

# Api Testine Giriş

## ##Client & Internet & Server Nedir?

- Client: Server'ın bir servisine erişmeye çalışan cihazlar (Genelde kişisel bilgisayarlarımız)
- Internet: Client ile Server arasındaki bağlantı
- Server: Uygulamanın tutulduğu ve Client'lardan gelen isteklere cevap veren yer
- -

## ##Client/Server Architecture

- 1-Tier Architecture: File Server ile Computer aynı yerde çalışır.
- 2-Tier Architecture: Computer ile Database ayrı yerlerde direkt bağlılar
- 3-Tier Architecture: Bilgisayar ile Database Application Server (Business Logic Tier) aracılığıyla birbirine bağlanır

Katmanlar:

1. En Üst Katman: Presentation Layer(Önyüz, Veriyi serverdan alıp sunar)(HTML, JS, CSS)
2. Ara Katman: Application Layer(Business Logic ve Web Server'ın bulunduğu yer)(Java, .NET, C#, Python,C++)
3. En alt katman: Data Layer(Verinin bulunduğu yer)(MySQL, Oracle,PostgreSQL,SQL Server, MangoDB)

Web Uygulamaları Nasıl çalışır?:

**Presentation Layerdan Request gelir -> Application Layer Requesti alır ve Data layera gönderilir.-> Data Layer'dan veriye ulaşılır ve response oluşur bu**

**response -> Application Layer'a gider -> Tekrardan en üste çıkararak Presentation Layer'a ulaşır.**

## **##API(Application Programming Interface) nedir?**

- API iki uygulamanın kendileri arasında iletişim kurma biçimidir. Bu uygulamalar farklı platformlarda olabilir

2 tür API vardır ve ikiside web servisidir

1. Simple Object Access Protocol (SOAP): Kısaca eski bir API türü artık kullanılmıyor eskisi kadar
2. Representational State Transfer (REST): Bizim kullanacağımız API türü güncel ve yeni.

Web Service = API + Internet

Her Web Service bir API'dır ama her API Web Service değildir

Rest API Methods/Http Request

1. GET: Sunucudan veri okumak için kullanılır.
2. POST: Sunucuya yeni veri eklemek için kullanılır.
3. PUT: Var olan veriyi tamamen güncellemek için kullanılır.
4. DELETE: Var olan veriyi silmek için kullanılır.

Request Mesajı(get,post,put,patch,delete vb.) → API → Response Mesajı(Response Status code, Data vb.)

HTTPS = GÜVENLİKLİ HTTP

Örnek: <https://dbbrain.com/articles/articlename>

Burada "https://" Scheme

"dbbrain.com" Host

"/articles/articlename" Endpoint oluyor.

Payload: Payload, bir istekte (özellikle POST/PUT) sunucuya gönderilen asıl veri içeriğidir.

Örnek:

-- Request Payload (istek sırasında gönderilen veri) --

Http: POST /login

Json:

```
{  
  "username": "eren",  
  "password": "1234"  
}
```

-- Response Payload (cevapta dönen veri) --

Http: 200 OK

Json:

```
{  
  "userId": 12,  
  "name": "Eren",  
  "token": "abc123xyz"  
}
```

Request payload: Client → Server

Response payload: Server → Client

- -