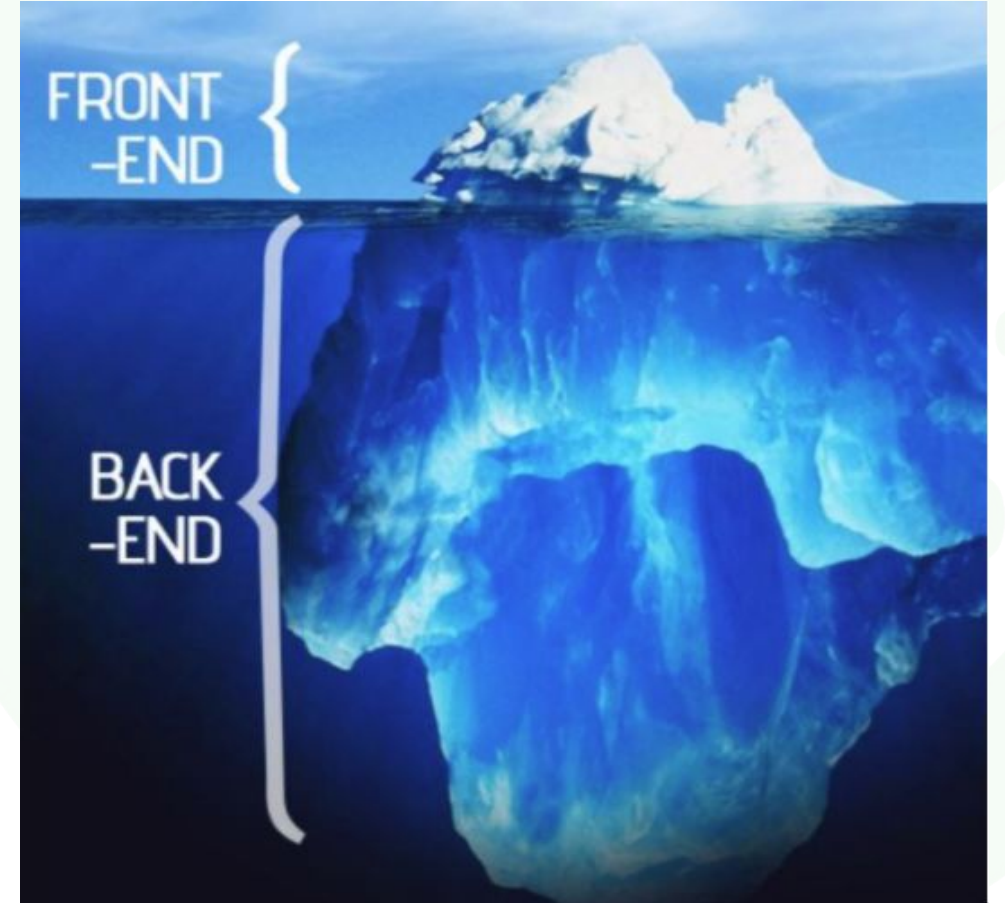
# DERSİN İÇERİĞİ

- Frontend, Backend
- **API nedir?**
- **Request, Response**
- **API nasıl çalışır?**
- API, Web Service
- http, https
- SSL Nedir?
- Http Status Code
- TCP
- TCP/IP
- End Point
- API Architectural Styles
- SOAP
- REST
- GraphQL
- Gateway
- **Postman**
- Postman'in Alternatifleri
- **Swagger Dökümanı**
- API İsimlendirilmesi
- **Path Param, Query Param**



# FRONTEND , BACKEND

- Frontend, bir web sitesine veya uygulamaya girdiğinizde; etkileşime girdiğiniz arayüzün, tasarım ve geliştirmesidir.
- Backend, bu web sitesinin veya uygulamanın perde arkasında yer alan işleyişin; server kısmı ve taban yazılımını geliştirme işine verilen adlardır.
- Steve Jobs: *“Tasarım bir şeyin yalnızca nasıl görüldüğü ve nasıl hissettirdiği ile ilgili değildir. Tasarım bir şeyin nasıl çalıştığıyla da alakalıdır.”*





# FRONTEND

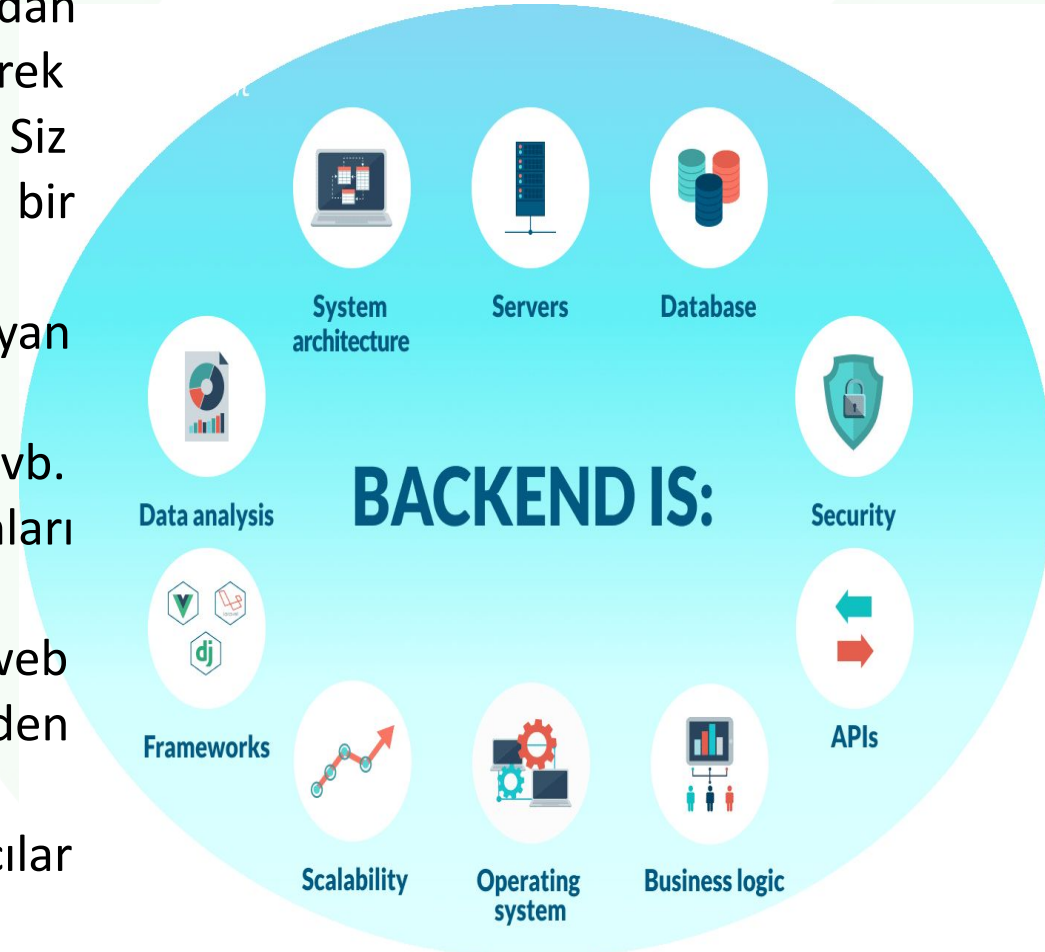
- Frontend, web sayfasında veya uygulamada, görüp etkileşime girebildiğiniz kısımlara verilen addır. Frontend genellikle uygulamanın ön yüzünün geliştirilmesini kapsar.
- Bir web sitesine girdiğinizde karşılaştığınız; renkli temalar, arka fonlar, yazı tipleri, tasarımsal görseller ve bunların kullanıcıya hitap edebilecek biçimde oluşturulması, sayfaya yerleştirilmesi gibi işlemlerin hepsi frontend yani önyüz olarak adlandırılır.
- Frontend'de yer alan bu öğelere eklenen bilgilerin depolanabilmesi, yani kısaca frontend'in hayata geçebilmesi için gereken alt yapı ve teknolojiyi sağlayan ise backend'dir.





# BACKEND

- Backend genellikle bir sunucu, bir uygulama ve bir veri tabanından oluşur. Bir havayolu veya otobüs firmasının, web sitesine girerek bilet aldığınızda; frontend ile etkileşime girmiş olursunuz. Siz bilgilerinizi web sitesine girdiğinizde, uygulama bu bilgiyi alır ve bir sunucu üzerinde kurulmuş olan veri tabanına depolar. Pegasus
- Sunucu, veri tabanı ve uygulamanın birlikte çalışmasını sağlayan kısma backend denir.
- Web'in backend kısmını oluşturmak için Java, PHP, Ruby, Python vb. yazılım dilleri ve MySQL, PostgreSQL ve Oracle gibi veri tabanları kullanılır.
- Frontend ve Backend geliştiriciler bir araya geldiklerinde bir web sitesi veya uygulama oluşturabilirler. Ancak her ikisi de birbirinden farklı işler yapar.
- Hem frontend hem de backend'i tek başına yapabilen yazılımcılar "Full-Stack Developer" olarak adlandırılır.

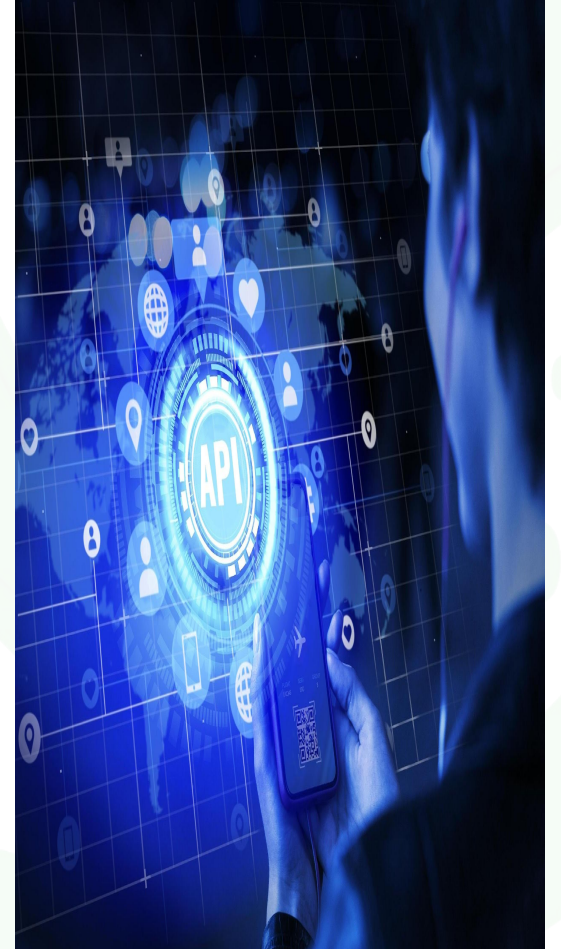


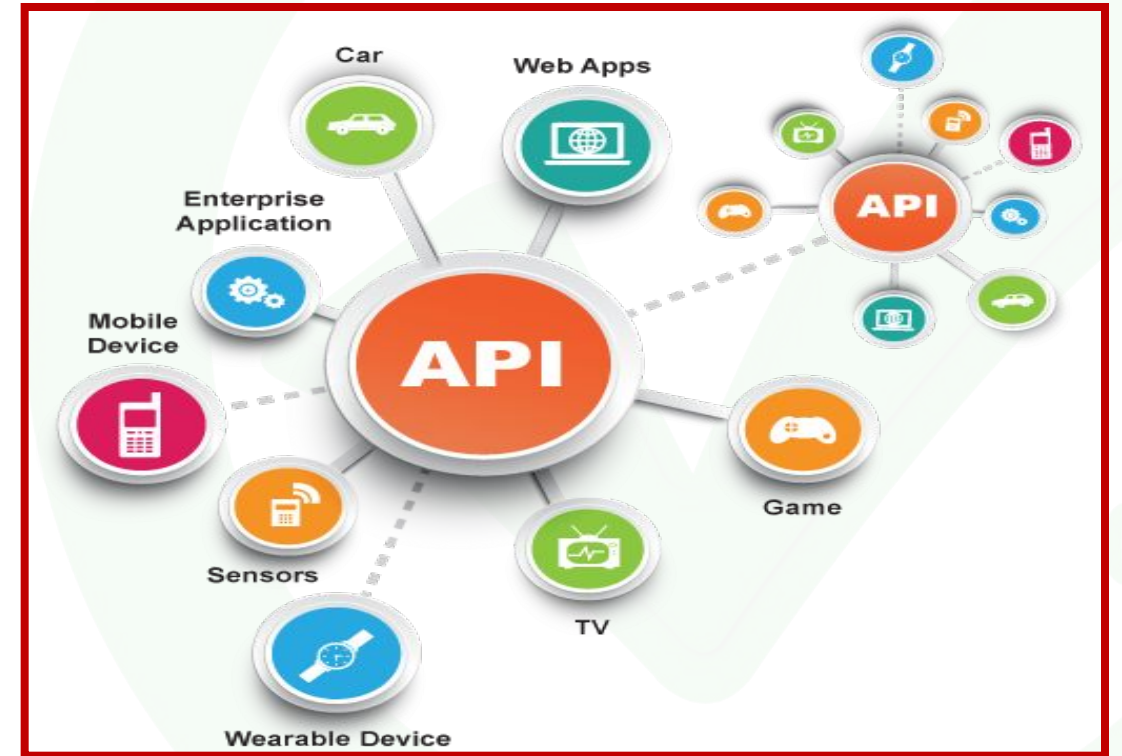
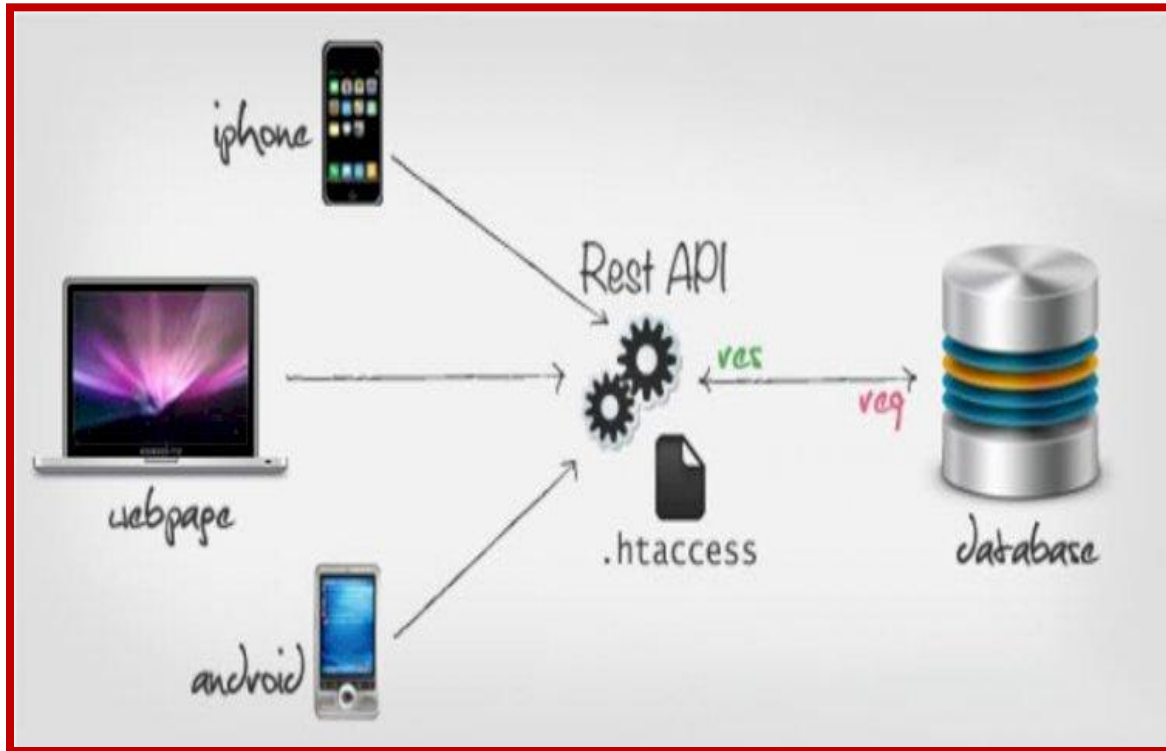


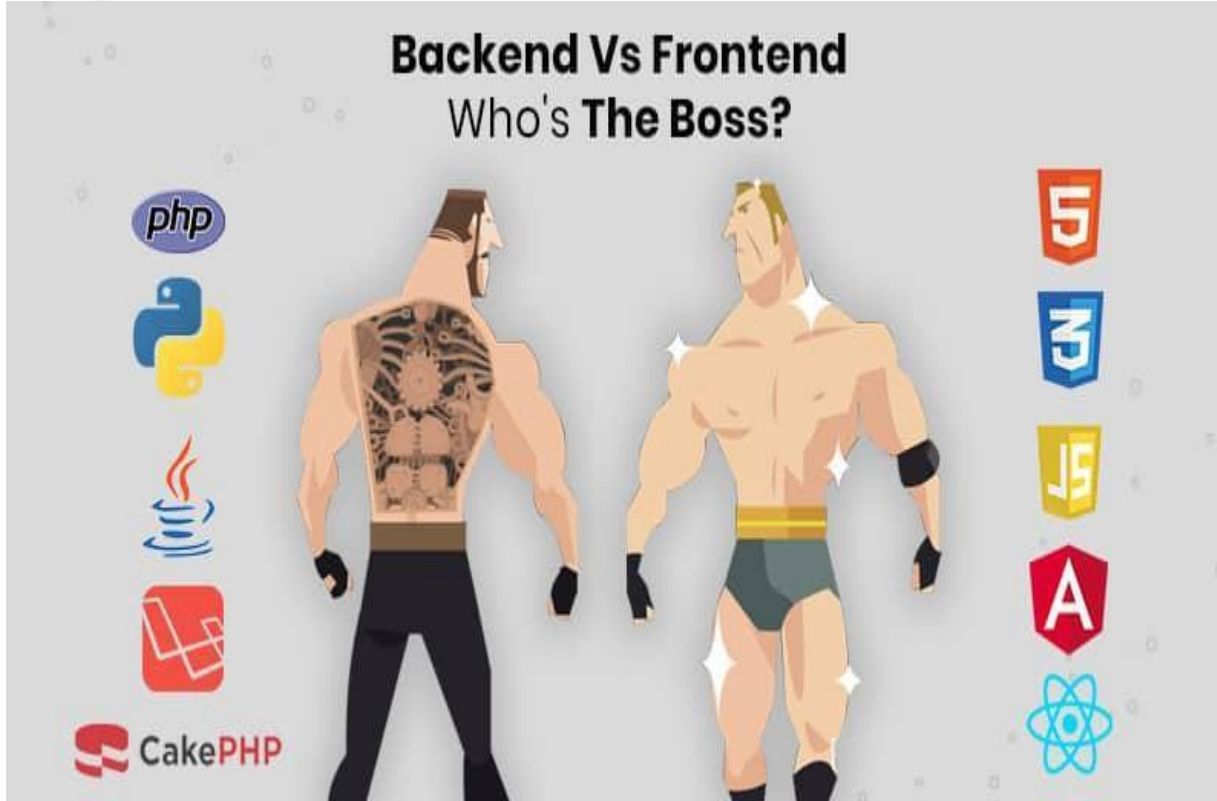


# API NEDİR?

- **API ( Application Programming Interface), bir yazılımın başka bir yazılımda; tanımlanmış işlevlerini kullanabilmesi için oluşturulmuş bir yapıdır.**
- **Bir uygulamanın başka bir uygulama ile iletişim kurmasına izin veren protokol** kümesidir.
- API; web uygulaması, işletim sistemi, veritabanı, donanımlar veya yazılım kütüphanesi için kullanılabilir.
- Günümüzde yoğunlukla web tabanlı uygulamalarda; istemci ve sunucu arasındaki iletişimi sağlayan bir yapı olarak kullanılmaktadır.
- İstemci spesifik bir formatta veri talep eder ve sunucudan yine belirli bir formatta cevap alır. Bu yapı Web API olarak da adlandırılır.





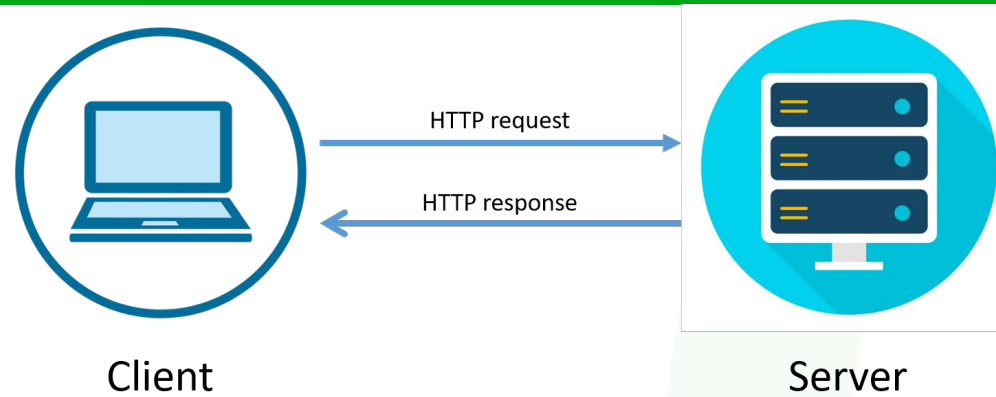


- API'nin UI (User Interface- Kullanici Arayuzu) yoktur.
- UI ile yaptığımız işlemleri, API ile kod yazarak yapabiliriz.
- Uygulamalar arası tüm bağlantıları koordine eden ve otomasyon ile önümüze getiren UI değil Backend'dir.





# REQUEST, RESPONSE

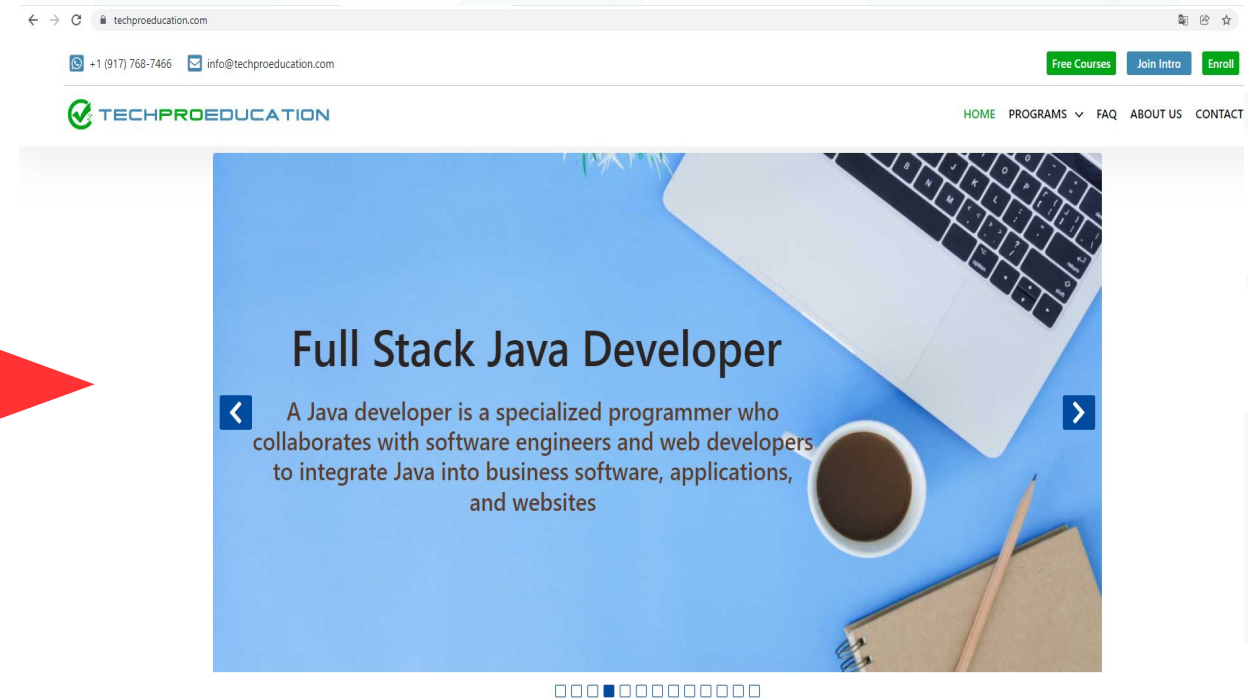


**Request:** İstek, Talep

**Response:** Yanıt, Cevap

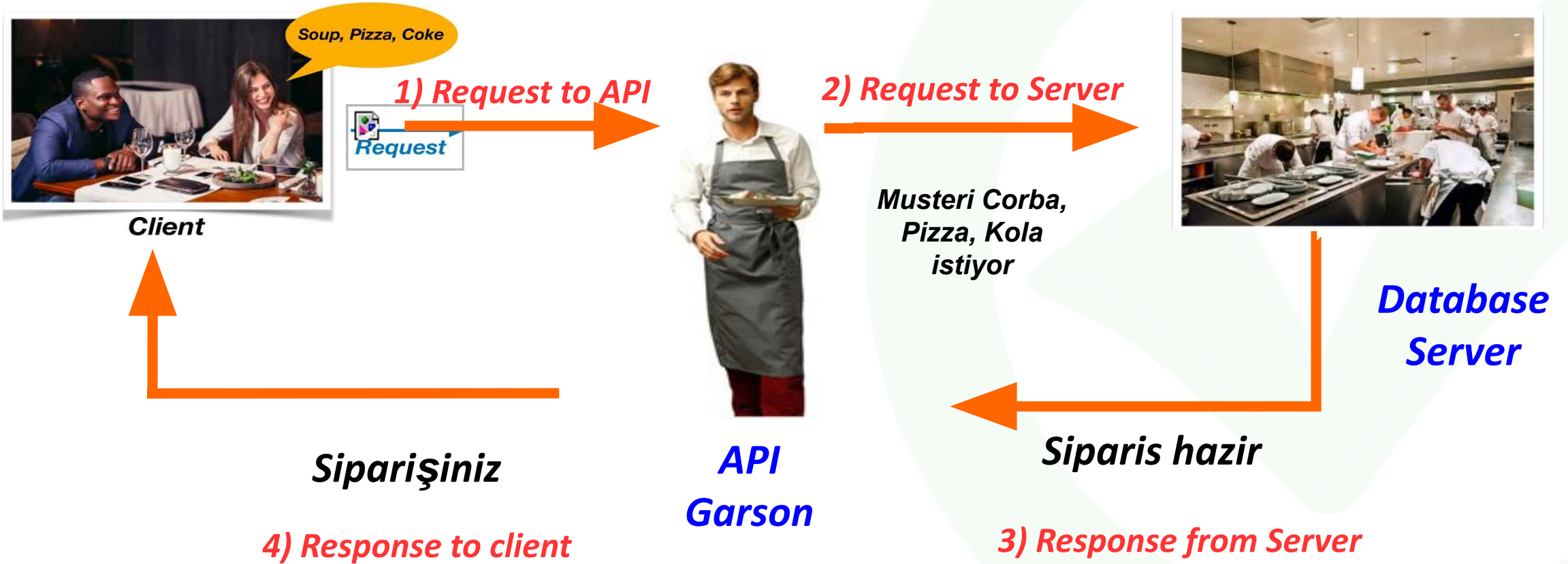
<https://www.techproeducation.com/>

UI



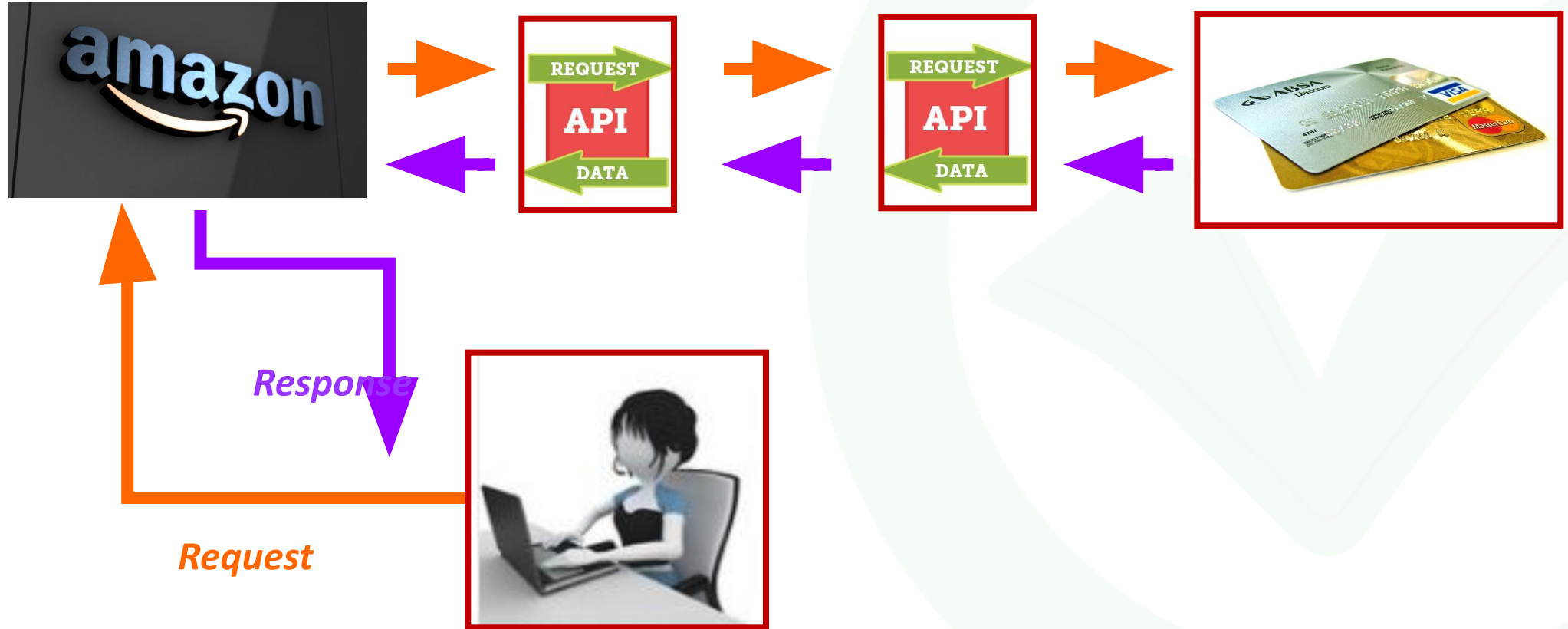


# API NASIL ÇALIŞIR?



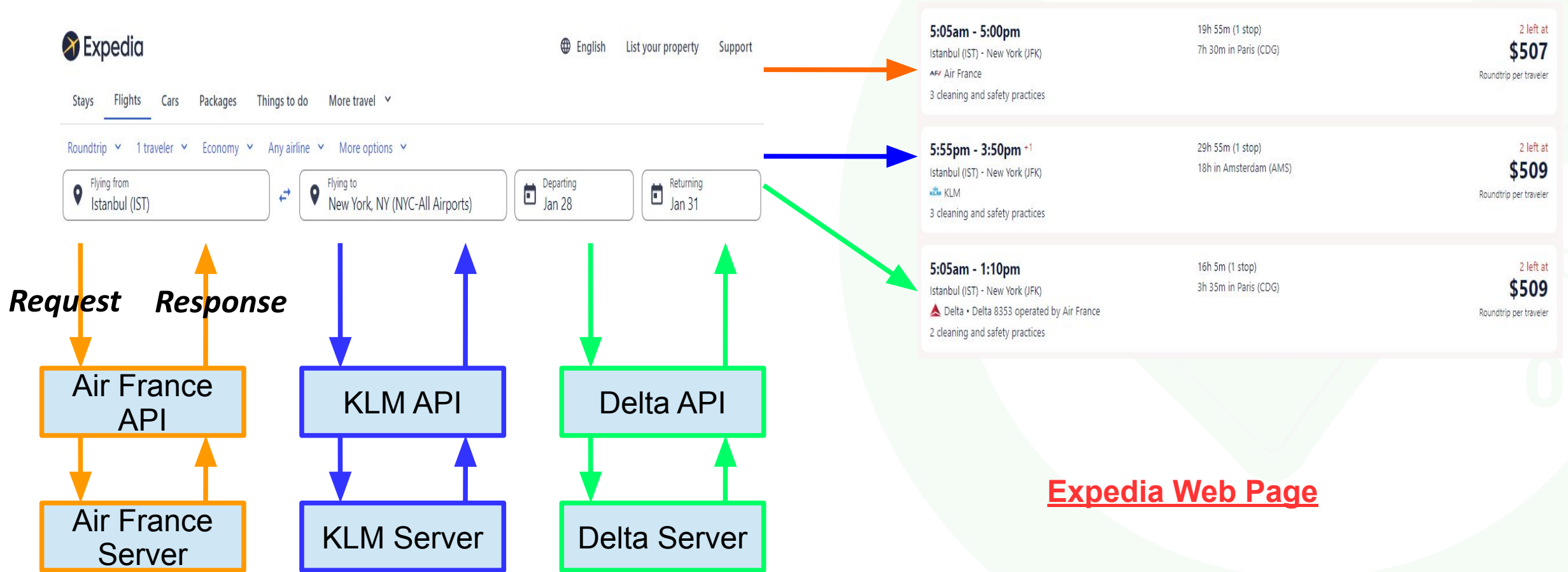


# API NASIL ÇALIŞIR?





# API NASIL ÇALIŞIR?





# API NASIL ÇALIŞIR?

**Kaydol** ×

Hızlı ve kolaydır.

Adın  Soyadin

Cep telefonu numarası veya e-posta

Yeni şifre

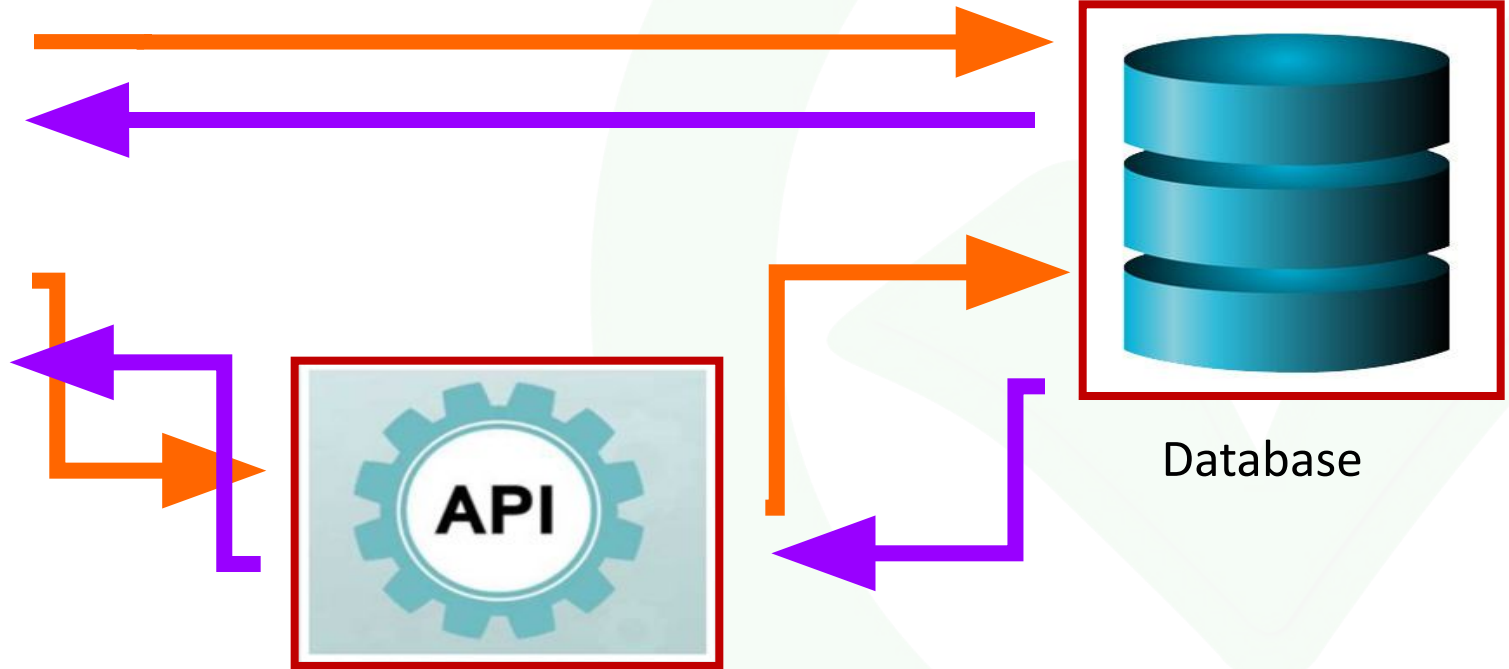
Doğum Tarihi ?  
1  Eki  2020

Cinsiyet ?  
Kadın ☐ Erkek ☐ Özel ☐

Kaydol düğmesine tıklayarak, Koşullarımızı, Veri İkimizi ve Çerezler İkimizi kabul etmiş olursun. Bizden SMS Bildirimleri alabilir ve bu bildirimleri istediğin zaman durdurabilirsin.

**Kaydol**

User Interface



API

API testi uygulamanın görüntüsü ile ilgilenmez, uygulamanın fonksiyon, performans ve güvenilirlik açısından beklentilere uygun çalışıp çalışmadığını test eder.





<https://developers.google.com/maps>

<https://developers.facebook.com/>

<https://developer.twitter.com/en>

<https://developers.garantibbva.com.tr/>

<https://sehirrehberi.ibb.gov.tr/developer/>

<https://apidocs.parasut.com/>

<https://developer.turkishairlines.com/>

<https://partner.turkcell.com.tr/>

<https://www.visualcapitalist.com/every-minute-internet-2020/>



# API, WEB SERVICE

- **Web Service** : İki makinenin bir ağ üzerinden birbirleri ile iletişim kurmak için kullandığı bir yöntemdir. Bilgisayarda çalışan bir web sunucusu diğer bilgisayarlardan gelen istekleri dinler. Başka bir bilgisayardan bir istek alındığında, bir ağ üzerinden web hizmeti istenen kaynakları döndürür. Bu kaynak JSON, XML, HTML dosyası, resimler, ses dosyaları vb. olabilir.
- API ve Web Service uygulamalar arasında iletişimi sağlar.
- **Web Service internet üzerinden iletişim sağlar. API ise internet olmadan da iletişim sağlayabilir.**
- Örneğin; Expedia, KLM Airlines DataBase'ine ulaşmak için internet kullanır (Web Service), bilgisayarımızdaki Microsoft Word gibi uygulamalar ise farklı uygulamalarla iletişim kurmak için kendi API'lerini kullanırlar. Intranet.
- **Tüm Web Servisler API'dır, ancak tüm API'lar Web Servis değildir.**
- Web servisleri, bir API'nin gerçekleştireceği tüm işlemleri gerçekleştiremeyebilir.
- Bir Web servisinin her zaman için bir ağ'a ihtiyacı olurken, bir API'nin çalışması için ağ'a ihtiyacı yoktur gerekmez.



# HTTP, HTTPS

- Bilginin; sunucudan kullanıcıya nasıl ve ne şekilde aktarılacağını gösteren protokoldür. Açılımı “Hyper Text Transfer Protocol” (Üstün Metin Transfer Protokolü). Kullanıcılar, bunu aktif olarak kullanmasa da otomatik olarak browser'lar ;bu protokolü, girilen adrese ekler.
- HTTP 1990 yılından beri dünya çapında ağ üzerinde kullanılan bir iletişim protokolüdür.
- HTTP protokolü ağ üzerinden web sayfalarının görüntülenmesini sağlayan protokoldür.
- HTTP protokolü istemci (PC) ile sunucu (Server) arasındaki veri alışverişi kurallarını belirler.
- Client ve Server arasındaki tüm iletişim request ve response'lar ile olur.





# SSL NEDİR?

**SSL (Secure Socket Layer):** Güvenli Giriş Katmanı anlamına gelen SSL, web siteniz ve ziyaretçileriniz arasındaki iletişimin; şifrelenmiş bir bağlantı üzerinden gerçekleştirilmesini amaçlar.

2018 yılında itibaren, Google Chrome gibi popüler tarayıcıların, SSL sertifikası olmayan web sitelerini “güvenli değil” şeklinde etiketlemeye başlamasıyla; SSL kullanımı yaygınlaşmıştır.



Kullanıcı 1

**HTTP**

http://www.example.com

şifre : abc123



**Şifreleme olmadan**

korsanların gördüğü şifre “abc123”



Kullanıcı 2

**HTTPS**

https://www.example.com

şifre : abc123



**Şifreleme varken**

korsanların gördüğü şifre “xgeaDarz”



# HTTP STATUS CODE

- Kullanıcılar bir web sitesini ziyaret etmek istediklerinde iki taraflı bir iletişim ortaya çıkar.
- Bu iletişimin bir tarafında tarayıcı bulunurken diğer tarafta sunucu yer alır.
- Bir web sayfasına giriş yapan kullanıcı aslında tarayıcı aracılığıyla ilgili web sayfasının yer aldığı sunucuya, sayfayı görüntülemek için bir istek gönderir.
- Sunucu ise bu isteğe üç haneli bir durum kodu ile yanıt verir.
- Sunucunun tarayıcıya verdiği üç haneli cevaplar HTTP durum kodları, HTTP status codes olarak adlandırılır.



[HTTP Status Codes](#)





# TCP (Transmission Control Protocol)

- TCP (Transmission Control Protocol) bilgisayarlar arasındaki iletişimin, küçük paketler hâlinde ve kayıpsız olarak gerçekleştirilmesini sağlayan bir protokoldür.
- Aslında TCP (Transmission Control Protocol) protokolünün en önemli özelliği; kimlik doğrulaması yapması ve veriyi karşı tarafa gönderirken veya alırken verinin bütünlüğünü sağlamasıdır. Gelişmiş bilgisayar ağlarında ortaya çıkan kayıpları önlemek için TCP protokolü yazılmıştır.
- HTTP, HTTPS, POP3, SSH, SMTP, TELNET ve FTP gibi günlük hayatta sıkça kullandığımız protokollerin veri iletimi TCP vasıtasıyla yapılır.



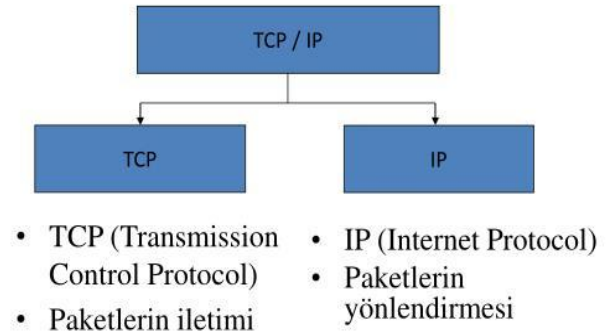


# TCP/IP

- TCP (Transmission Control Protocol) ve IP (Internet Protocol) protokollerinin birleştirilmesiyle oluşturulan internet üzerindeki bir iletişim metodudur.
- Bu metot sayesinde internete bağlanan tüm cihazlar birbirleri ile haberleşebilir. **Bir ağa bağlanan bilgisayarlar veri iletmek ve almak için birbirleri arasında TCP/IP protokolü ile haberleşmektedir.**
- TCP/IP protokolü, bilgisayarlar arası veri iletişiminin kurallarını koyan bir iletişim protokolleri bütünüdür. Bilgisayarlar arası iletişim farklı protokol aileleri veya tipleri üzerinden gerçekleştirilir. Kullanım amacına göre ise bu protokoller birbirlerinden ayrılmaktadır.

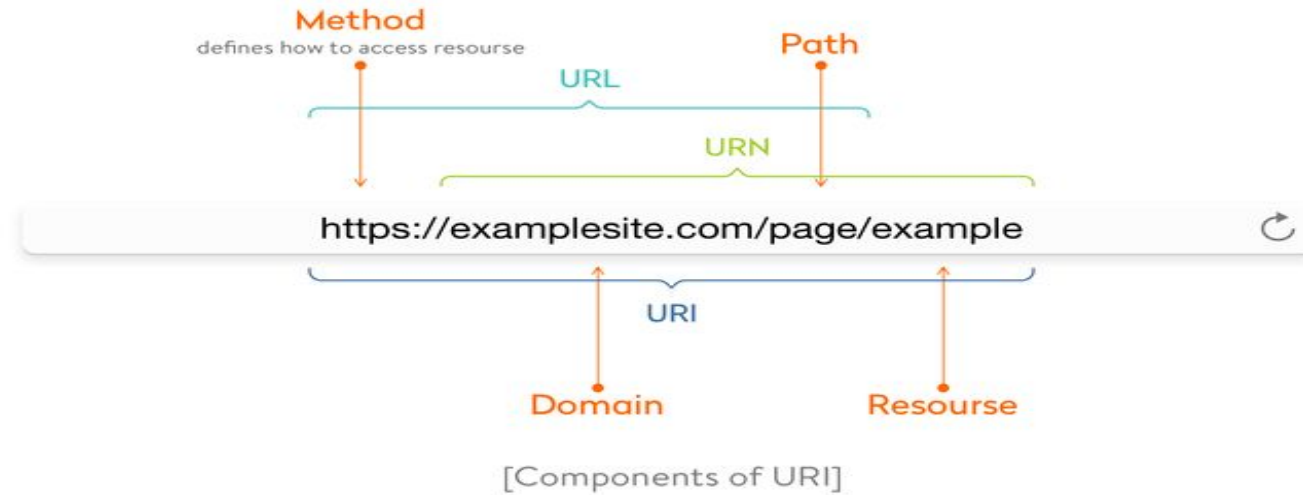
## TCP/IP

İnternet üzerinden temel veri alış-veriş protokolleridir.





# END POINT



- End point kaynağa nasıl erisebileceğimizi gösteren URI (Uniform Resource Identifiers)'lara denir.
- Bir API oluşturduğunuzda kullanıcıların ulaşabilmesi için bu endpoint'i ve kullanılacak Http method'larını kullanıcılara bildirmemiz gerekir.
- URI günlük hayatta kullandığımız URL (Uniform Resource Locator)'a benzer.
- URL kullanıcının görebileceği ve etkileşimde bulunacağı bir web sayfasını ifade ederken URI o sayfanın içerdiği bilgiyi ifade eder.




# API ARCHITECTURAL STYLES





# API ARCHITECTURAL STYLES

	RPC	SOAP	REST	GraphQL
Organized in terms of	local procedure calling	enveloped message structure	compliance with six architectural constraints	schema & type system
Format	JSON, XML, Protobuf, Thrift, FlatBuffers	XML only	XML, JSON, HTML, plain text,	JSON
Learning curve	Easy	Difficult	Easy	Medium
Community	Large	Small	Large	Growing
Use cases	Command and action-oriented APIs; internal high performance communication in massive micro-services systems	Payment gateways, identity management CRM solutions financial and telecommunication services, legacy system support	Public APIs simple resource-driven apps	Mobile APIs, complex systems, micro-services

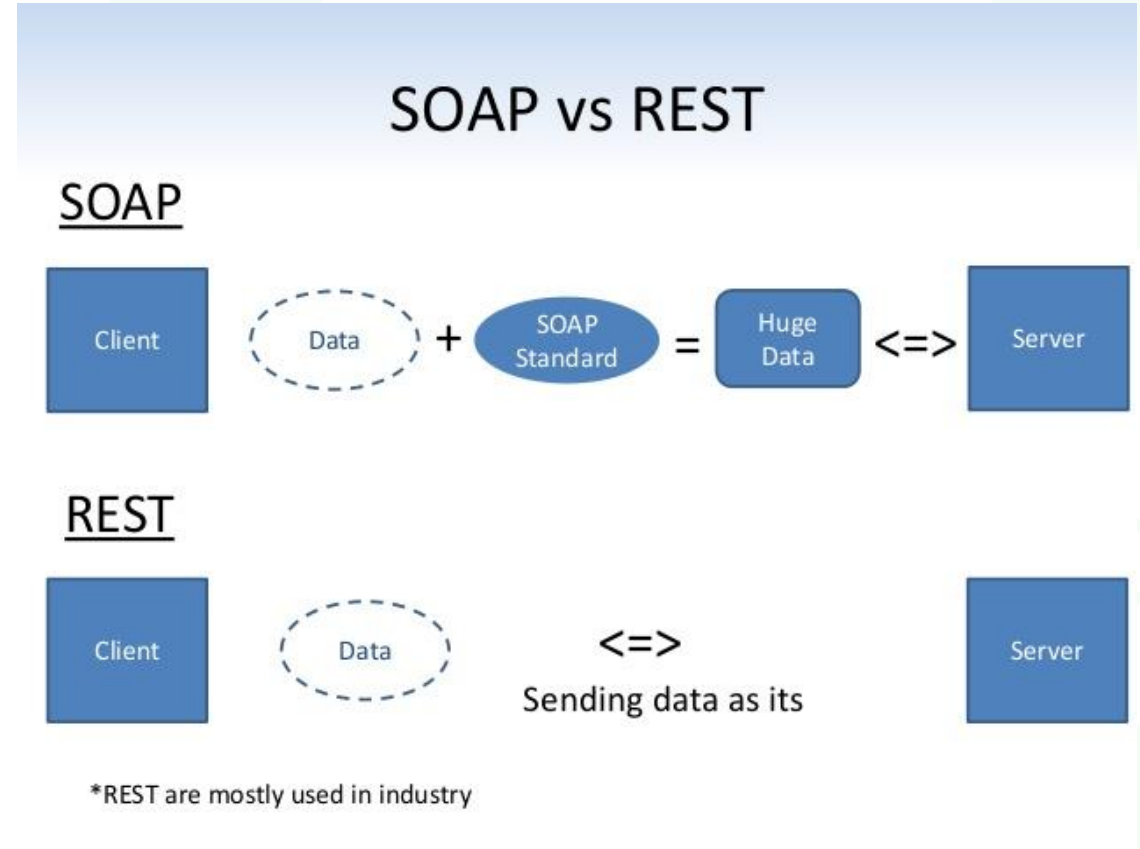
**altexsoft**  
software r&d engineering





# SOAP, REST

- Web servis mimarisinin temeli HTTP üzerine kurulmuştur. Genel olarak web servise bir istek gelir ve web servis bu isteği yapıp, bir sonuç döndürür.
- Web servisin bu işlemi yapabilmesi için tanımlanmış farklı yöntemler bulunmaktadır. Bu yapılardan biri SOAP protokolü diğeri ise REST'dir.





# SOAP (Simple Object Access Protocol)

- SOAP (Simple Object Access Protocol- Basit Nesne Erişim Protokolü) uygulamalar ile web servislerin bilgi aktarımını sağlayan, XML tabanlı bir protokoldür.
- Web servise giden bilgi XML olarak gönderilir, web servis bu bilgiyi yorumlar ve sonucunu XML olarak geri döndürür. SOAP tabanlı bir web servisin, gönderilen XML verisini nasıl yorumlayacağını tanımlanması gerekir. Bu web servis tanımlaması WSDL (Web Service Description Language- Web Servisleri Tanımlama Dili) standardı ile yapılır.

```
<customer>
  <customer_id> 1001 </customer_id>
  <customer_name> Mark Star </customer_name>
</customer>
```

```
<?xml version="1.0" encoding=
<definitions name="AktienKurs
  targetNamespace="http://loc
  xmlns:xsd="http://schemas.xmlsoap.or
  xmlns="http://schemas.xmlsoap.org/wsd
  <service name="AktienKurs">
    <port name="AktienSoapPort" binding
      <soap:address location="http://loc
    </port>
    <message name="Aktie.HoleWert">
      <part name="body" element="xsd:Tra
    </message>
    ...
  </service>
</definitions>
```

**WSDL**



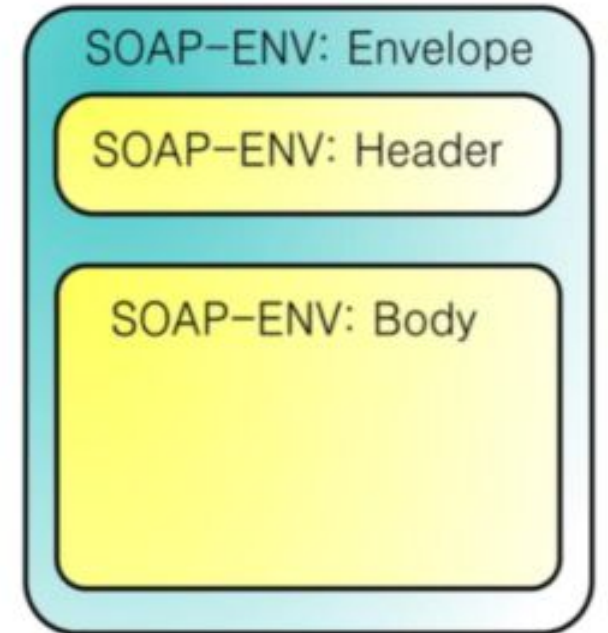
# SOAP (Simple Object Access Protocol)

Bir SOAP mesajı 3 şeyi içerir.

**Envelope (Zarf):** Tüm mesaj içeriğinin içinde olduğu kısımdır. SOAP mesajları XML formatında olduğundan bir root elemanı olmalıdır. Envelope, SOAP mesajının root etiketidir de denebilir. Envelope'un içinde Header ve Body gibi kısımlar da bulunur.

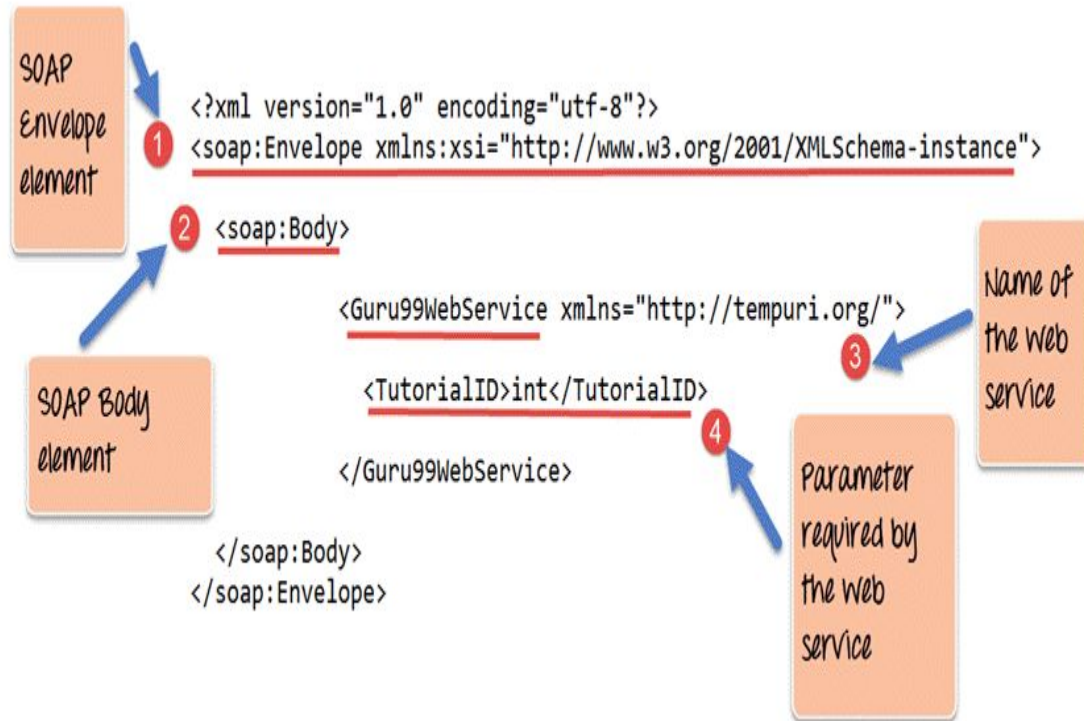
**Header (Başlık):** Klasik html'de bulunan <head></head> etiketine benzetilebilir. Bu bölümde mesajın meta-data bilgileri gönderilir.

**Body:** Soap mesajının ana içeriğini barındıran kısımdır. Bu bölümde metotlarla ilgili bilgiler veya metodun sonucu yer alır.





# SOAP (Simple Object Access Protocol)

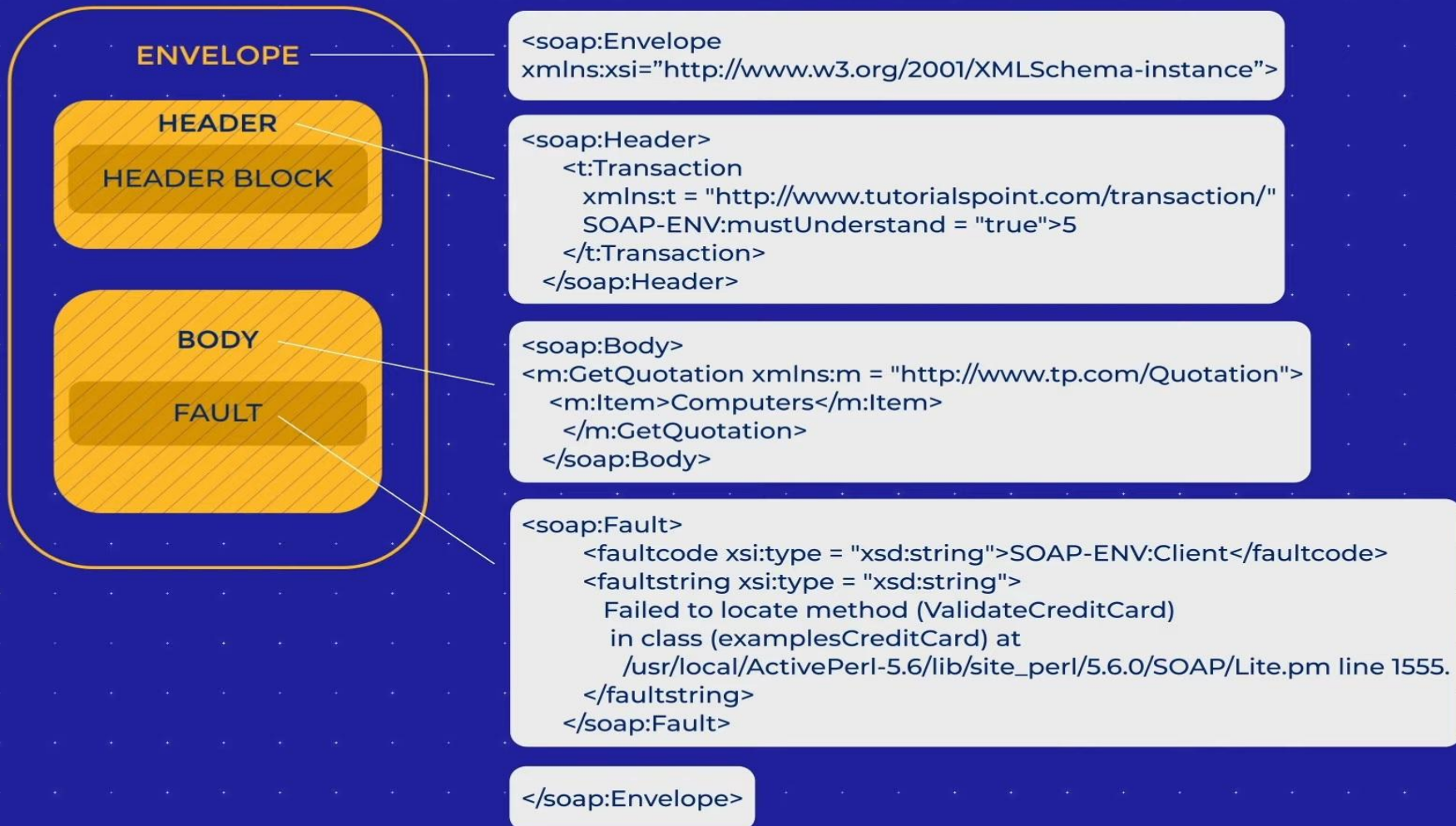


```
<?xml version="1.0"?>  
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2001/12/soap-envelope" SOAP-ENV:encodingStyle=" http://www.w3.org/2001/12/soap-encoding">  
  <soap:Body>  
    <Guru99WebService xmlns="http://tempuri.org/">  
      <TutorialID>int</TutorialID>  
    </Guru99WebService>  
  </soap:Body>  
</SOAP-ENV:Envelope>
```





# SOAP (Simple Object Access Protocol)

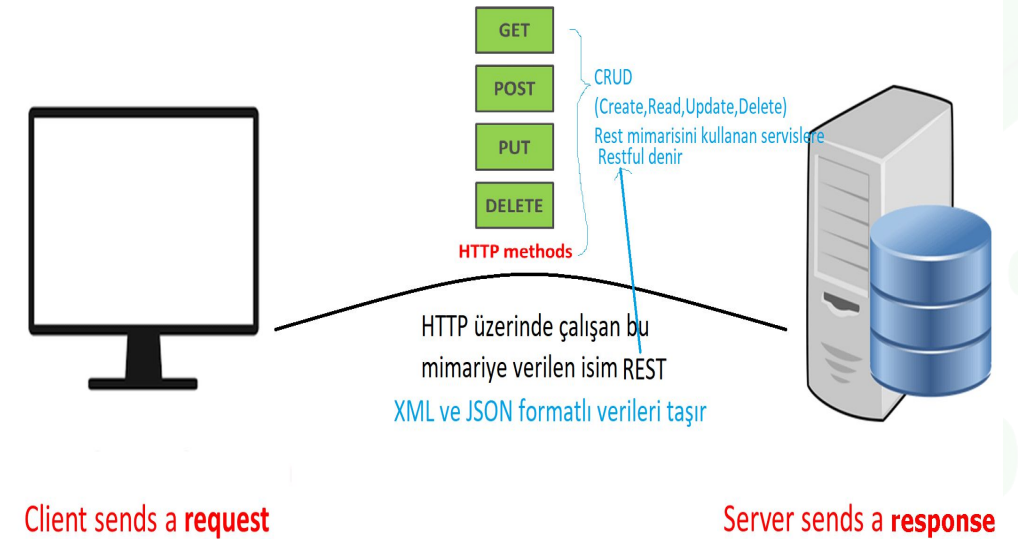






# REST (REpresentational State Transfer)

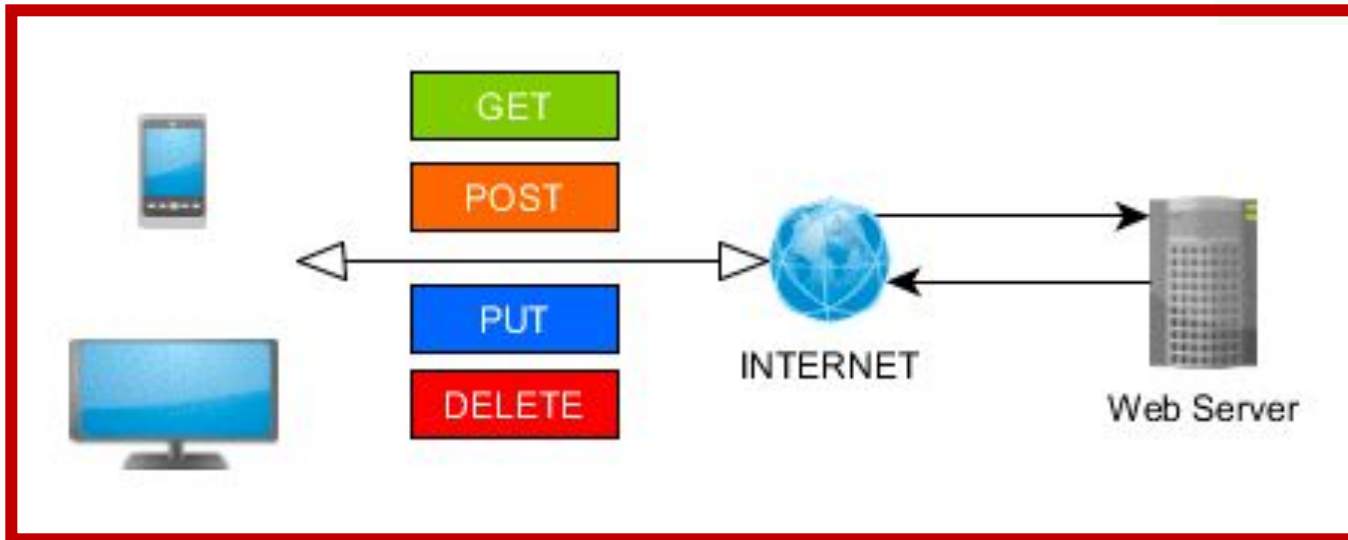
- REST (Representational State Transfer) mimarisinde ise işlemler resource kavramıyla yapılır. Resource URI (Uniform Resource Identifiers) ile tanımlanır. Yani REST'te SOAP'ta olduğu gibi XML yardımıyla metodlar çağırılmaz bunun yerine o metodu çağırarak URI'ler ile web servise HTTP protokolüyle istek yapılır. Böylece işlemler tamamen HTTP metodları üzerinden yapılır.
- Örneğin, bir web servisin metodunu SOAP ile "getProductName" şeklinde çağırırken REST ile "/products/name/1" URI'si ile çağırabiliriz. Ayrıca RESTin döndürdüğü veri tipinin de XML olması zorunlu değildir JSON (Java Script Object Notation) , XML, TXT, HTML gibi istenen veri türünde değer döndürülebilir.
- REST standartlarına uygun yazılan web servislerine RESTful servisler denir.





# REST (REpresentational State Transfer)

REST API, SOAP ve WSDL tabanlı web servislerinin daha basite indirgenmiş hali olarak nitelendirilebilir. Bu amaçla; internet tarayıcısı sayfa işlemlerinin bir parçası olan HTTP protokolünü (GET, POST, PUT DELETE gibi talep tipleri) kullanır. Böylelikle REST API geliştiricilere yaygın, pratik ve oldukça esnek bir kullanım olanağı sunduğu gibi, minimum içerikle veri alıp gönderdiği için de daha hızlıdır.



```
{
  "id": 114351,
  "firstName": "Della",
  "lastName": "Heaney",
  "middleInitial": "Russell Homenick V",
  "email": "ricardo.larkin@yahoo.com",
  "mobilePhoneNumber": "123-456-7893",
  "phoneNumber": "213-456-7893",
  "zipCode": "53081",
  "address": "75164 McClure Stream",
  "city": "Gislasonburgh",
  "ssn": "823-25-7239",
  "createDate": "2021-12-05T23:00:00Z",
  "zelleEnrolled": true,
  "country": {
    "id": 3,
    "name": "USA",
    "states": null
  },
  "state": "New York43",
  "user": null,
  "accounts": null
},
```



# REST (REpresentational State Transfer)

**HTTP metodların REST ile kullanımı:** REST tabanlı web servislerde, HTTP metodlarına özel anlamlar yüklenir ve böylece web servise bir HTTP isteği geldiği anda metod çalıştırılmış olur. Bu durumda HTTP metodlarının REST ile nasıl kullanılacağı önemlidir.

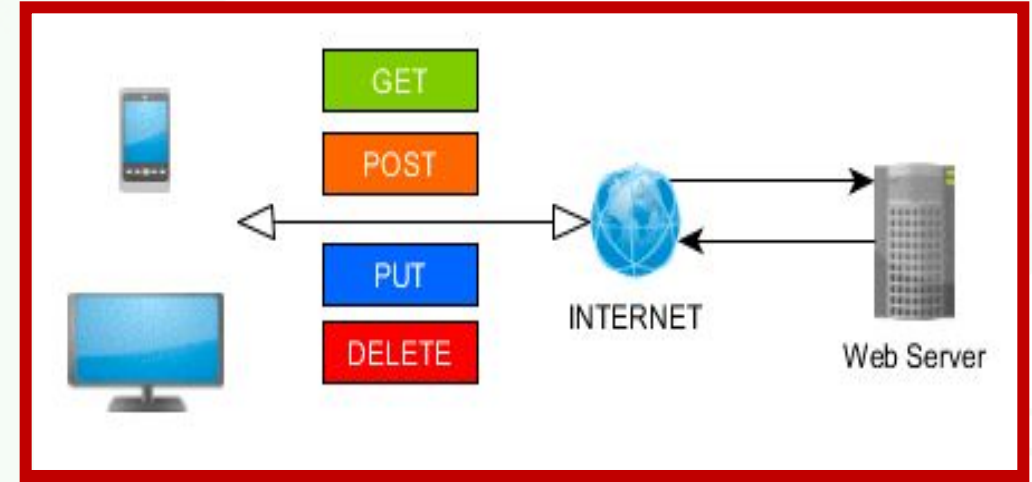
**Örnek:** <https://www.gmibank.com/api/tp-customers/114351>

**GET:** Adresi verilen nesneyi döndürmek için kullanılır. Bu metot kullanıldığında kayıtlarda bir değişiklik yapılmaz.

**PUT:** Var olan bir nesneyi değiştirmek için veya eğer yoksa yeni bir tane oluşturmak için kullanılır.

**POST:** Yeni bir nesne oluşturmak için kullanılır. Her seferinde yeni bir nesne oluşturur. PUT ile yapılan bir işlem POST ile de yapılabilir fakat aralarındaki fark tarayıcıların bu iki metodu farklı yorumlayıp iki metot için farklı tepkiler verebilmesidir.

**DELETE:** Adresi verilen nesneyi silmek için kullanılır.





# REST (REpresentational State Transfer)

En yalın haliyle bir HTTP isteğini aşağıdaki bilgileri taşır:

**Request-Line:** HTTP isteğin türü, hangi url'e yapılacağı ve http/https protokolü bilgisi

**Header:** Yapılan isteği niteleyen ve isteğe ait temel bilgileri içeren parametreleri taşır. Gönderilmesi zorunlu değildir, bir ve birden fazla header parametresi gönderebiliriz.

**Body:** Eğer POST, PUT, PATCH gibi API üzerinden bazı kayıt ve işlemler yapılmasını istiyorsak, bu bilgileri de isteğimizin Body alanında göndeririz. Restful bir API'da, JSON olarak bu bilgileri gönderilmesi tercih edilir.



# REST (REpresentational State Transfer)

GET <https://www.gmibank.com/api/tp-customers/114351>

```
1  {
2    "id": 114351,
3    "firstName": "Della",
4    "lastName": "Heaney",
5    "middleInitial": "Russell Homenick V",
6    "email": "ricardo.larkin@yahoo.com",
7    "mobilePhoneNumber": "123-456-7893",
8    "phoneNumber": "213-456-7893",
9    "zipCode": "53081",
10   "address": "75164 McClure Stream",
11   "city": "Gislasonburgh",
12   "ssn": "823-25-7239",
13   "createDate": "2021-12-05T23:00:00Z",
14   "zelleEnrolled": true,
15   "country": {
16     "id": 3,
17     "name": "USA",
18     "states": null
19   },
20   "state": "New York43",
21   "user": null,
22   "accounts": [
```

```
23   {
24     "id": 2333,
25     "description": "musteri omer hesap3",
26     "balance": 69700,
27     "accountType": "CREDIT_CARD",
28     "accountStatusType": "ACTIVE",
29     "createDate": "2020-11-06T23:00:00Z",
30     "closedDate": "2024-11-06T23:00:00Z",
31     "employee": null,
32     "accountlogs": null
33   },
34   {
35     "id": 107250,
36     "description": "New Account_6thGenQA_01",
37     "balance": 11190,
38     "accountType": "CHECKING",
39     "accountStatusType": "ACTIVE",
40     "createDate": "2021-11-24T23:00:00Z",
41     "closedDate": "2022-11-24T23:00:00Z",
42     "employee": null,
43     "accountlogs": null
44   }
45 ]
46
```





# SOAP, REST

SOAP ve REST karşılaştırması yapmak ve her iki web servisinin artı ve eksilerini değerlendirmek gerekirse;

- REST ile JSON, XML, HTML, TEXT ile çalışmaktadır. SOAP ile sadece XML formatında çalışabilir. JSON ile daha ufak boyutlardaki verilerle işlemler kolayca gerçekleşebilir.
- SOAP, isteklerini oluşturmak için W3C standardı olan bir WSDL dili kullanmak zorundadır. Rest'te buna gerek yoktur. İstekler URI üzerinden oluşturulabilir. Bir dile ihtiyaç duymadığından REST'in kullanımı ve tasarlanması daha kolaydır.
- SOAP'ı öne çıkaran en önemli özelliklerden biri güvenliği sağlamak daha hızlı ve kolay. Hazır yapıları mevcuttur. Güvenlik konusu REST yapısında daha komplekstir.
- Uygulama hızını önemsiyorsanız kesinlikle REST'i kullanabilirsiniz. Hız bakımından SOAP'a göre oldukça hızlıdır.
- SOAP, XML-Scheme kullanırken REST, URI-scheme kullanır.
- Hem SOAP hemde REST'te HTTP protokolünü kullanılır. REST için HTTP kullanma zorunluluğu bulunmaktadır. Fakat SOAP'ta TCP, SMTP gibi başka protokollerde kullanılabilir.
- SOAP'ta kaynakça bulmak REST'e göre daha fazladır.
- Test ve hata tespitlerinde REST'i kullanmak daha kolaydır. HTTP hatalarını döndürür ve herhangi bir toola ihtiyaç duyulmadan görülebilir. SOAP'ta ise hata ayıklama toollarına ihtiyaç duyulabilir.

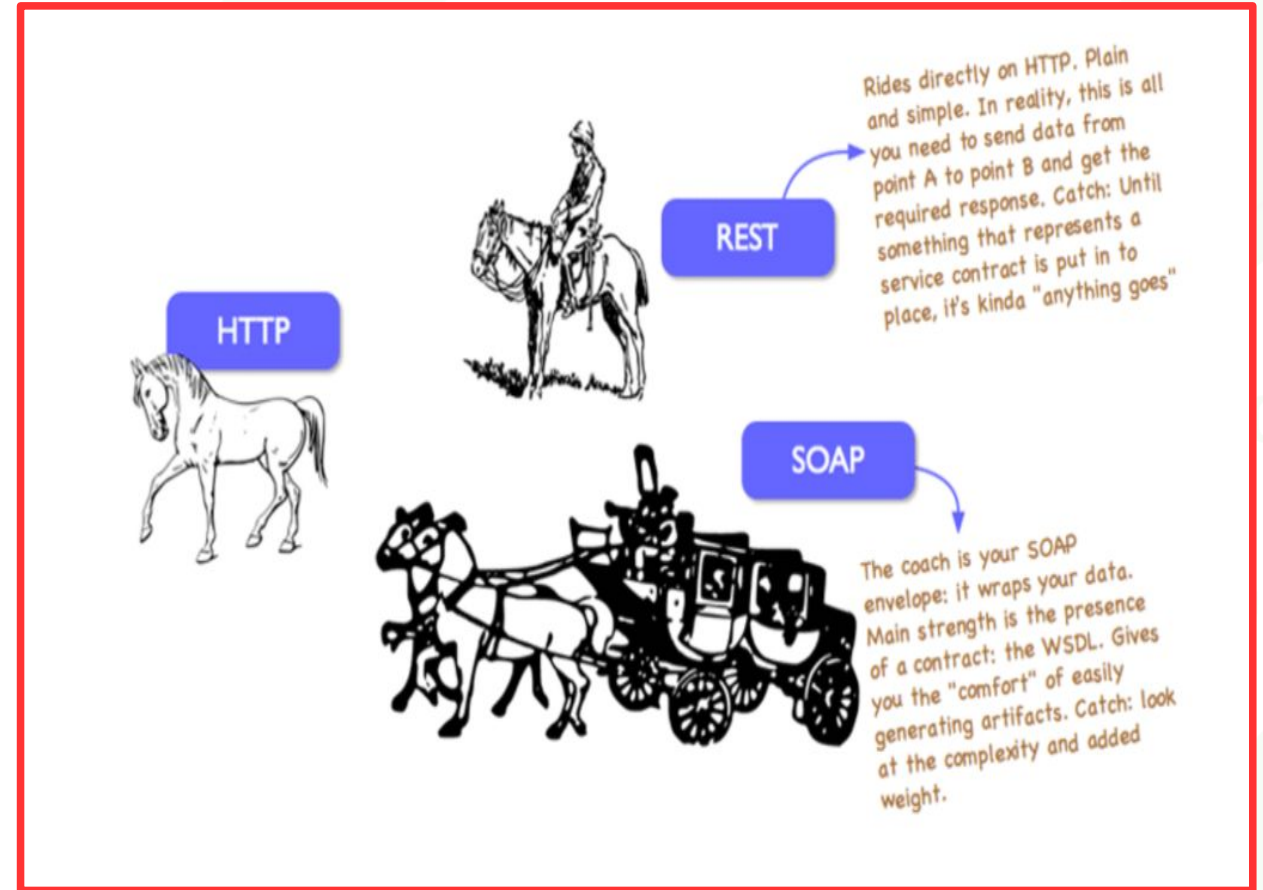
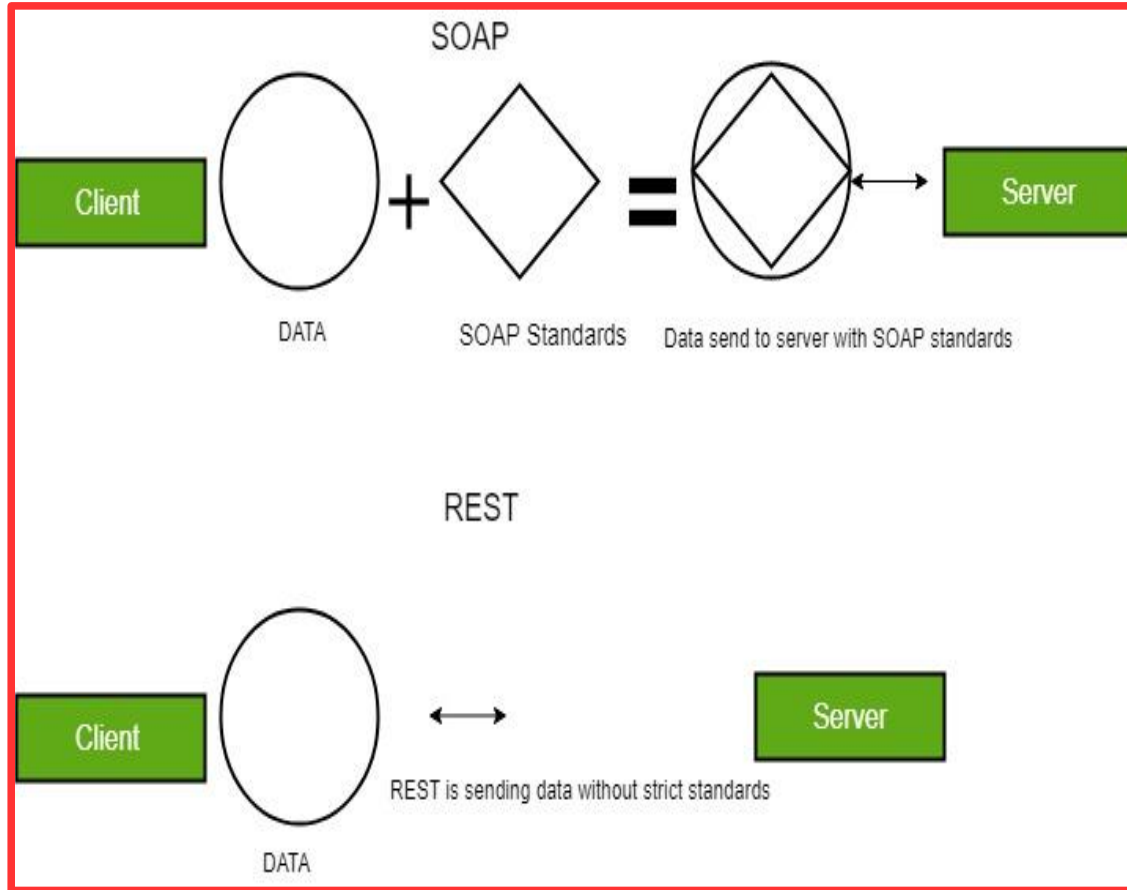


# SOAP, REST

SOAP	REST
SOAP bir protokoldür	HTTP protokolünü kullanan bir mimaridir
WDSL ile tasarlama yapmak gerektiğinden kullanması daha zor	HTTP methodları ile tasarlandığı için kullanması daha kolay
Yalnızca XML format kullanır.	XML , JSON , HTML , TXT format kullanır
Önbelliği okuyamaz ( can not be cached)	Önbelliği okur (can be cached)
Rest e göre daha yavaş	Soap' a göre daha hızlı
Finansal,iletişim ve ödeme noktalarında kullanılır	Sosyal medya, Web Chat , Mobil uygulamalar da kullanılır
Rest e göre daha güvenlidir.	



# SOAP, REST





# SOAP, REST

## Use in Technology driven sectors



### REST

- Social Media
- Web Chat
- Mobile



### SOAP

- Financial
- Telecommunication
- Payment Gateways

A postcard is faster and cheaper!

POST

It takes a few extra steps, to access a postcard



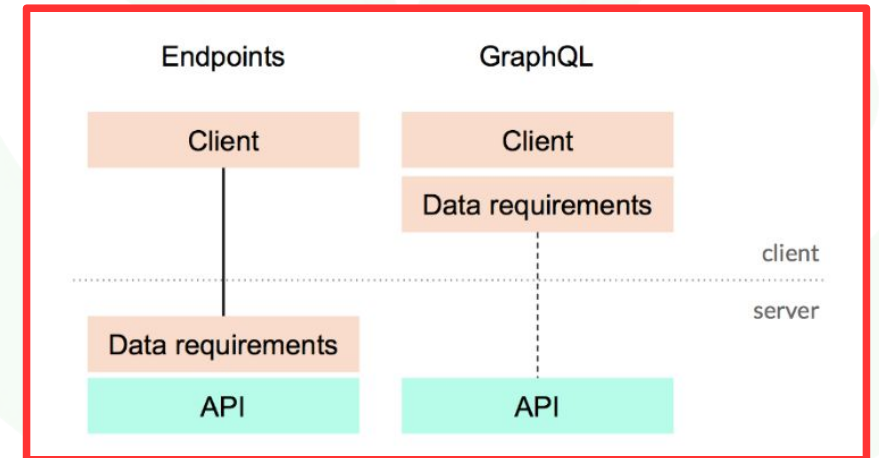


# GraphQL

- GraphQL, Facebook tarafından geliştirilmiş; verimli, etkili ve esnek bir alternatif sunan yeni bir API standartıdır. Açık kaynaklı olarak geliştirilen GraphQL'i, birçok şirket aktif olarak kullanmaktadır.
- GraphQL, API tasarlamak ve kullanmak için bir yol sunmaktadır. İstemcinin ihtiyaç duyduğu verilerin tam olarak belirtilmesini sağlayarak birden fazla kaynaktan veriye ulaşmanızı kolaylaştırmayı hedefler.
- GraphQL, genel kullanıma sunulduğundan bu yana büyüyen ve popüler hale gelen ve REST'e göre bazı büyük avantajları olan bir sorgu dilidir. Ana odak noktası, performansı optimize etmek ve esnekliği artırmaktır.



GraphQL







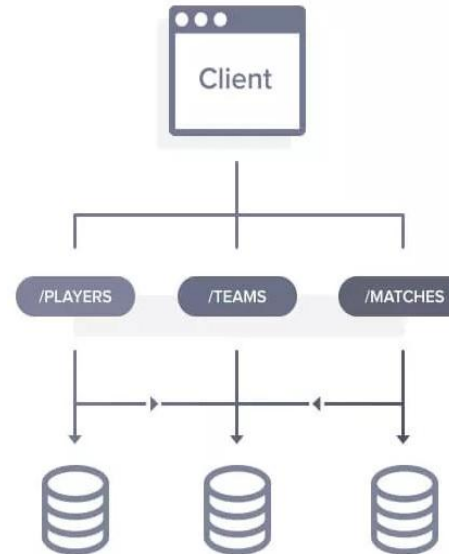
# GraphQL

GraphiQL

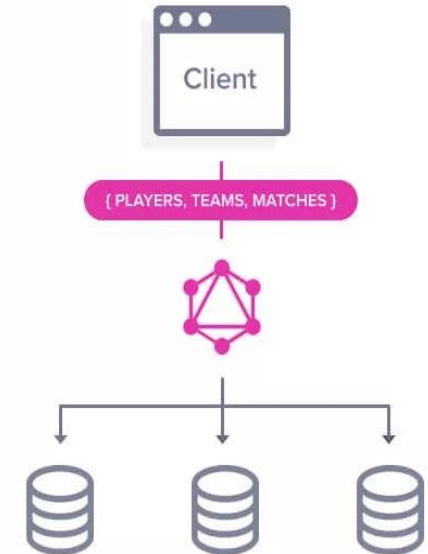
```
1 {  
2   artist{  
3     lastName  
4     firstName  
5     age  
6   }  
7 }
```

```
{  
  "data": {  
    "artist": {  
      "lastName": "Sutton",  
      "firstName": "Tierney",  
      "age": 54  
    }  
  }  
}
```

Rest API



GraphQL API



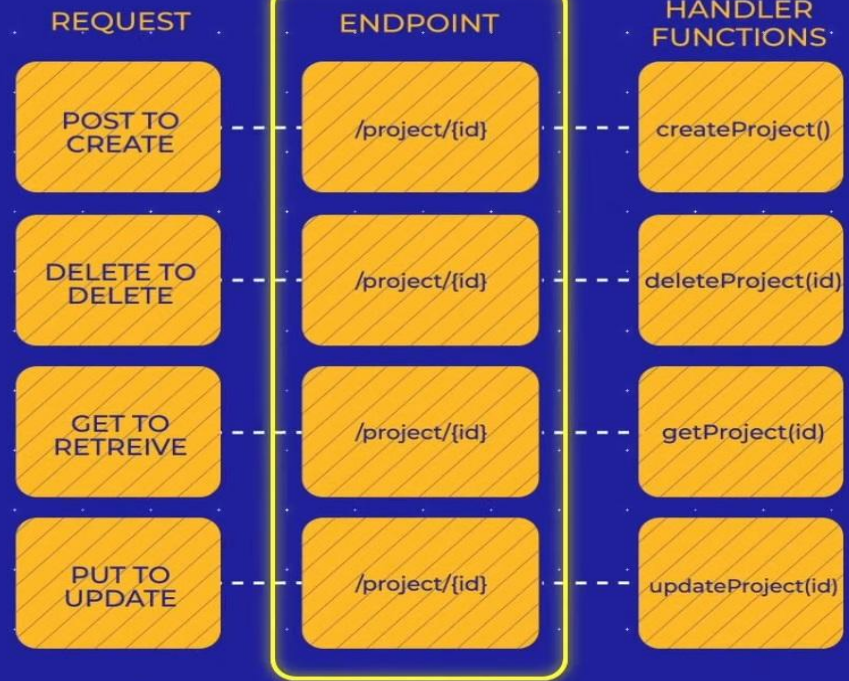
[GraphQL Web Sites](#)



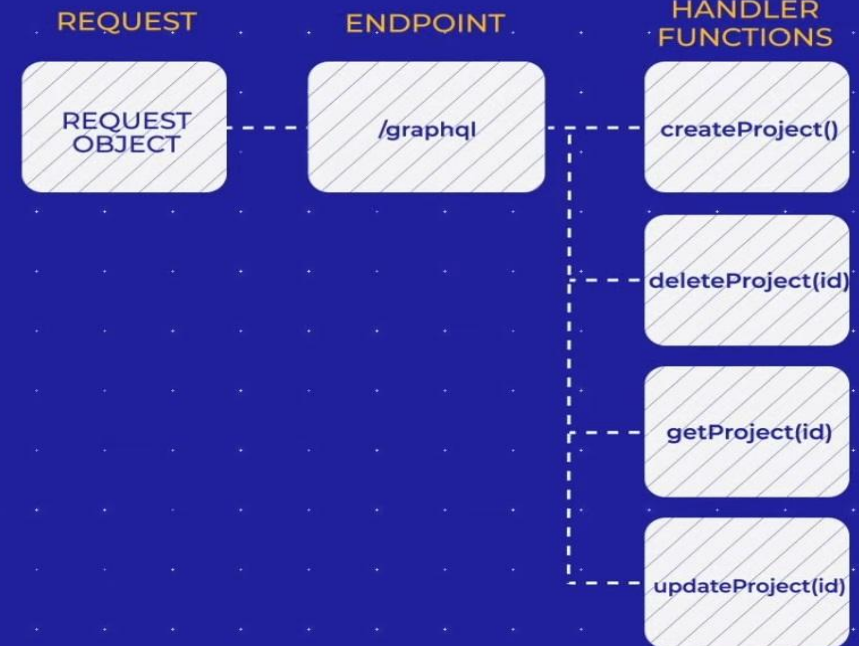
# GraphQL

GraphQL

## REST



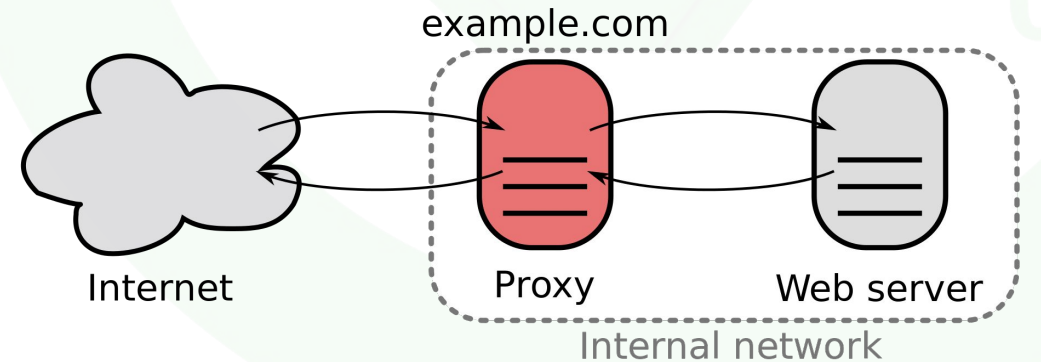
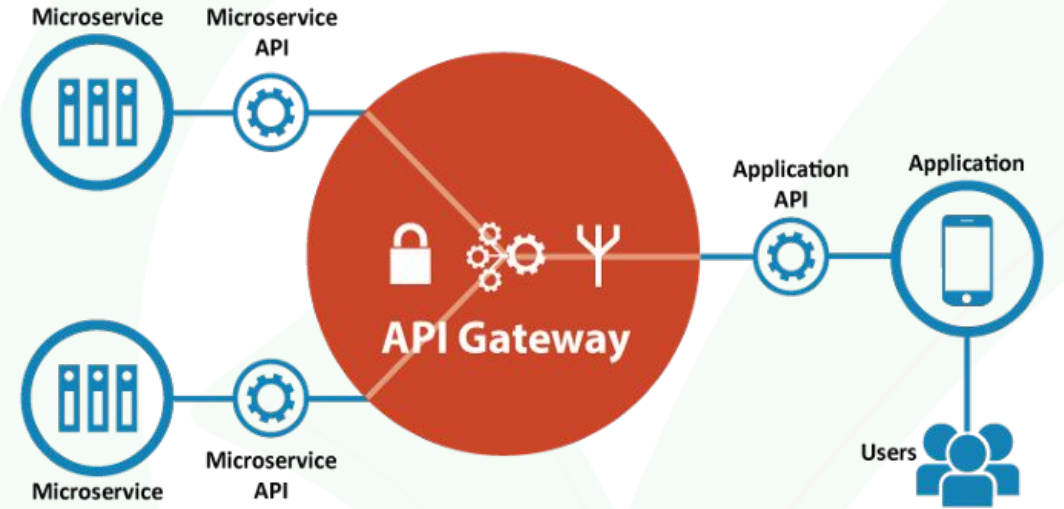
## GRAPHQL





# GATEWAY

- API Gateway, istemciler ile backend sunucuları veya mikro servisler arasında duran bir API yönetim aracıdır.
- API Gateway API isteklerini alarak çeşitli kurallara göre uygun servislere yönlendiren bir ters vekil sunucusu (reverse proxy) olarak çalışır.
- API Gateway istek sınırlandırma, istatistik, kimlik doğrulama vs. çeşitli sık kullanılan işlevleri üzerine alarak asıl API sunucularınızın önünde bir üst katman oluşturur.





# POSTMAN

- Postman, backend API'lerini test etmek için oldukça kullanışlı bir tool'dur.
- Elinizde var olan bir API'yi hızlıca test etmek ve sonuçlarını incelemek isterseniz; Postman kullanıcı dostu arayüzü ile işlerinizi oldukça kolaylaştırır.
- Hazırladığınız request'leri export edip arkadaşlarınızla da paylaşabilirsiniz.
- Çok az Javascript bilgisi ile hızlıca testler yazabiliyor, yazdığınız testleri kaydedip sonra tek bir tık ile çalıştırabiliyorsunuz.



POSTMAN



# POSTMAN

- Postman, API geliřtirmek için bir iřbirlięi platformudur.
- Postman'ı kendi bařınıza veya takım arkadaşlarınız ile birlikte API'ları tasarlamak, geliřtirmek, oluřturmak ve test etmek için kullanabilirsiniz.
- Rest, SOAP ve GraphQL sorgulamalarınız gönderebilirsiniz.
- API'nızın kullanımını ve okunmasını kolaylařtırmak için dökümantasyon oluřturabilirsiniz.
- API'lerinizin iřleyiř durumunu ve performansını kontrol edebilirsiniz.
- Takım arkadaşlarınız ile birlikte workspace dedięimiz alıřma ortamında birlikte API hazırlayıp, kullanabilirsiniz.
- Bireysel olarak ücretsizdir. Sınırsız takım alıřması ve kaynak paylařımı gibi özellikleri istiyorsanız bunlar ücretlidir.
- Ücretsiz hesaplar için bazı özelliklere ulařımda limit vardır.





## Request URI Satırı

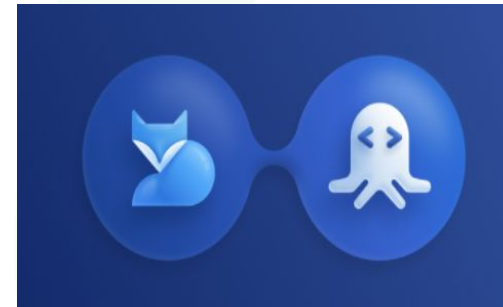
# Response Window





# POSTMAN

## POSTMAN alternatifleri



Hoppscotch



# SWAGGER DOKÜMANI

- Swagger , REST API'lerini tasarlamana, oluşturmana, belgelemenize ve kullanmamıza yardımcı olabilecek; OpenAPI Spesifikasyonu etrafında oluşturulmuş açık kaynaklı araçtır.
- Swagger'ın önemli bir amacı RestApi ler için bir arayüz sağlamaktır. Bu, insanların kaynak koda erişmeden RestApi lerin özelliklerini görmesine, incelemesine ve anlamasına olanak sağlar.

<https://app.swaggerhub.com/apis/MuratTANC/murat/1.0.0>



# API İSİMLENDİRİLMESİ

1) İsimlendirme anlamlı ve sade olmalı. API'de; sonradan isim değiştirmek, ona bağlı çalışan bir çok uygulamayı etkileyecektir.

<https://www.techproeducation.com/createCustomer> -

<https://www.techproeducation.com/customers> +

2) Hiyerarşi için / (slash) kullanılmalıdır.

<https://www.gmibank.com/api/tp-customers/114351>

<https://www.gmibank.com/api/tp-customers/114351/address>

SS

3) İsimlendirme iki kelime olduğu zaman araya – işareti konulur.

/api/tp-employees/{id}

/api/tp-customers/accounts/{id}

GET	/api/tp-customers	getAllTPCustomers	✓↵
POST	/api/tp-customers	createTPCustomer	✓↵
PUT	/api/tp-customers	updateTPCustomer	✓↵
GET	/api/tp-customers/accounts/{id}	getTPAccounts	✓↵
POST	/api/tp-customers/transfer	getMakeTransfer	✓↵
GET	/api/tp-customers/{id}	getTPCustomer	✓↵
DELETE	/api/tp-customers/{id}	deleteTPCustomer	✓↵



# API İSİMLENDİRİLMESİ

4) Versiyonlar “v1” şeklinde gösterilebilir. Yeni bir versiyon oluşturulduğunda “v1.1” veya “v2” şeklinde gösterilebilir. (Versiyon: API versiyon)

/api/v1/tp-customers/accounts/{id}  
/api/v1.1/tp-customers/accounts/{id}  
/api/v2/tp-customers/accounts/{id}

5) API dökümantasyonu mutlaka olmalı

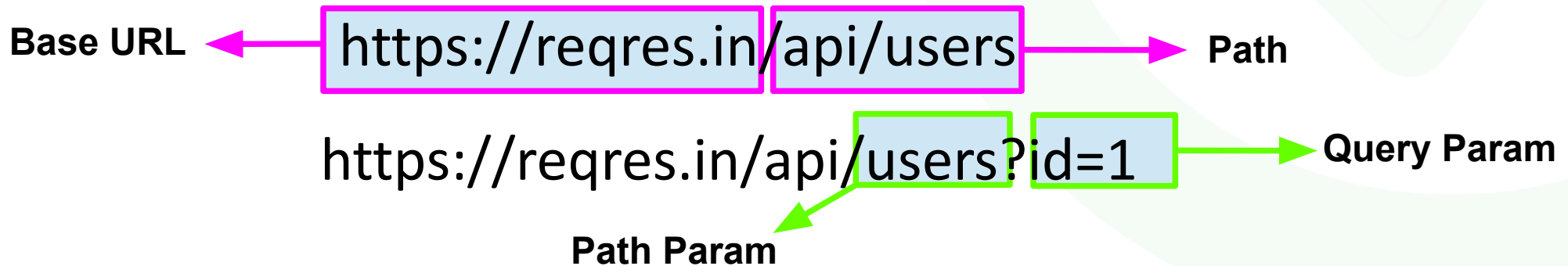






# Path Param, Query Param

- Bir kaynağı tanımlamak veya daha ayrıntılı nesneleri üzerinde hareket etmek istiyorsanız Path Param kullanmalısınız. Ancak öğeleri sıralamak veya filtrelemek istiyorsanız, Query Parametresi kullanmalısınız. Query parametreleri, kaynakları daha iyi bir şekilde tanımlamaya yardımcı olan benzersiz özelliklere sahiptir.
- Query parametreleri URL'de "?" işaretinin sağ tarafında görünürken, Path parametreleri soru işareti işaretinden önce gelir.
- URL'nin bir parçası oldukları için Path parametrelerindeki değerleri atlayamazsınız.
- Query parametreleri key-value şeklinde kullanılır.





# Path Param, Query Param

https://techproeducation.com/users

Base URL

Path

https://techproeducation.com/users?id=1

Path Param

Query Param

https://techproeducation.com/users?gender=female

https://techproeducation.com/users?gender=female&sort=ASC

https://techproeducation.com/users?gender=female&sort=ASC&page=1



<https://reqres.in/>



<https://restful-booker.herokuapp.com/booking>