

## WEB TEMELLERİ

### Eğitim Özeti

"Web Temelleri" eğitimi, internetin nasıl çalıştığına ve web teknolojilerinin temel prensiplerine giriş yapar. Bu kurs, World Wide Web'in ne olduğunu, internet ile web arasındaki farkları açıklar ve katılımcılara online dünyanın temel yapı taşlarından biri olan web'in işleyiş mekanizmasını anlama fırsatı sunar. İstemci ve sunucu modeli üzerinde durularak, web sayfalarının nasıl istemci (kullanıcı tarafı) tarafından talep edildiği ve sunucu tarafından nasıl sunulduğu detaylı bir şekilde işlenir.

Eğitim ayrıca, internet üzerindeki her cihazın ve kaynağın birbiriyle nasıl iletişim kurduğunu anlamak için kritik öneme sahip olan Domain Name System (DNS) ve Hypertext Transfer Protocol (HTTP) gibi kavramlara da derinlemesine dalış yapar. DNS'nin, insanların kolayca hatırlayabileceği domain isimlerini bilgisayarların anlayabileceği IP adreslerine çevirme işlevini nasıl üstlendiği ve HTTP'nin web sayfalarının istemci ile sunucu arasında nasıl transfer edildiğini yöneten bir protokol olduğu açıklanır.

### Giriş

#### World Wide Web (WWW)

World Wide Web, internet üzerinde çalışan ve hiper-metin dokümanları şeklinde organize edilen geniş bir bilgi sistemidir. Hiper-metin, dokümanların birbirine bağlantılar (linkler) aracılığıyla entegre edilmesine olanak tanıyan bir yapıdır. Bu bağlantılar, kullanıcıların bir dokümandan diğerine kolayca ve hızlı bir şekilde geçiş yapmalarını sağlar. Bir web sayfasında karşılaştığınız bir terim ya da konu hakkında daha fazla bilgi almak istediğinizde veya farklı bir sayfaya geçmek istediğinizde ilgili bağlantıya tıklayarak konuyla ilgili başka bir sayfaya yönlendirilirseniz buna iyi bir örnektir.

World Wide Web'in icadı, 1989 yılında İngiliz bilim insanı Tim Berners-Lee tarafından gerçekleştirilmiş bir devrimdir. Berners-Lee, o dönemde Avrupa

Nükleer Araştırma Merkezi (CERN) bünyesinde çalışmakta olan bir bilgisayar mühendisiydi. İcadın temel amacı, dünya çapındaki bilim insanları arasında otomatik bilgi paylaşımını sağlamak ve böylece deney ve çalışmalarının veri ve sonuçlarını birbirleriyle kolayca paylaşabilmelerine olanak tanımaktı.

CERN, 100'den fazla ülkeden 1700'den fazla bilim insanının toplandığı uluslararası bir araştırma topluluğudur. Bu bilim insanları, zamanlarının bir kısmını CERN'de, geri kalan zamanlarını ise kendi ülkelerindeki üniversitelerde ve ulusal laboratuvarlarda geçirmekteydiler. Bu nedenle, bilgi alışverişinde bulunabilmek için güvenilir iletişim araçlarına ihtiyaç duyulmaktaydı.

O dönemde internet ve hiper-metin teknolojileri mevcuttu; ancak, internetin bir dokümanı diğerine bağlamak veya paylaşmak için nasıl kullanılacağı konusunda kimse somut bir adım atmamıştı. Tim Berners-Lee, bilgisayarların birbirlerini anlamasını sağlayacak üç ana teknoloji üzerine odaklandı: **HTML (Hiper Metin İşaretleme Dili)**, **URL (Birleşik Kaynak Yer Belirleyici)** ve **HTTP (Hiper Metin Transfer Protokolü)**. Dolayısıyla, World Wide Web'in icadının arkasındaki temel hedef, mevcut bilgisayar teknolojilerini, veri ağlarını ve hiper-metni, kullanıcı dostu ve etkili bir küresel bilgi sistemi haline getirecek şekilde birleştirmektir.

### **Internet vs Web**

Internet ve World Wide Web (Web), sık sık birbiriyle karıştırılan ancak aslında birbirinden farklı kavramlardır. Bu iki terimin farklarını açıklamak, onların nasıl birlikte çalıştığını ve birbirlerini nasıl tamamladıklarını anlamamıza yardımcı olur.

Internet, dünya çapında milyonlarca bilgisayarın birbirine bağlandığı devasa bir ağıdır. 1960'ların sonlarına doğru, ABD Savunma Bakanlığı'nın ARPANET projesi ile başlayan bu ağ, farklı bilgisayar ağlarının birbirine bağlanmasıyla oluşur. Internet, e-posta, dosya transferi, anlık mesajlaşma ve çevrimiçi oyunlar gibi çeşitli hizmetleri ve protokolleri destekler. Dolayısıyla, Internet bir altyapıdır ve bu altyapı üzerinde çalışan bir dizi hizmet ve protokolü içerir.

Öte yandan, World Wide Web ya da kısaca Web, Internet üzerinde çalışan ve hiper-metin dokümanlarını kullanarak bilgi sunan bir bilgi paylaşım sistemidir. Tim Berners-Lee tarafından 1989 yılında icat edilen Web, kullanıcıların HTML diliyle hazırlanmış web sayfalarına erişimini sağlar. Bu sayfalar, URL'ler aracılığıyla birbirine bağlanır ve HTTP protokolü üzerinden iletilir. Web, kullanıcıların metin, görsel ve video gibi çeşitli medya türlerine erişebilmesini sağlayan zengin ve interaktif bir deneyim sunar. Kısacası, Internet, farklı bilgisayarların birbirine bağlandığı geniş kapsamlı bir ağıdır ve Web, bu ağ üzerinde çalışan ve kullanıcıların bilgiye erişimini kolaylaştıran bir hizmettir. Web, Internet'in sunduğu birçok hizmetten sadece biridir, ancak günümüzde öylesine yaygın kullanılmaktadır ki, bazen Internet ile eş anlamlı gibi algılanabilir. Ancak teknik olarak, Internet altyapısı; Web ise bu altyapı üzerinde çalışan uygulamalardan birini temsil eder

## **İstemci ve Sunucu Modeli**

Internet dünyasında, **istemci**(client) ve **sunucu**(server) terimleri sıkça karşımıza çıkar. Bu iki terim, internetin temel bir prensibi olan istemci-sunucu modelini tanımlar. Bu modelde, bir istemci bir sunucudan (web sitelerini barındıran bilgisayar) bilgi veya hizmet talep eder. Sunucu, bu talebi işler ve gerekli yanıtı istemciye geri gönderir. Bu etkileşim, web'in temelini oluşturur.

İstemci-sunucu modeli, merkezi bir sistemle tüm verilerin tek bir yerde toplanması nedeniyle maliyet açısından büyük bir avantaj sağlar ve kaynaklar yalnızca gerektiğinde kullanılır.

## Client-Server Model



## Domain Name System (DNS)

**Alan Adı Sistemi (DNS)**, kullanıcıları World Wide Web'deki web sitelerine, hizmetlere ve kaynaklara bağlamada temel bir rol oynayan İnternet altyapısının kritik bir bileşenidir. Esasen İnternetin "telefon rehberi"dir ve kullanıcı dostu alan adlarını ([www.example.com](http://www.example.com) gibi) bilgisayarların ve ağ cihazlarının İnternette birbirlerini bulmak için kullandıkları sayısal IP adreslerine (192.0.2.1 gibi) çevirir.

DNS'nin gelişimi, İnternetin isminin ARPANET olduğu İnternetin ilk günlerinde başlamıştır. 1980'lerin başında ARPANET, ana bilgisayar adlarını IP adresleriyle eşleyen "hosts.txt" adlı merkezi olarak yönetilen bir dosyayı kullanıma sundu. Bilgisayarlar, alan adlarına ait IP adreslerini öğrenebilmek için, bu dosyada sorgu yapıyordu. Ancak İnternet hızla büyüdüğü, bu yaklaşım yönetilemez hale geldi.

1983 yılında Paul Mockapetris ve Jon Postel, **RFC 882** ve **RFC 883** aracılığıyla bugün bildiğimiz DNS'yi tanıttı ve alan adı çözümlemesi için dağıtılmış ve hiyerarşik bir sistem sağladı. Bu yenilik, modern İnternetin temelini oluşturan ölçeklenebilir ve verimli DNS mimarisinin önünü açmıştır.

## Alan Adı (Domain)

Alan adı, internet üzerindeki belirli bir konumu veya kaynağı temsil eden, insan tarafından okunabilir bir etikettir. Alan adları, noktalarla (noktalar) ayrılmış seviyelerle bir hiyerarşi olarak yapılandırılmıştır.

Örneğin, [www.example.com](http://www.example.com) alan adı üç bölümden oluşur:

- **www** - alt alan adı (subdomain)
- **example** - ikinci düzey alan adı (second-level domain)
- **com** - üst düzey alan adı (top-level domain)

## Üst Düzey Alan Adı (Top-Level Domain, TLD):

TLD, (Üst Düzey Alan Adı) internet alan adları hiyerarşisinin en üstünde yer alır. Bir alan adının (domain) son kısmıdır. Genellikle **alan adı uzantısı** da denir. TLD'ler, web sitelerinin türünü, amacını veya coğrafi konumunu belirtmek için kullanılır. Örneğin, **.com** ticari kuruluşlar için, **.org** genellikle kar amacı gütmeyen kuruluşlar için, **.gov** hükümet kurumları için, **.edu** eğitim kurumları için ayrılmıştır. Ayrıca, her ülkenin kendine özgü bir veya daha fazla ülke kodu TLD'si (ccTLD) vardır, örneğin **.uk** Birleşik Krallık için, **.de** Almanya için kullanılır.

İnternetin erken dönemlerinde, TLD'ler sınırlı sayıdaydı ve genellikle sitenin türünü veya amacını belirtmek için kullanılıyordu. Ancak, internetin büyümesiyle birlikte, daha fazla TLD'nin oluşturulması ve kullanılması gerekliliği ortaya çıkmıştır. Bu nedenle, günümüzde yüzlerce farklı TLD mevcuttur ve bunlara genel TLD'ler (gTLD) de dahildir, ki bunlar özel ilgi alanları, sektörler veya toplulukları hedefleyebilir, örneğin **.photography**, **.guru**, **.tech**. Bu genişleme, internet alan adları sisteminin daha esnek ve çeşitli olmasını sağlamıştır, aynı zamanda kullanıcıların ve işletmelerin web varlıklarını daha iyi tanımlamalarına olanak tanımıştır.

## DNS Sunucu Türleri

DNS sistemi, internetin telefon rehberi gibi olduğundan bahsetmiştik. DNS'in, alan adını sorgulayarak IP adresini elde ettiği süreçte üç ana oyuncu vardır:

### 1. Yetkili Ad Sunucusu (Authoritative Name Server)

Bu sunucu, belirli bir alan adı için "resmi" bilgi kaynağıdır. Her alan adının en az bir yetkili ad sunucusu vardır, ve bu sunucu o alan adına ait DNS

kayıtlarını tutar. DNS kayıtları, bir alan adının IP adresi, mail sunucuları (MX kayıtları), alt alan adları (CNAME kayıtları) gibi bilgileri içerir. Bir DNS sorgusu sonunda bu sunucuya ulaşılsa, gerekli bilgi doğrudan ve kesin olarak buradan alınır.

## **2. Özyineli Çözücü (Recursive Resolver)**

Kullanıcı bir web sitesine erişmek istediğinde, bu talep ilk olarak özyineli çözücüye gider. Özyineli çözücü, kullanıcı adına sorguyu çözmek için gerekli araştırmayı yapar. Bu süreç, önce genel bir yönlendirme (kök sunuculardan başlayarak) ile başlar ve sonunda istenen bilgiyi bulana kadar çeşitli DNS sunucuları arasında adım adım ilerler. Bu süreç sırasında özyineli çözücü, gerekli DNS kayıtlarını önbelleğine alır, bu da gelecekteki sorguların daha hızlı çözülmesini sağlar.

## **3. Kök Ad Sunucuları (Root Name Servers)**

İnternetin DNS hiyerarşisinin en üstündeki sunuculardır ve tüm TLD'lerin (örneğin, .com, .net, .org) adres bilgilerini barındırır. Bir özyineli çözücü, alan adının hangi TLD sunucusuna sorması gerektiğini bilmiyorsa, sorguyu çözmeye başlamak için önce kök ad sunucularına başvurur. Kök ad sunucuları, sorunun doğru TLD sunucusuna yönlendirilmesi için gerekli ilk yönlendirme bilgisini sağlar.

## **Daha iyi anlayalım**

DNS çözümleme sürecini 7 adımda daha iyi anlayalım:

### **1. Kullanıcı Bilgisayarından DNS Sorgusu:**

Bir kullanıcı, internet tarayıcısına [www.example.com](http://www.example.com) gibi bir URL girer. Bu eylem, kullanıcının bilgisayarının (istemci) DNS çözümlemesi için bir sorgu oluşturmaya neden olur. Bu sorgu, ilk olarak kullanıcının İnternet Servis Sağlayıcısı'nın (ISP) Recursive Resolver'ına gönderilir.

### **2. ISP'nin Recursive Resolver'ına Sorgu:**

Recursive Resolver, bu sorguyu alır ve öncelikle kendi önbelleğinde (cache) bu sorguya dair bir kayıt olup olmadığını kontrol eder. Eğer uygun bir kayıt mevcutsa, çözümleme işlemi burada sona erer ve IP adresi doğrudan kullanıcıya döndürülür. Eğer kayıt önbellekte bulunmazsa, Recursive Resolver sorguyu çözmek için bir dizi adım gerçekleştirir.

### **3. Kök DNS Sunucusuna Sorgu:**

Önbellekte uygun bir kayıt bulunamazsa, Recursive Resolver sorguyu DNS hiyerarşisinin en üstünde bulunan Kök DNS Sunucusuna (Root Name Server) yönlendirir. Kök sunucu, sorunun hangi Üst Düzey Alan Adı Sunucusu'na (TLD DNS Server, örneğin ".com") gitmesi gerektiği hakkında bilgi sağlar.

#### **4. TLD DNS Sunucusuna Sorgu:**

Recursive Resolver, Kök DNS Sunucusu'ndan aldığı yönlendirme bilgisiyle, ilgili TLD DNS Sunucusu'na (örneğin, ".com" için) sorguyu yönlendirir. TLD sunucusu, istenen alan adının (örneğin, "example.com") Yetkili Ad Sunucusu'nun (Authoritative Name Server) adresini sağlar.

#### **5. Yetkili Ad Sunucusuna Sorgu:**

Recursive Resolver, elde ettiği bilgiyle doğrudan alan adının Yetkili Ad Sunucusu'na sorgu yapar. Yetkili sunucu, [www.example.com](http://www.example.com) için gerekli DNS kaydını, örneğin A kaydını (alan adının IP adresi) içerir.

#### **6. DNS Kaydının İstemciye Dönmesi:**

Yetkili Ad Sunucusu, istenen DNS kaydını (IP adresi) Recursive Resolver'a gönderir. Recursive Resolver, bu bilgiyi kullanıcının bilgisayarına iletir ve böylece kullanıcı, istediği web sitesine tarayıcısı üzerinden erişebilir.

#### **7. Web Sitesine Bağlantı:**

Kullanıcı bilgisayar, aldığı IP adresi kullanarak doğrudan web sitesinin sunucusuna bir HTTP isteği gönderir ve web sitesinin içeriği kullanıcıya sunulur

### **Hypertext Transfer Protocol (HTTP)**

Hypertext Transfer Protocol(HTTP), internet üzerinde bilgi alışverişi yapılmasını sağlayan temel bir protokoldür. Web sayfalarının ve diğer verilerin, bir ağ üzerindeki sunuculardan istemcilere (genellikle web tarayıcılarına) aktarılmasını mümkün kılar. 1990'ların başında Tim Berners-Lee tarafından icat edilmiştir ve World Wide Web'in temel taşlarından biridir.

HTTP, istemci-sunucu modelini kullanır. Bu modelde, bir istemci (kullanıcının tarayıcısı gibi) bir sunucuya (web sitesini barındıran sunucu

gibi) bir istek (request) gönderir ve sunucu bu isteğe bir yanıt (response) ile karşılık verir. Bu istek ve yanıt mekanizması, web sayfalarının, resimlerin, videoların ve diğer içeriklerin internet üzerinden alınıp gönderilmesini sağlar.

## HTTP İstekleri (Request)

HTTP istekleri, aşağıdaki dört temel bileşenden oluşur:

- URL
- Method
- Başlıklar
- İstek Gövdesi (opsiyonel)

Aşağıda örnek bir HTTP başlığı bulunmaktadır:

```
GET /index.html HTTP/1.1
Host: www.example.com
Cookie: SESSIONID=badb655ebd99ed6c8e58c0a1aab44eb9
```

- İlk satırda HTTP Methodu, isteğin gönderildiği yol (path) ve HTTP protokolünün versiyonu bulunmaktadır. Örneğimizde, isteğin **GET** methodu ile **index.html** yoluna gönderildiği görülmektedir. HTTP versiyonu ise **HTTP/1.1** olarak belirtilmiştir.
- İkinci satırda, isteğimizin gideceği hedef sunucunun adresi yer almaktadır. Hedef sunucu adresine alan adı veya IP adresi yazılabilir. Örneğimizde [www.example.com](http://www.example.com) alan adına istek gönderildiği görülmektedir.
- Üçüncü satır, istemci tarafında veri saklamak için kullanılan bir çerez (cookie) başlığı içeriyor. Web sitelerindeki oturum açma süreçleri genellikle çerezler ile yürütülür. Bu örnekte, hedef sunucuya istemciyi tanıyabilmesi için **SESSIONID** isminde bir çerezi **badb655ebd99ed6c8e58c0a1aab44eb9** değeri ile gönderildiği anlaşılmaktadır.



## HTTP İstek Methodları (Request Method)

HTTP istek methodları sunucuya, istemcinin sunucudan ne tür bir işlem yapmasını istediğini söyler. En yaygın yöntemler şunlardır:

HTTP Metodu	Tanım
GET	Bir kaynağı almak için kullanılır; URL ile belirtilen kaynağın içeriğini döndürür.
POST	Sunucuya veri göndermek için kullanılır; genellikle form verilerini gönderirken kullanılır.
PUT	Belirli bir kaynağı oluşturmak veya mevcut bir kaynağı güncellemek için kullanılır.
DELETE	Belirli bir kaynağı silmek için kullanılır.
HEAD	GET benzeridir, ancak kaynağın kendisi yerine başlıkları (headers) döndürür.
OPTIONS	Bir kaynağın hangi HTTP metodlarını desteklediğini sorgulamak için kullanılır.
PATCH	Bir kaynağın bir kısmını güncellemek için kullanılır.
CONNECT	İstemci ile sunucu arasında tünel kurmak için kullanılır.
TRACE	İsteğin sunucuya ulaşma yolunu test etmek için kullanılır.

## HTTP İstek Başlıkları (Request Header)

HTTP istek başlıkları (request headers), bir HTTP isteğinin parçası olarak sunucuya gönderilen ek bilgileri içerir. Bu başlıklar, isteğin nasıl işleneceği, istemcinin tercihleri, gönderilen içeriğin türü gibi çeşitli bilgileri taşır. İstemci ve sunucu arasındaki iletişimi daha etkili ve anlamlı hale getirmek için kullanılırlar.

Aşağıdaki örnekte olduğu gibi isim/değer formatında yazılırlar.

NAME=VALUE

Bu başlıklar sunucuya, isteği işlemek veya yanıtı özelleştirmek için kullanılabilecek bağlam ve ek talimatlar sağlamaktadır.

Aşağıda örnek bir HTTP başlığı bulunmaktadır:

```
GET /api/info HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.82 Safari/537.36
Accept: application/json
Accept-Language: en-US,en;q=0.5
Authorization: Token secret123
Cache-Control: no-cache
Connection: keep-alive
Referer: https://www.google.com/
Pragma: no-cache
```

Bu örnekte, GET methodu ile /api/info yoluna HTTP/1.1 protokolü ile istek gönderilmiştir. Ayrıca bu istek, 10 farklı başlık örneği içermektedir:

## Başlıklar

Host

User-Agent

Accept

Accept-Language

## Tanımı

Bu başlık, istemcinin bağlanmaya çalıştığı sunucunun ana bilgisayar adını belirtir.

Bu başlık, isteği yapan tarayıcı hakkında bilgi sağlar (örnekteki User-Agent değeri, Chrome tarayıcısı olduğunu göstermektedir).

Bu başlık, istemcinin yanıtta kabul etmeye istekli olduğu MIME veri türlerini belirtir.

Bu başlık, yanıt için tercih edilen dil(ler)i belirtir.

Authorization

Bu başlık, kimlik doğrulama amaçları için bir erişim belirteci(token) sağlar.

Cache-Control

Bu başlık, hem istekler hem de yanıtlar için önbellekleme yönergelerini belirtir.

Connection

Bu başlık, istemci ve sunucu arasındaki bağlantıyı yönetmek için seçenekler belirtir.

Referer

Bu başlık, mevcut sayfaya bağlantı veren sayfanın URL'sini belirtir.

Pragma

Bu başlık, istek-yanıt zinciri boyunca herhangi bir ajanı etkileyebilecek uygulamaya özgü yönergeleri belirtir.

Content-Type

Bu başlık, istek gövdesinde gönderilen verinin MIME türünü belirtir, ancak bu örnekte bir GET isteği olduğu için ve istek gövdesi kullanılmadığı için kullanılmamıştır.

## HTTP İstek Gövdesi (Request Body)

HTTP request body, bir HTTP isteğinin gövdesidir ve istemciden sunucuya gönderilmek üzere paketlenen veriyi içerir. Genellikle POST, PUT, PATCH gibi metodlarla yapılan isteklerde, form verileri, dosyalar veya JSON/XML gibi yapılandırılmış veriler bu bölümde yer alır. Örneğin, bir kullanıcı formu doldurup gönderdiğinde, formdaki bilgiler request body içinde sunucuya iletilir; ya da bir API'ya JSON formatında bir veri göndermek istendiğinde, bu JSON verisi request body içinde yer alır.

HTTP request body'nin içeriği ve formatı,

**Content-Type** başlığı ile belirlenir. Bu, sunucunun gönderilen veriyi nasıl işleyeceğini anlamasına yardımcı olur. Örneğin, **Content-Type: application/json** başlığı, gövdenin JSON formatında olduğunu ve sunucunun bu formata göre işlem yapması gerektiğini belirtir. Bu mekanizma, istemci ile sunucu arasında daha karmaşık ve anlamlı veri alışverişinin yapılabilmesine olanak tanır, web uygulamalarının ve API'lerin dinamik ve etkileşimli hale gelmesini sağlar.

## Request Body içeren örnek bir HTTP başlığı:

```
POST /v1/photos/upload HTTP/1.1
Host: api.myphotos.com
x-api-key: API_KEY
Content-Length: 232
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW

-----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="file"; filename="/C:/Users/John/Photos/my_photo.jpg"
Content-Type: image/jpeg

(data)
-----WebKitFormBoundary7MA4YWxkTrZu0gW--
```

Yukarıdaki istekte,

api.myphotos.com adresindeki sunucunun /v1/photos/upload yolundaki API uç noktasına POST methodu ile yerel bilgisayarda bulunan C:/Users/John/Photos/my\_photo.jpg yolundaki fotoğrafın yüklenildiği görülmekte.

Sunucu bu isteği aldığı anda, istek gövdesini ayrıştıracak ve fotoğrafların bulunduğu veritabanında yeni bir giriş oluşturmak için kullanacaktır.

## HTTP Yanıtları (Response)

HTTP yanıtı, bir web sunucusunun bir istemcinin HTTP isteğine verdiği cevaptır. Bu yanıt, istenen kaynağın durumunu, sunucunun kendisi hakkında bilgileri ve isteğe bağlı olarak istenen kaynağın kendisini içerir.

```
HTTP/1.1 200 OK
Date: Sun, 28 Mar 2023 10:15:00 GMT
Content-Type: application/json
Server: Apache/2.4.39 (Unix) OpenSSL/1.1.1c PHP/7.3.6
Content-Length: 1024

{
  "name": "John Doe",
  "email": "johndoe@example.com",
}
```

İlk satırda yer alan **durum satırı(status line)** HTTP sürümünü, isteğin sonucunu gösteren bir durum kodunu gösterir.

Aşağıdaki tabloda en yaygın karşılaşılan durum kodları listelenmiştir:

Kod	Anlamı
100	Devam
101	Protokolleri Değiştir
200	Başarılı
201	Oluşturuldu
202	Kabul Edildi
203	Yetkili Olmayan Bilgi
404	Bulunamadı: İstenen kaynak sunucuda bulunamadı.
500	İç Sunucu Hatası: Sunucu, isteği işlerken bir hata ile karşılaştı.
301	Kalıcı Olarak Taşındı: İstenen kaynak kalıcı olarak yeni bir URL'ye taşındı.

Örnekteki Content-Type başlığından da anlayacağımız üzere, yanıtın gövdesi application/json yani **JSON** formatındadır. Yapılan istek sonucunda John Doe isimli kullanıcının bilgilerinin dönüldüğü görülmektedir.

## Adım Adım Çalışma Prensibi

Tüm bu öğrendiklerimizi özetlemek için, bir web sitesine erişmek istediğimizde gerçekleşen adımların ne olduğuna bakalım:

### 1. Web Tarayıcısını Açma ve URL Girme:

İnternette gezinmeye başlamak için ilk adım, bir web tarayıcısı açarak ziyaret etmek istediğiniz web sitesinin adresini, yani URL'ini (Uniform Resource Locator) girerek başlar. Örneğin: <https://google.com>

### 2. DNS Sorgusu:

Girdiğiniz URL, insanlar için anlamlı ve hatırlaması kolay bir adrestir; fakat internet ağları ve bilgisayarlar bu adresler yerine sayısal IP adreslerini

kullanır. Bu nedenle, tarayıcınız URL'yi, web sitesinin barındırıldığı sunucunun IP adresine çevirmek için bir Domain Name System (DNS) sorgusu yapar. DNS, internetin telefon rehberi gibidir; girdiğiniz web sitesi adını alır ve bunu karşılık gelen IP adresine dönüştürür.

### **3. Sunucuya İstek Gönderme:**

DNS sorgusunun sonucunda elde edilen IP adresi ile tarayıcınız, web sitesinin barındırıldığı sunucuya bir HTTP isteği gönderir. Bu istek, sunucudan belirli bir sayfanın içeriğini talep eder.

### **4. Sunucudan Yanıt Alınması:**

Web sunucusu, tarayıcınızın isteğini alır ve gerekli işlemleri yaparak istenilen web sayfasının içeriğini bir HTTP yanıtı ile geri gönderir. Bu yanıt, genellikle HTML (Hypertext Markup Language) kodlarından oluşur ve sayfanın nasıl görüneceğini tanımlar.

### **5. Web Sayfasının Görüntülenmesi:**

Tarayıcınız, sunucudan aldığı HTML içeriğini işler ve kullanıcıya görsel bir web sayfası olarak sunar. Bu aşamada, sayfanın düzgün bir şekilde görüntülenebilmesi için ek kaynakların (resimler, CSS dosyaları, JavaScript dosyaları vb.) de sunucudan istenmesi ve alınması gerekebilir.

### **6. Bağlantının Sonlandırılması:**

Web sayfası başarıyla yüklendikten sonra, HTTP/1.1 protokolünde **keep-alive** özelliği olmadığı takdirde, istemci ve sunucu arasındaki bağlantı otomatik olarak sonlandırılır. Eğer kullanıcı sayfa içinde başka bir linke tıklar veya yeni bir sayfa isteği yaparsa, bu süreç yeniden başlar ve yeni bir bağlantı kurulur.

## **SORULAR-**

### **Soru 1**

A-)WWW kısaltmasının tam açılımı nedir?

B-)World Wight Web

- C-)World Wide Web
- D-)World Web Wide
- E)Web World Wide

## SORU2

internet ile Web arasındaki temel fark nedir?

- A-Her ikisi de e-posta ve dosya transferi hizmetleri sağlar.
- B-Web, 1960'larda ABD Savunma Bakanlığı tarafından geliştirilmiştir.
- C-Internet ve Web, teknik olarak aynı şeydir ve aralarında hiçbir fark yoktur.
- D-Internet, dünya çapında milyonlarca bilgisayarın birbirine bağlandığı bir ağıdır; Web ise bu ağ üzerinde çalışan bir bilgi paylaşım sistemidir.

## Soru 3

- A-Web'in kullanıcılara sunduğu özelliklerden biri aşağıdakilerden hangisidir?
- B-Yalnızca metin tabanlı bilgi sunumu.
- C-E-posta ve dosya transferi gibi hizmetlerin sağlanması.
- Bilgisayar ağlarının birbirine bağlanması ve ağ oluşturulması.
- D-HTML diliyle hazırlanmış web sayfalarına erişim ve bu sayfaların URL'ler aracılığıyla birbirine bağlanması.

## Soru 4

Sunucuların temel amacı nedir?

- A-Web sitelerini görüntülemek ve kullanıcı etkileşimini sağlamak.
- B-İstemcilerden gelen bilgi veya hizmet taleplerini işlemek ve gerekli yanıtları göndermek.
- C-İnternet bağlantısı sağlamak ve ağ oluşturmak.
- D-Kişisel bilgisayarlar, akıllı telefonlar ve tabletler gibi cihazlar aracılığıyla internete erişim sağlamak.

### Soru 5

A-İstemci-sunucu modelinde, istemcinin rolü nedir?

İnternette bilgi veya hizmet talep etmek.

B-Web sitelerini barındırmak ve hizmetleri sağlamak.

C-Ağ güvenliğini sağlamak ve siber saldırılara karşı koruma sunmak.

D-Verileri yedeklemek ve depolamak.

### Soru 6

Recursive resolver, alan adının hangi TLD sunucusuna sorması gerektiğini bilmiyorsa, ilk olarak hangi sunuculara başvurur?

A-Bölgesel DNS Sunucuları

B-Yetkili Ad Sunucuları

C-Root DNS Sunucuları

D-Önbellek DNS Sunucuları

### Soru 7

HTTP istekleri aşağıdaki dört temel bileşenden hangilerinden oluşur?

A-URL, Method, Başlıklar, İstek Gövdesi

B-IP Adresi, Domain Adı, Port Numarası, Protokol

C-Kullanıcı Adı, Parola, Session ID, Cookie

D-Tarayıcı Adı, İşletim Sistemi, Ekran Çözünürlüğü, Diller

### Soru-8

Genellikle sunucuya form verilerini göndermek için kullanılan HTTP istek methodu nedir? **POST**



### Soru-9

Soru 11

Belirli bir kaynağı almak için kullanılan HTTP istek methodu nedir?

GET

### SORU-10

OPTIONS HTTP methodunun kullanım amacı nedir?

A-Bir kaynağı güncellemek

B-Bir kaynağın hangi HTTP metodlarını desteklediğini sorgulamak

C-Sunucuya veri göndermek

D-Bir kaynağı silmek

### Soru 11-

İstek yapan tarayıcı veya istemci uygulamasının türünü ve sürümünü belirten HTTP istek başlığı nedir?

User-Agent



WEB TEMELLERİ AKADEMİ KISMINI TAMAMLAMIŞ BULUNMAKTAYIM

EMİRHAN KELLEÇİGİL

25.12.2025

KAYNAK HACKVİSER