Application Programming Interface

API

(Uygulama Programlama Arayüzü)

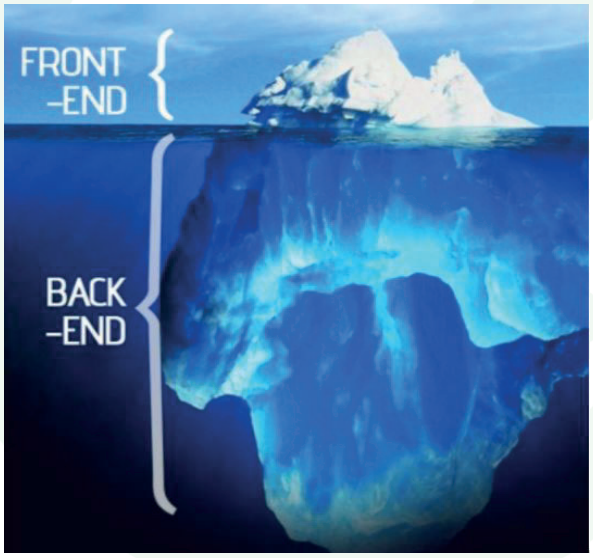
28/02/2022

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 Altenatifleri
- 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 göründüğü ve nasıl hissettirdiği ile ilgili değildir. Tasarım bir şeyin nasıl çalıştığıyla da alakalıdır.''*

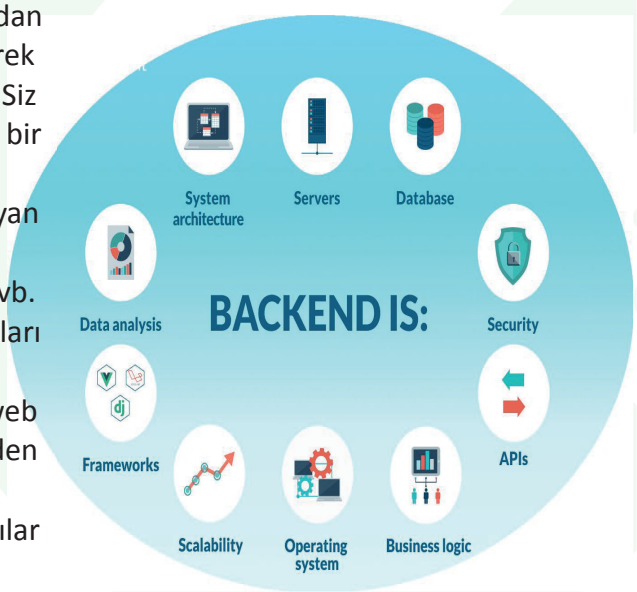


FRONTEND

- 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.

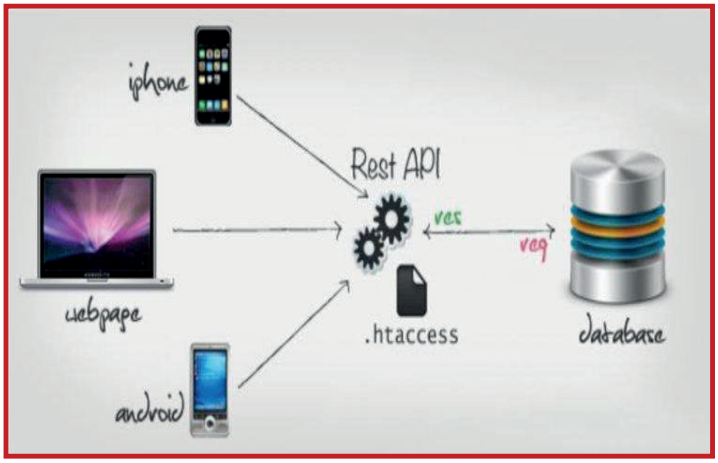
BACKEND

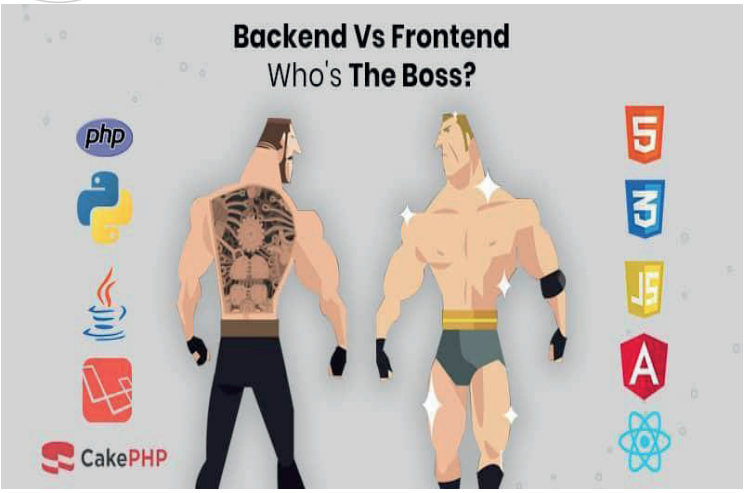
- Backend genellikle bir sunucu, bir uygulama ve bir veri tabanından oluşur. Bir havayolu veya otobüs firmasının, web sitesine girerek bilet aldığınızda; frontend ile etkileşime girmiş olursunuz. Siz bilgilerinizi web sitesine girdiğinizde, uygulama bu bilgiyi alır ve bir sunucu üzerinde kurulmuş olan veri tabanına depolar. Pegasus
- Sunucu, veri tabanı ve uygulamanın birlikte çalışmasını sağlayan kısıma 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?

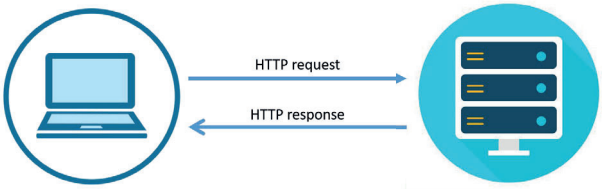
- **API (Aplication Programming Interface), bir yazılımın başka bir yazılımda; yalnızca belirli bölümlerine erişmesine ve kullanmasına izin veren bir yapıdır.**
- Bir uygulamanın başka bir uygulama ile iletişim kurmasına izin veren protokol kümesidir.
- API; web uygulaması, işletim sistemi, veritabanı, 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



Client

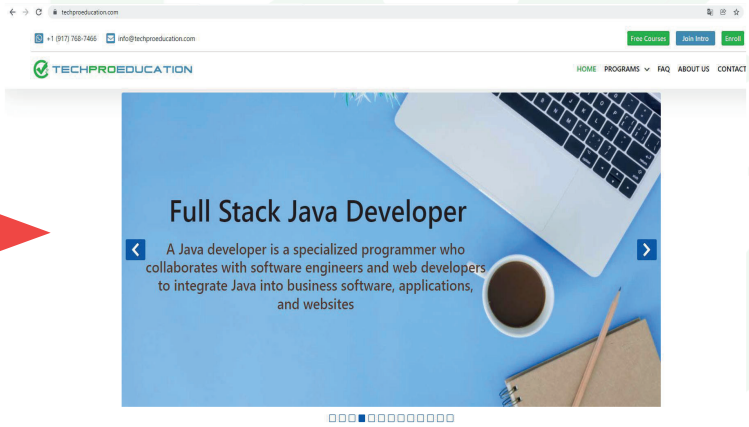
Server

Request: İstek, Talep

Response: Yanıt, Cevap

https://www.techproeducation.com/

UI



API NASIL ÇALIŞIR?

Client

1) Request to API

API Garson

2) Request to Server

Musteri Corba, Pizza, Kola istiyor

Database Server

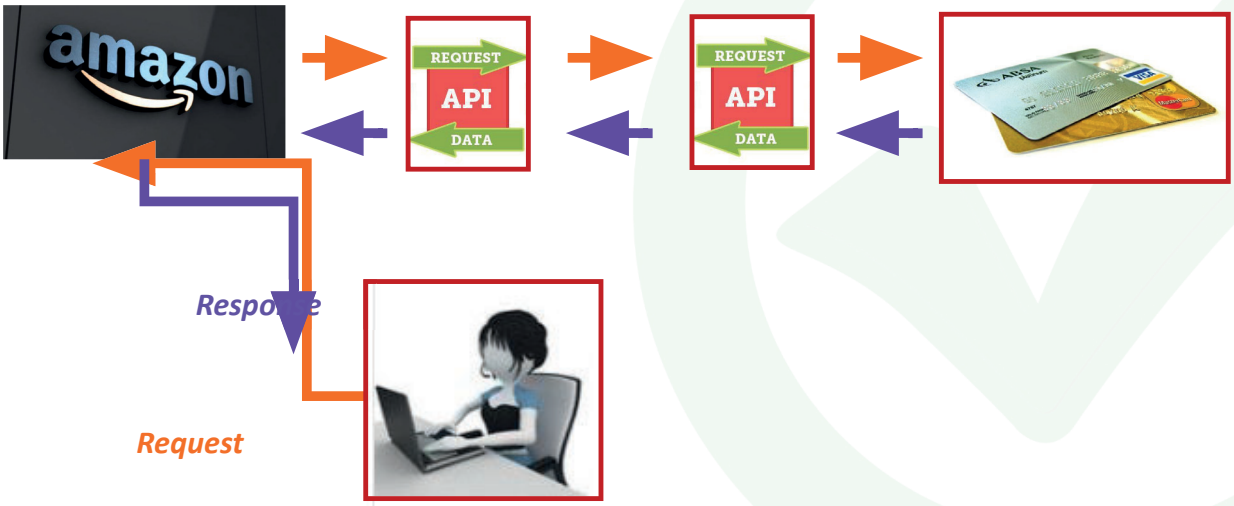
Siparis hazir

3) Response from Server

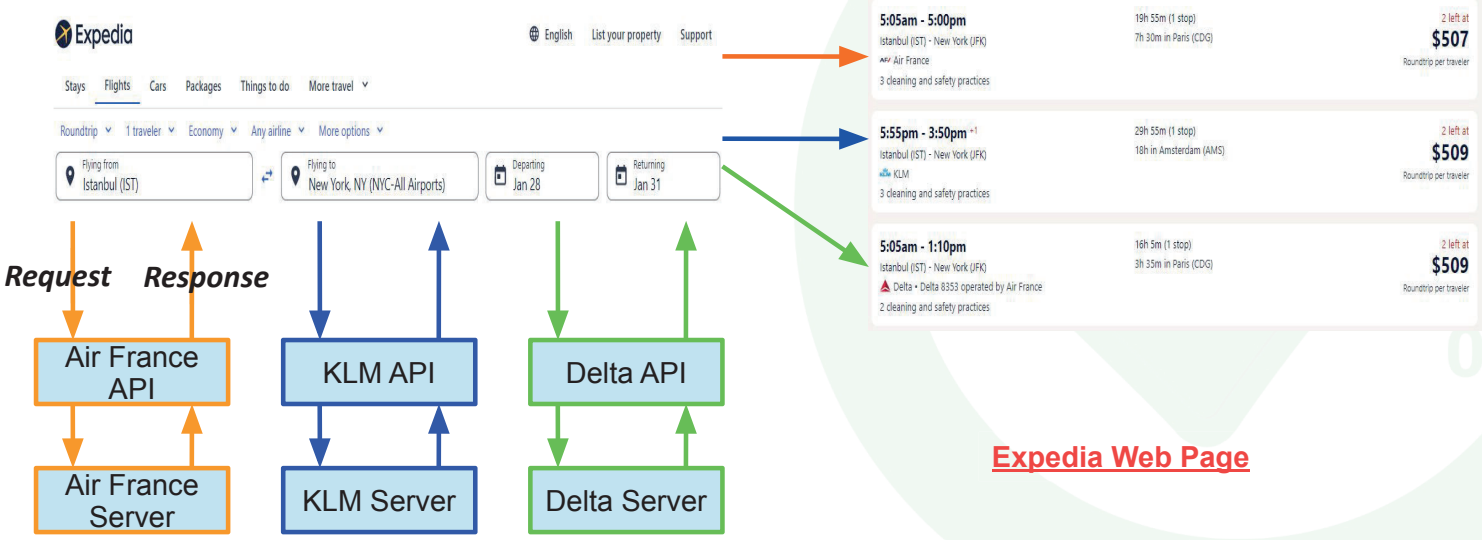
Sipariş iniz

4) Response to client

API NASIL ÇALIŞIR?

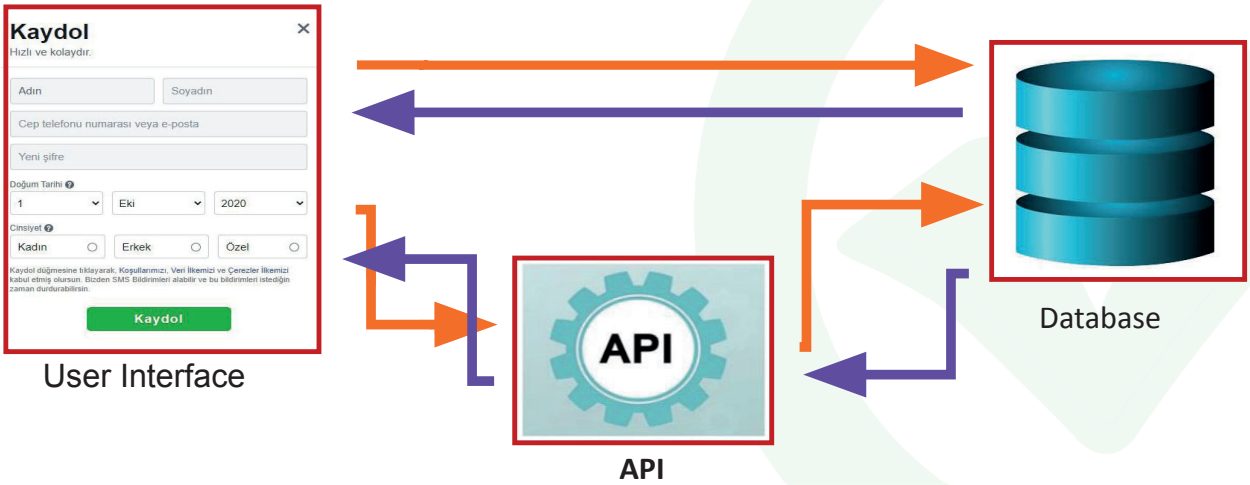


API NASIL ÇALIŞIR?



Expedia Web Page

API NASIL ÇALIŞIR?



API testi uygulamanin görüntüsü ile ilgilenmez, uygulamanın fonksiyon, performans ve güvenirlik açısından beklentilere uygun calışıp 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 iletisim sağlayabilir.**
- Örnegin; Expedia, KLM Airlines DataBase'ine ulaşmak için internet kullanır (Web Service), bilgisayarimizdaki Microsoft Word gibi uygulamalar ise farkli uygulamalarla iletisim 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 arasindaki tum iletisim 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.

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.

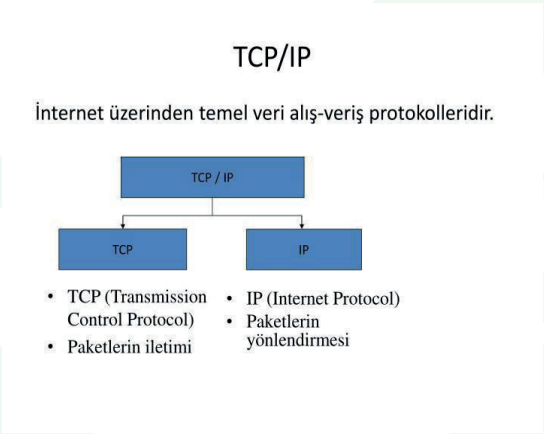
HTTP Status Codes

TCP (Transmission Control Protocol)

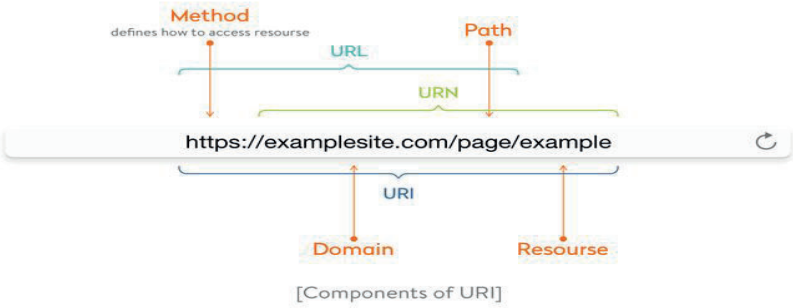
- TCP (Transmission Control Protocol) bilgisayarlar arasındaki iletişimin, küçük paketler hâlinde ve kayıpsız olarak gerçekleştirilmesini sağlayan bir protokoldür.
- Aslında TCP (Transmission Control Protocol) protokolünün en önemli özelliği; kimlik doğrulaması yapması ve veriyi karşı tarafa gönderirken 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.

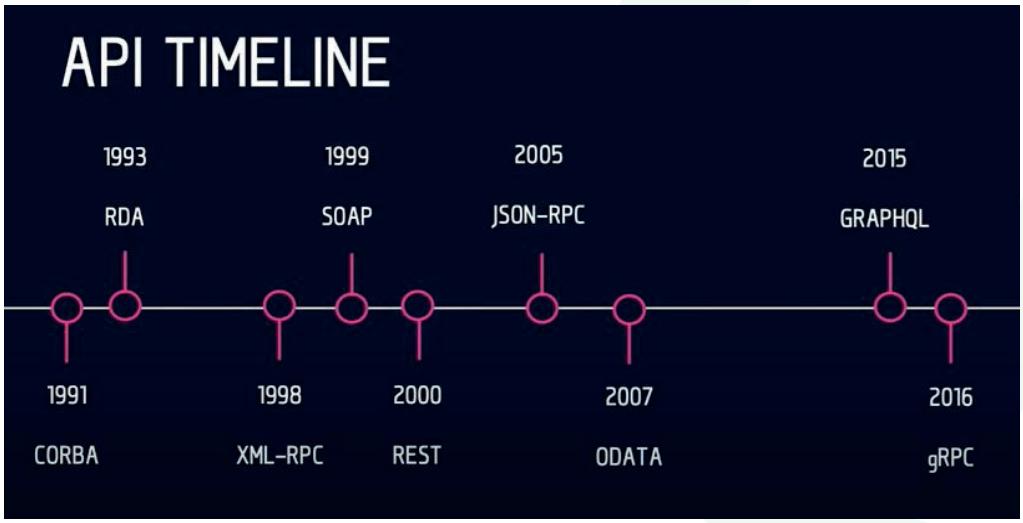


END POINT



- **End point kaynağa nasil erisebileceğimizi gosteren URI (Uniform Resource Identifiers)'lara denir.**
- Bir API olusturdugunuzda kullanicilarin ulasabilmesi icin bu endpoint'i ve kullanilabilecek Http method'larini kullanicilara bildirmemiz gerekir.
- URI gunluk hayatta kullandigimiz URL (Uniform Resource Locator)'a benzer.
- URL kullanicinin gorebilecegi ve etkilesimde bulunacagi bir web sayfasini ifade ederken URI o sayfanin icerdigi bilgiyi ifade eder.

API ARCHITECTURAL STYLES



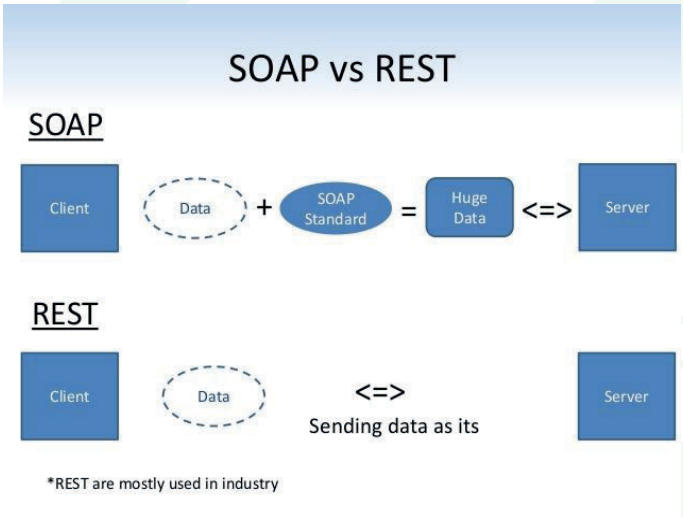
API ARCHITECTURAL STYLES

	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 yorumlayacağının tanımlanması gerekir. Bu web servis tanımlaması WSDL (Web Service Description Language- Web Servisleri Tanımlama Dili)

```
<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

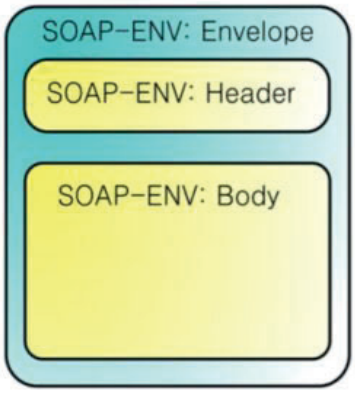
SOAP (Simple Object Access Protocol)

Bir SOAP mesajı 3 şeyi içerir.

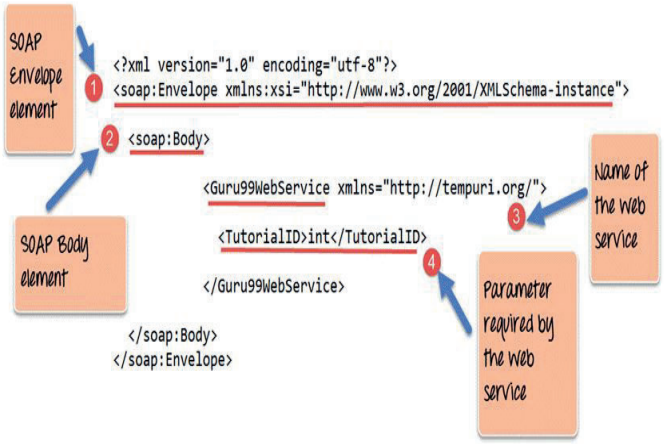
Envelope (Zarf): Tüm mesaj içeriğinin içinde olduğu kısımdır. SOAP mesajları XML formatında oluştundan bir root elemanı olmalıdır. Envelope, SOAP mesajının root etiketidir de denebilir. Envelope'un içinde Header ve Body gibi kısımlar da bulunur.

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.

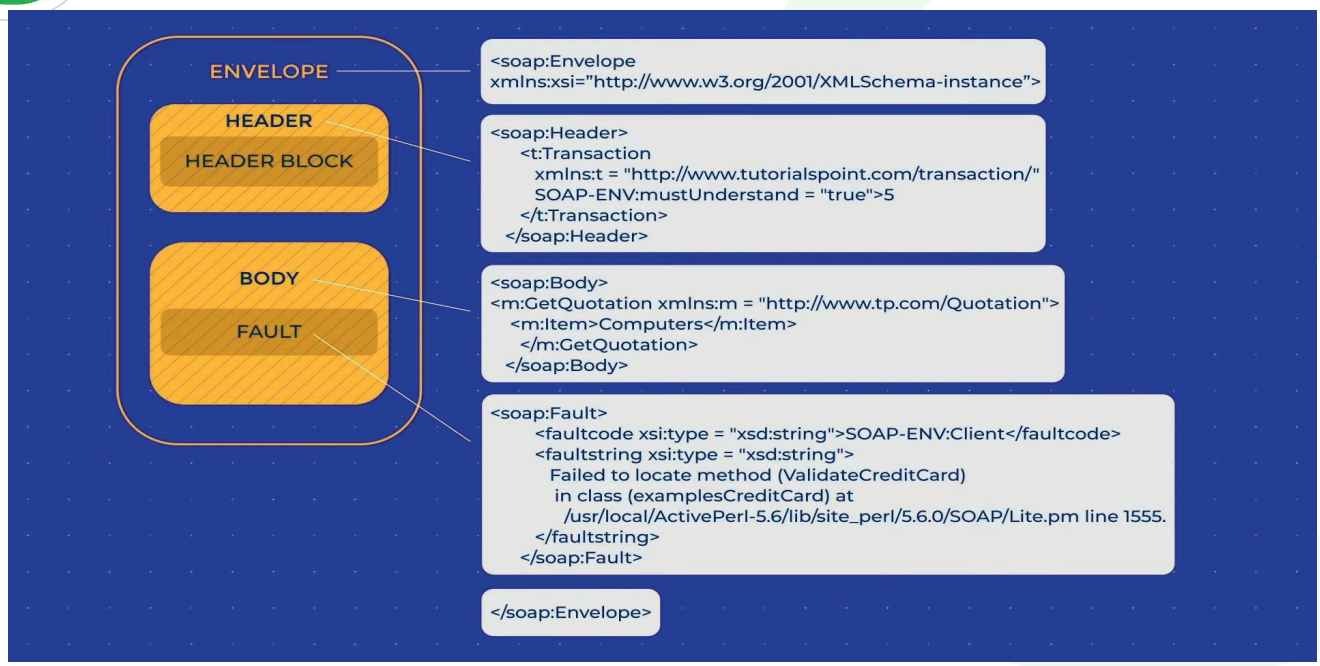


SOAP (Simple Object Access Protocol)




```
<?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)

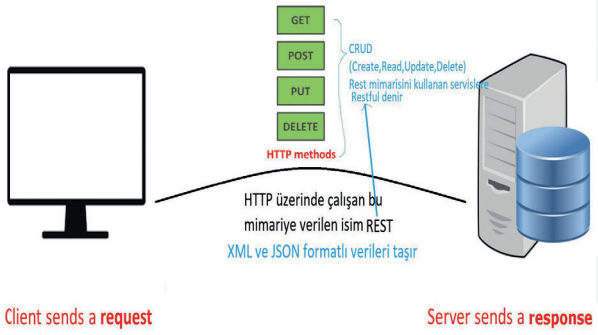


REST (REpresentational State Transfer)

- **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 çağıracak URI'ler ile web servise HTTP protokolüyle istek yapılır. Böylece işlemler tamamen HTTP metodları üzerinden yapılır.**

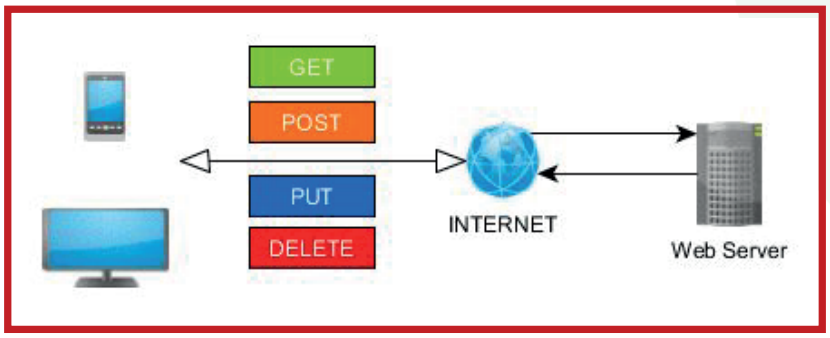
- Örneğin, bir web servisin metodunu SOAP ile "getProductName" şeklinde çağırırken REST ile "/products/name/1" URI'si ile çağırabiliriz. Ayrıca RESTin döndürdüğü veri tipinin de XML olması zorunlu değildir JSON (Java Script Object Notation) , XML, TXT, HTML gibi istenen veri türünde değer döndürülebilir.

- REST standartlarına uygun yazılan web servislerine RESTful servisler denir.



REST (REpresentational State Transfer)

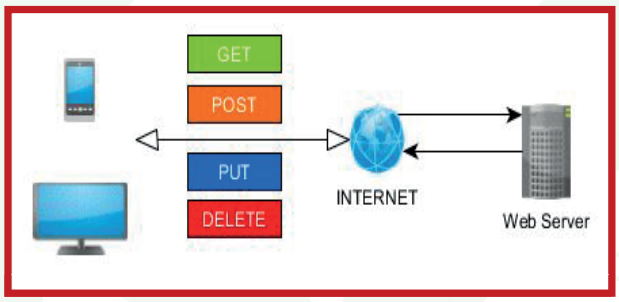
REST API, SOAP ve WSDL tabanlı web servislerinin daha basite indirgenmiş hali olarak nitelendirilebilir. Bu amaçla; internet tarayıcısı sayfa işlemlerinin bir parçası olan HTTP protokolünü (GET, POST, PUT DELETE gibi talep tipleri) kullanır. Böylelikle REST API geliştiricilere yaygın, pratik ve oldukça 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


```
{
    "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": [
        {
            "id": 2333,
            "description": "musteri omer hesap3",
            "balance": 69700,
            "accountType": "CREDIT_CARD",
            "accountStatusType": "ACTIVE",
            "createDate": "2020-11-06T23:00:00Z",
            "closedDate": "2024-11-06T23:00:00Z",
            "employee": null,
            "accountlogs": null
        },
        {
            "id": 107250,
            "description": "New Account_6thGenQA_01",
            "balance": 11190,
            "accountType": "CHECKING",
            "accountStatusType": "ACTIVE",
            "createDate": "2021-11-24T23:00:00Z",
            "closedDate": "2022-11-24T23:00:00Z",
            "employee": null,
            "accountlogs": null
        }
    ]
}
```


SOAP, REST

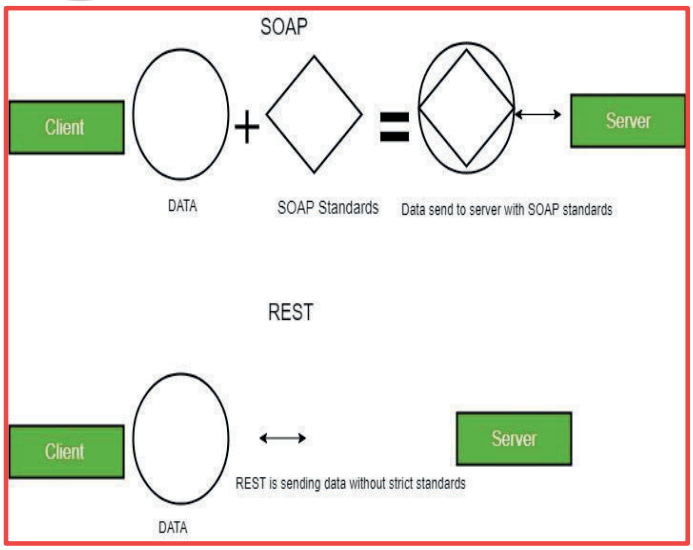
SOAP ve REST karşılaştırması yapmak ve her iki web servisininin artı ve eksilerini değerlendirmek gerekirse;

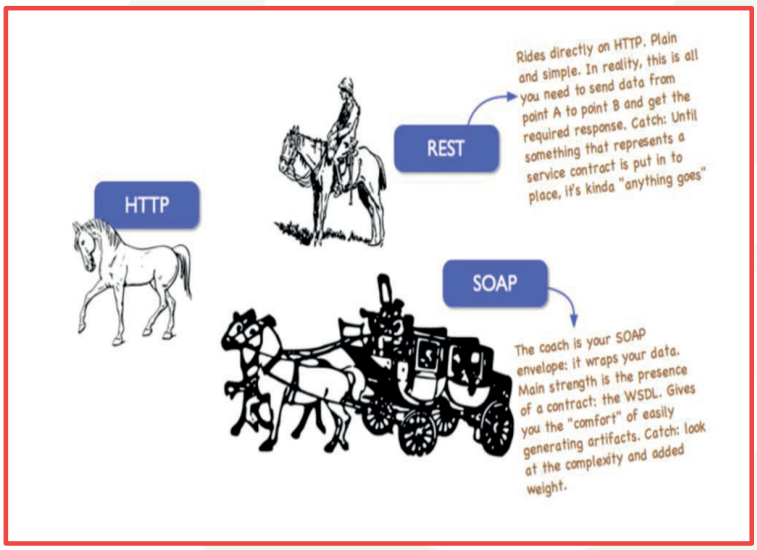
- REST ile JSON, XML, HTML, TEXT ile çalışmaktadır. SOAP ile sadece XML formatında çalışabilir. JSON ile daha ufak boyutlardaki verilerle işlemler kolayca gerçekleşebilir.
- SOAP, isteklerini oluşturmak için W3C standartı olan bir WSDL dili kullanmak zorundadır. Rest'te buna gerek yoktur. İstekler URI 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

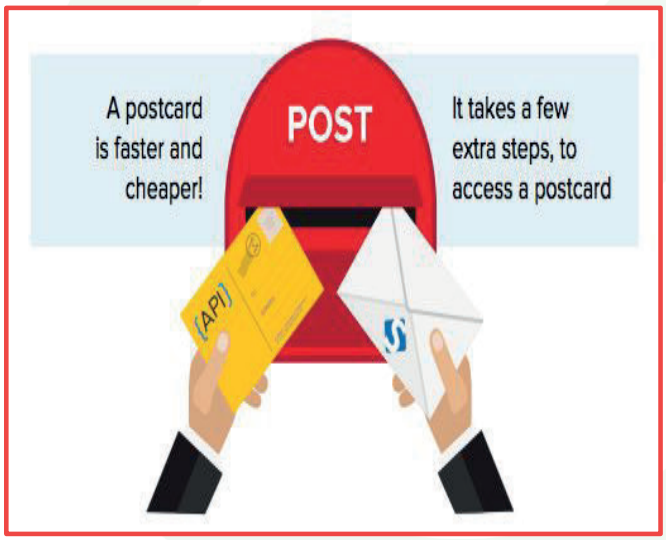
SOAP	REST
SOAP bir protokoldür	HTTP protokolünü kullanan bir mimaridir
WDSL ile tasarlama yapmak gerektiğinden kullanması daha zor	HTTP methodları ile tasarlandığı için kullanması daha kolay
Yalnızca XML format kullanır.	XML , JSON , HTML , TXT format kullanır
Önbelliği okuyamaz (can not be cached)	Önbelliği 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





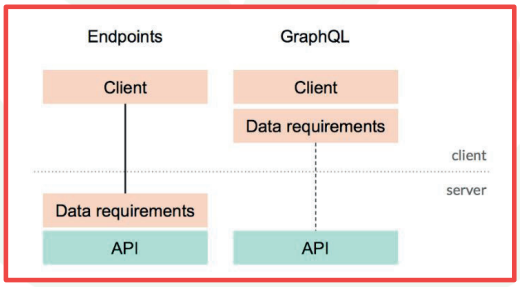
SOAP, REST



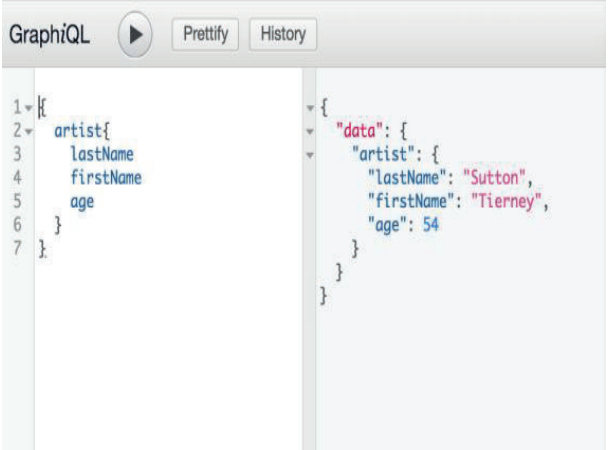
GraphQL

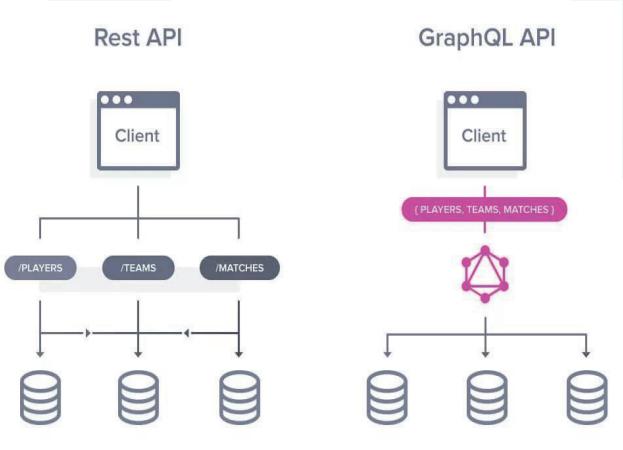
- GraphQL, Facebook tarafından geliştirilmiş; verimli, etkili ve esnek bir alternatif sunan yeni bir API standartıdır. Açık kaynaklı olarak geliştirilen GraphQL'i, birçok şirket aktif olarak kullanmaktadır.
- GraphQL, API tasarlamak ve kullanmak için bir yol sunmaktadır. İstemcinin ihtiyaç duyduğu verilerin 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



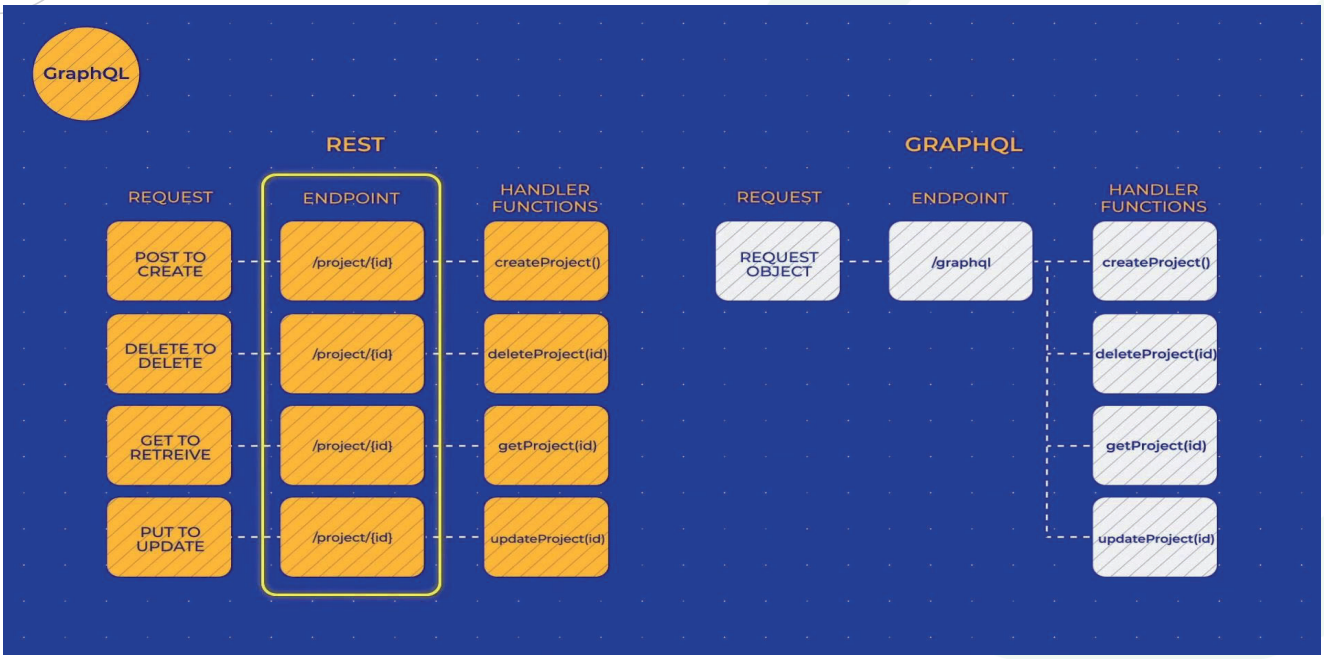
GraphQL





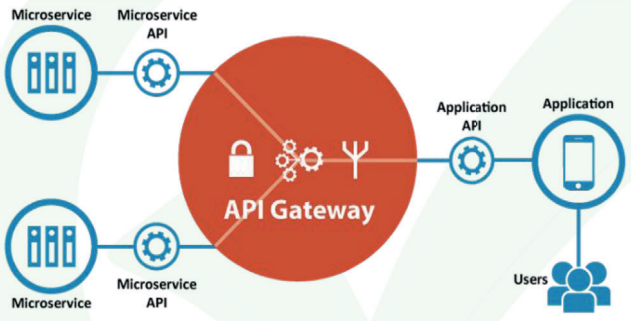
GraphQL Web Sitesi

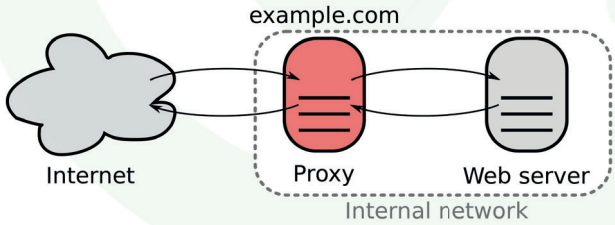
GraphQL



GATEWAY

- **API Gateway, istemcile 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.





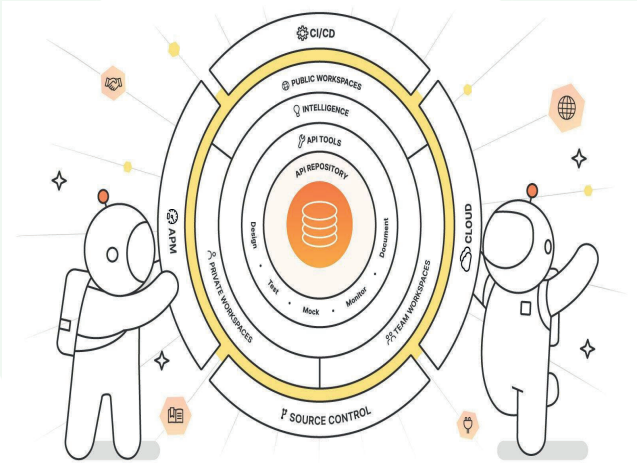
POSTMAN

- Postman, backend API'lerini test emek 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.

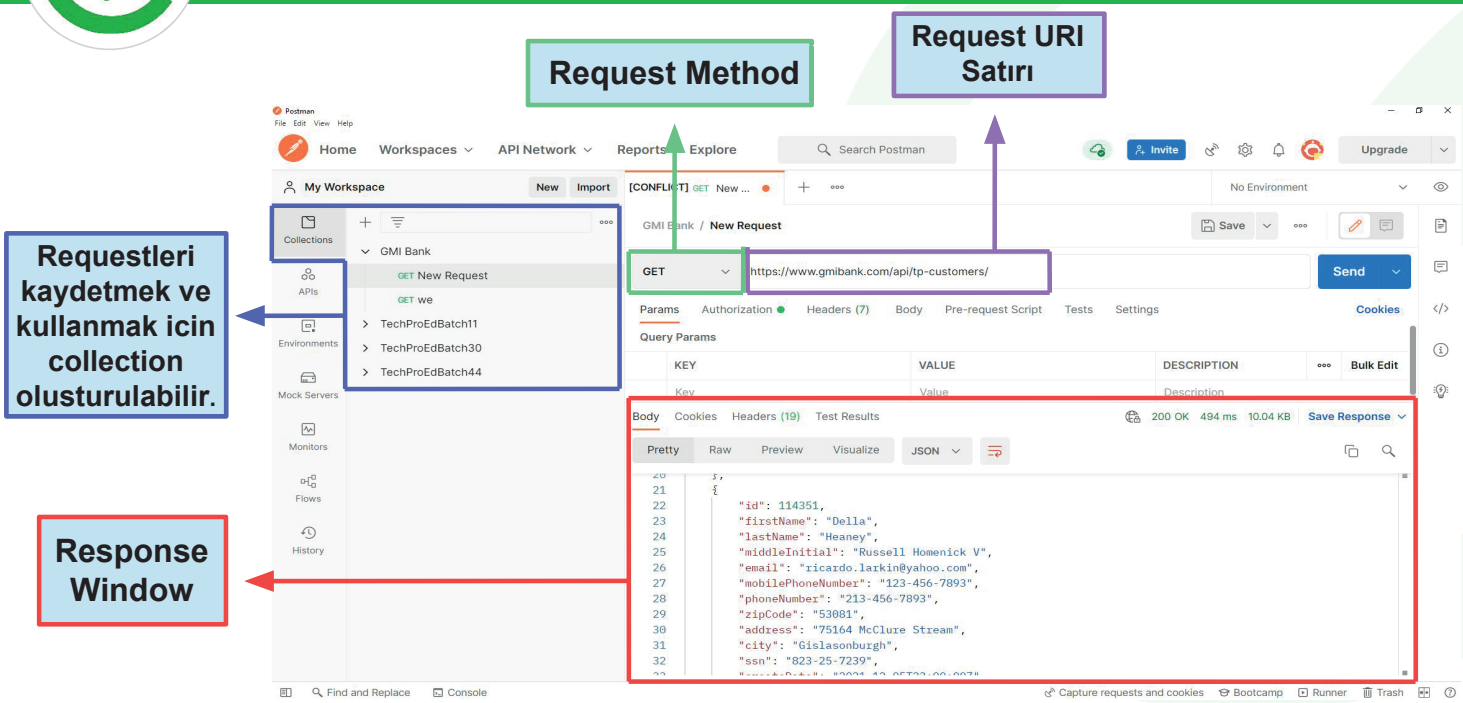
POSTMAN

POSTMAN

- Postman, API geliştirmek için bir işbirliği platformudur.
- Postman'ı kendi başınıza veya takım arkadaşlarınız ile birlikte API'ları tasarlamak, geliştirmek, oluşturmak ve test etmek için kullanabilirsiniz.
- Rest, SOAP ve GraphQL sorgulamalarınız gönderebilirsiniz.
- API'ınızı kullanımını ve okunmasını kolaylaştırmak için dökümantasyon olusturabilirsiniz.
- API'lerinizin işleyiş durumunu ve performansını kontrol edebilirisiniz.
- Takım arkadaşlarınız ile birlikte workspace dediğimiz çalışma ortamında birlikte API hazırlayıp, kullanabilirsiniz.
- Bireysel olarak ücretsizdir. Sınırsız takım çalışması ve kaynak paylaşımı gibi özellikleri istiyorsanız bunlar ücretlidir.
- Ücretsiz hesaplar için bazı özelliklere ulaşımda limit vardır.



POSTMAN



POSTMAN

POSTMAN alternatifleri

Testfully

Insomnia by Kong

Thunder Client

Firecamp

Test Mace

Nightingale

Hoppscotch

SWAGGER DOKÜMANI

- **Swagger , REST API'lerini tasarlamanıza, oluşturmanıza, 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.**

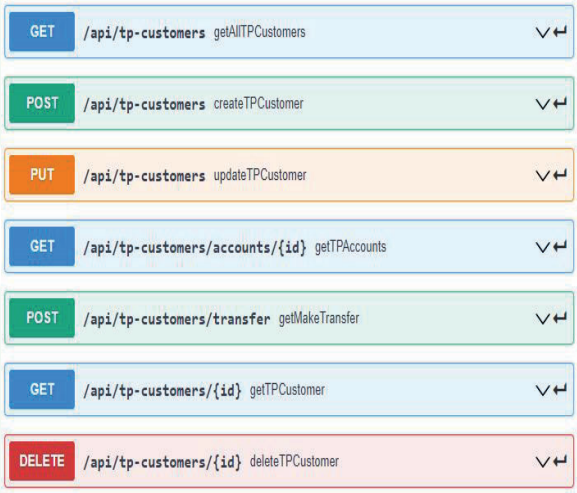
https://app.swaggerhub.com/apis/MuratTANC/murat/1.0.0

API İSİMLENDİRİLMESİ

1) İsimlendirme anlamlı ve sade olmalı. API'de; sonradan isim değiştirmek, ona bağlı çalışan bir çok uygulamayı etkileyecektir.
https://www.techproeducation.com/createCostomer –
https://www.techproeducation.com/costomers +

2) Hiyerarşi için / (slash) kullanılmalıdır.
https://www.gmibank.com/api/tp-customers/114351
https://www.gmibank.com/api/tp-customers/114351/address

3) İsimlendirme iki kelime olduğu zaman araya – işareti konulur.
/api/tp-employees/{id}
/api/tp-customers/accounts/{id}



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

Swagger

POSTMAN

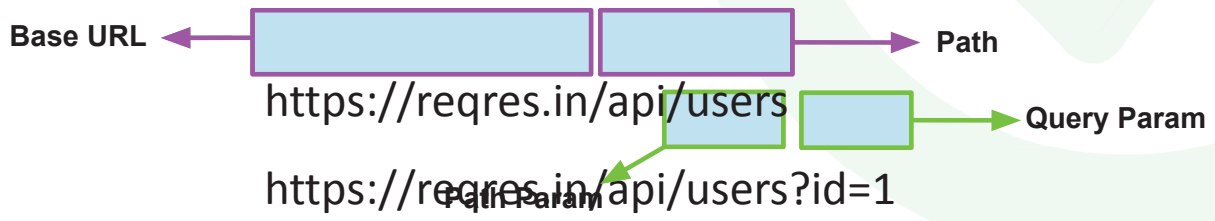
Path Param, Query Param

Bir kaynağı tanımlamak veya daha ayrıntılı nesneleri üzerinde hareket etmek istiyorsaniz Path Param kullanmalısınız. Ancak öğeleri sıralamak veya filtrelemek istiyorsanız, Query Parametresi kullanmalısınız. Query parametreleri, kaynakları daha iyi bir şekilde tanımlamaya yardımcı olur.

Query parametreleri URL'de "?" İşaretinin sağ tarafında görünürken, Path parametreleri soru işareti işaretinden önce gelir.

URL'nin bir parçası oldukları için Path parametrelerindeki değerleri atlayamazsınız.

Query parametreleri key-value seklinde kullanılır.



Path Param, Query Param

https://techproeducation.com/users

Base URL

Path

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

Path Param

Query Param

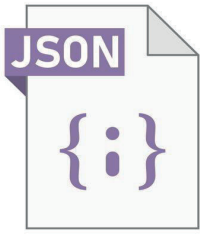
https://techproeducation.com/users?gender=female

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

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

JSON (JavaScript Object Notation)

- JSON formatı 2000'lerin başında Douglas Crockford tarafından belirlenmiştir. Aralık 2005'te Yahoo! web hizmetlerinden bazılarını JSON biçiminde sunmaya başlamıştır.
- JSON, 2013 yılında bir ECMA (European Computer Manufacturers Association, Avrupa Bilgisayar Üreticileri Birliği) standardı haline gelmiştir
- JSON; bütün programlama dilleri arasında, yapılandırılmış veri değişimini kolaylaştıran bir metin biçimidir.
- Sistemler arası veri aktarımı ile çalışan teknolojilerin çoğu JSON formatını destekler.



JSON SYNTAX

- Datalar name-value yapısındadır.
- Name-Value yapısı arasında : olur.
- JSON isimleri " " çift tırnak içerinde olur.
- Datalar , virgül ile birbirilerinden ayrılırlar
- Objeler { } (curly braces) içerisindedirler.
- Arrayler [ ] (square brackets) içindedir.

```
{ "Student" : [
  {"firstName" : "Ali" , "lastName" : "Can"},
        {"firstName" : "Sena" ,  "lastName" : "Bal"},
            {"firstName" : "Burak" , "lastName" : "Cengiz"}]
}
```

```
{
    "bookingid": 34,
    "booking": {
        "firstname": "Ali",
        "lastname": "CAN",
        "totalprice": 111,
        "depositpaid": true,
        "bookingdates": {
            "checkin": "2022-01-01",
            "checkout": "2022-01-01"
        },
        "additionalneeds": "Breakfast, Dinner"
    }
}
```

https://reqres.in/

https://reqres.in/api/users
Kayıtlı dataları gösterir.

https://reqres.in/api/users/3
3. kayıtlı datayı gösterir.

https://reqres.in/api/users?page=1
1. sayfadaki kullanıcıları gösterir.

https://reqres.in/api/users?page=2
2. sayfadaki kullanıcıları gösterir.


```
{
    "page": 1,
    "per_page": 6,
    "total": 12,
    "total_pages": 2,
    "data": [
        {
            "id": 1,
            "email": "george.bluth@reqres.in",
            "first_name": "George",
            "last_name": "Bluth",
            "avatar": "https://reqres.in/img/faces/1-image.jpg"
        },
        {
            "id": 2,
            "email": "janet.weaver@reqres.in",
            "first_name": "Janet",
            "last_name": "Weaver",
            "avatar": "https://reqres.in/img/faces/2-image.jpg"
        },
        {
            "id": 3,
            "email": "emma.wong@reqres.in",
            "first_name": "Emma",
            "last_name": "Wong",
            "avatar": "https://reqres.in/img/faces/3-image.jpg"
        },
```


https://restful-booker.herokuapp.com/

https://restful-booker.herokuapp.com/booking
Kayıtlı tüm datayı gösterir.

https://restful-booker.herokuapp.com/booking/10
Kayıtlı 10. datayı gösterir.


```
[
    {
        "bookingid": 45
    },
    {
        "bookingid": 10
    },
    {
        "bookingid": 6
    },
    {
        "bookingid": 22
    },
    {
        "bookingid": 44
    },
    {
        "bookingid": 40
    },
    {
```

http://dummy.restapiexample.com/

http://dummy.restapiexample.com/api/v1/employees
Kayıtlı işcilerin bilgilerini getirir.

http://dummy.restapiexample.com/api/v1/employee/1
1 numaralı çalışanın bilgilerini getirir.


```
{
    "status": "success",
    "data": [
        {
            "id": 1,
            "employee_name": "Tiger Nixon",
            "employee_salary": 320800,
            "employee_age": 61,
            "profile_image": ""
        },
        {
            "id": 2,
            "employee_name": "Garrett Winters",
            "employee_salary": 170750,
            "employee_age": 63,
            "profile_image": ""
        },
        {
            "id": 3,
            "employee_name": "Ashton Cox",
            "employee_salary": 86000,
            "employee_age": 66,
            "profile_image": ""
        },
```


REST ASSURED

- **Rest Assured, REST API'lerini test etmek ve doğrulamak için kullanılan bir Open Source (Açık Kaynak) bir Java kütüphanesidir.**
- REST Assured, HTTP Builder üzerine kurulu REST tabanlı hizmetlerin test edilmesini basitleştirmek için bir Java DSL (Domain Specific Language: Özel bir uygulama alanı için kullanılan dil)'dir.
- 2010 yılınında Johan Haleby tarafından literatüre kazandırılmış.
- Maven ile çok iyi entegre olur.
- XML ve JSON tabanlı web servislerini test etmek için kullanılabilir.

REST-assured

REST ASSURED

- GET, POST, PUT, PATCH, DELETE, OPTIONS,HEAD ... isteklerini destekler ve bu isteklerin yanıtını doğrulamak için kullanılabilir.
- Junit ve TestNG gibi test frameworkleri ile entegre edilebilir.
- Apache Http Client karşılaştırıldığında daha az kodlama gerektirir.
- API'ler için farklı kimlik doğrulama mekanizmaları için çok iyi destek.Kodu okunabilir kılan ve temiz kodlamayı destekleyen verilen(), while(), then() gibi BDD anahtar sözcüklerini takip eder.
- JSON ve XML yanıtını ayrıştırmaya yardımcı olan JsonPath ve XmlPath'i destekler. Rest Assured varsayılan olarak her ikisini de birleştirir.

JSON PATH

- JsonPath, Json Format ile verilen bir dataya ulaşmak veya değiştirmek için kullanılır.
- JSONPath, JSON verileri arasında bize sorgu yapma fırsatının verir.
- $ isareti Json dokumanindaki tüm node'ları verir.
- Child bölümlere ulaşmak icin (.) kullanılabilir. store.book bize tüm kitaplari verir
- Belirli bir kitaba ulaşmak için; yapı array olduğundanindex kullanabiliriz. store.book[1] , Birden fazla kitaba ulasmak istersek virgule indexleri yazabiliriz. store.book[1,3]
- 2.kitabin price bilgisine ulasmak icin store.book[1].price kullanabiliriz
- Son kitaba ulasmak icin index olarak -1 kullanabiliriz.
- Tum kitaplarin yazarlarini listelemek icin store.book[*].author kullanabiliriz

https://jsonpath.herokuapp.com/


```
{
  "store" : {
    "book" : [
      {
        "category" : "reference",
        "author" : "Nigel Rees",
        "title" : "Sayings of the Century",
        "price" : 8.95
      },
      {
        "category" : "fiction",
        "author" : "Evelyn Waugh",
        "title" : "Sword of Honour",
        "price" : 12.99
      },
      {
        "category" : "fiction",
        "author" : "Herman Melville",
        "title" : "Moby Dick",
        "isbn" : "0-553-21311-3",
        "price" : 8.99
      },
      {
        "category" : "fiction",
        "author" : "J. R. R. Tolkien",
        "title" : "The Lord of the Rings",
        "isbn" : "0-395-19395-8",
        "price" : 22.99
      }
    ],
    "bicycle" : {
      "color" : "red",
      "price" : 19.95
    }
  },
  "expensive" : 10
}
```

JSON PATH

// https://restful-booker.herokuapp.com/booking url'ine
// accept type'i "application/json" olan GET request'i yolladigimda
// donecek cevap (response) icin
// 1) HTTP status kodunun 200
// 2) Content Type'in Json
// 3) Ve Status Line'in HTTP/1.1 200 OK
// oldugunu test edin.

API TEST

IntelliJ ile API testleri yapabilmek icin POM Xml'e rest-assured, junit ve json dependency'lerinin yüklenmesi gerekir,

given: Genellikle, request'e basmadan önce contentType, body vb. tüm ön koşulları bu bölüme yazıyoruz.

when: get(), post(), put(), patch() ve delete() gibi göndermek istediğimiz http request yöntemini içerir.

then: istek gönderildikten sonra yapmak istediğimiz tüm doğrulamaları yazıyoruz.

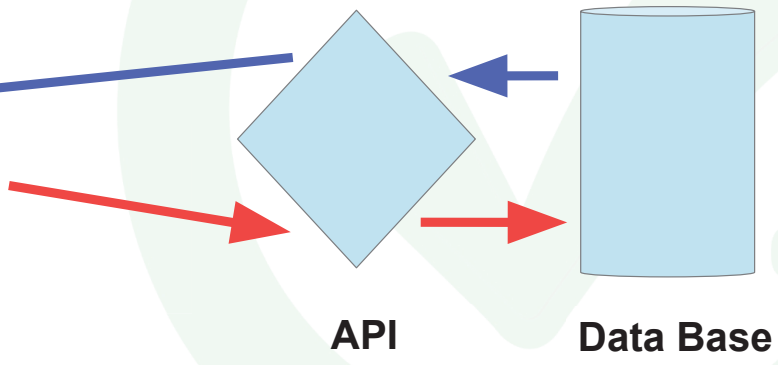
and: Çoklu işlem yapılacaksa kullanılır.

```
public class GetRequest {

    @Test
    public void test01(){
            given().contentType(ContentType.JSON).
            when().get( s: "https://restful-booker.herokuapp.com/booking/3").
            then().statusCode(200);
    }
}
```

API TEST

Response response =
given().when().get(url);



API TEST

GetRequest03:
https://restful-booker.herokuapp.com/booking/7 url'ine
accept type'i "application/json" olan GET request'i yolladigimda
gelen response'un
status kodunun 200
ve content type'inin "application/json"
ve firstname'in "Sally"
ve lastname'in "Ericsson"
ve checkin date'in 2018-10-07"
ve checkout date'in 2020-09-30 oldugunu test edin

body("key", Matchers.*equalTo*("Value")) : key olarak verilen değisken'in değerinin value'ya eşit olup olmadığını kontrol eder.

API TEST

GetRequest04:
https://restful-booker.herokuapp.com/booking/5 url'ine
accept type'i "application/json" olan GET request'i yolladigimda
gelen response'un
status kodunun 200
ve content type'inin "application/json"
ve firstname'in “Jim"
ve totalprice’in 600
ve checkin date'in 2015-06-12"oldugunu test edin

API TEST

GetRequest05:
http://dummy.restapiexample.com/api/v1/employees url'ine
GET request'i yolladigimda gelen response'un
status kodunun 200 ve content type'inin "application/json"
ve employees sayisinin 24
ve employee'lerden birinin "Ashton Cox"
ve gelen yaslar icinde 21, 61, ve 23 degerlerinden birinin oldugunu test edin.

body("data.id", Matchers.hasSize(" value ")) : key olarak verilen degisken’in sayisinin value’ya esit olup olmadigini kontrol eder.
body("data.employee_name",Matchers.hasItem(" value ")) : key olarak verilen degisken’in aldigi degerlerin icinde value var mi diye kontrol eder. Value birden fazla ise hasItems kullanip value’lari vigulle yanyana yazilabilir

API TEST

GetRequest06:

https://jsonplaceholder.typicode.com/todos/123 url'ine accept type'i "application/json" olan GET request'i yolladigimda gelen response'un

status kodunun 200

ve content type'inin "application/json"

ve Headers'daki "Server" in "cloudflare"

ve response body'deki "userId"'nin 7

ve "title" in "esse et quis iste est earum aut impedit"

ve "completed" bolumunun false oldugunu test edin