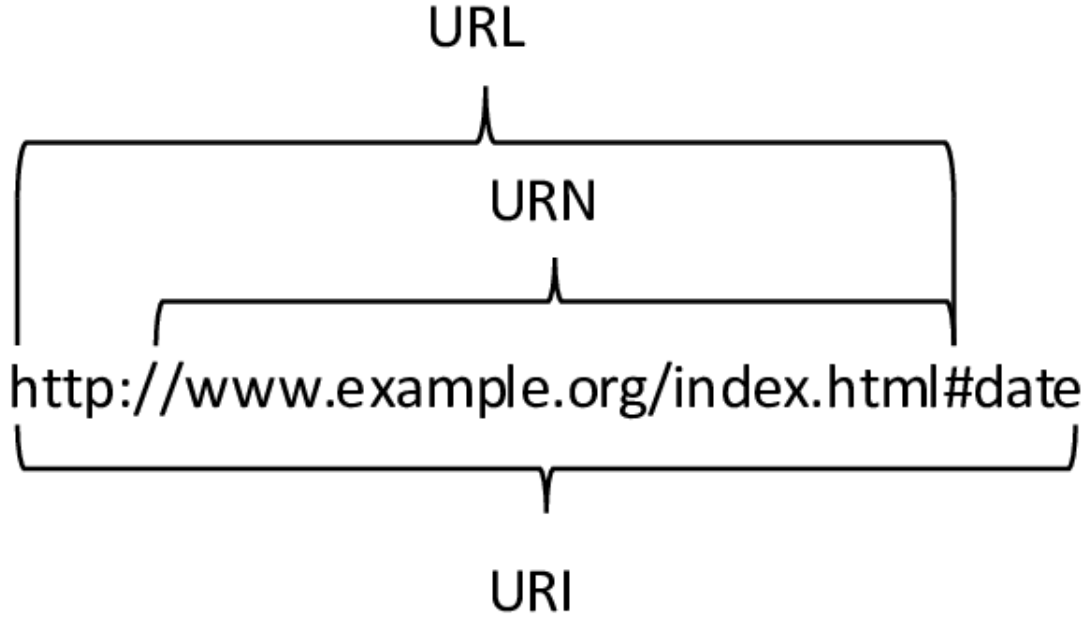


URI, URL ve URN



URI (Uniform Resource Identifier), bir adı veya bir kaynağı tanımlamak için kullanılan bir karakter dizisidir. Bir kaynağı konuma, ada veya her ikisine göre tanımlar. Eğer bir URI bir kaynağı konuma tanımlarsa aynı zamanda bir URL, ada göre tanımlarsa aynı zamanda bir URN'dir. URN ve URL, URI'nın alt kümeleridir. Örnek; <https://www.ietf.org/rfc/rfc2396.txt>

URL (Uniform Resource Locator), bir kaynağı tanımlayan ve aynı zamanda bu kaynağa ulaşmamız için gereken bilgiyi içeren bir URI'dir. Her URL bir URI'dir ancak her URI, URL değildir. Örnekler;

- <ftp://ftp.is.co.za/rfc/rfc1808.txt>
- <http://www.ietf.org/rfc/rfc2396.txt>
- [ldap://\[2001:db8::7\]/c=GB?objectClass=one](ldap://[2001:db8::7]/c=GB?objectClass=one)
- <mailto:John.Doe@example.com>
- <telnet://192.0.2.16:80/>

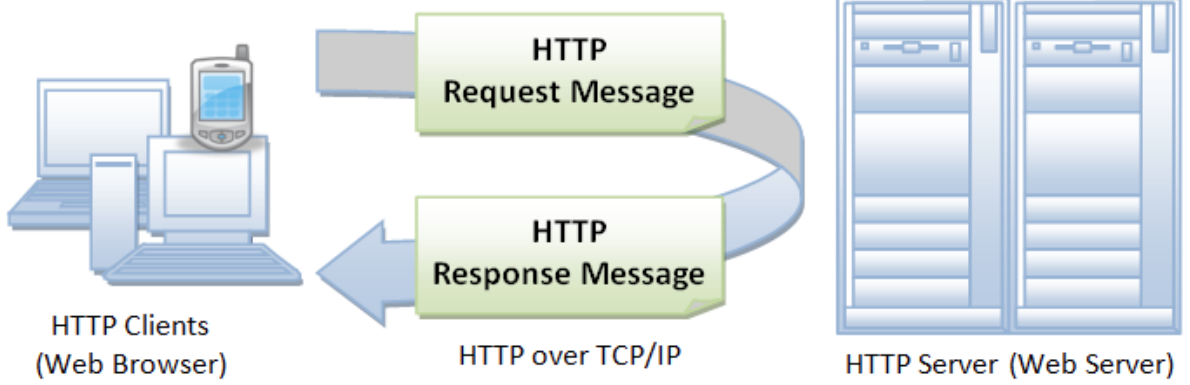
Kıyas yapmak gerekirse;

- URL, bir kaynağı tanımlayan ve ayrıca kaynağa erişmenin yolunu açıklayarak kaynağı bulma aracı sağlayan bir URI'dir.
- Bir URL bir URI'dir ama bir URI mutlaka bir URL değildir.

URN (Uniform Resource Name), bir kaynağı benzersiz bir adla tanımlar ancak bunun nerede bulunduğunu belirtmez. URL'den farklı olarak, URN'ler kaynağın konumunu belirtmez. Bir URN genelde 'urn:' prefix'i ile başlar. Örnekler;

- urn:oasis:names:specification:docbook:dtd:xml:4.1.2
- tel:+1-816-555-1212

HyperText Transfer Protocol



Bir web sayfası bir veya birden fazla nesneden oluşur. Bir nesne belirli bir URL üzerinden erişilebilen video, jpeg görüntüsü, HTML gibi bir dosyadır. Genelde web sayfalarına erişirken bir HTML dosyasına erişiriz ve onun içerisinde diğer nesnelere referanslar olur. HTML dosyaları ulaşmak istediğimiz içeriği taşıırken onun referans olduğu nesneler (CSS, JavaScript, medya dosyaları vb.) barındırır. HTTP ise bu sayfaların sistematik olarak server ve client arasında alışverişini sağlar.

İstemcilerin (client) web sayfalarını sunuculardan (server) nasıl isteyeceğini ve sunucuların bu sayfaları istemcilere nasıl aktaracağını HTTP tanımlar. Örnek olarak; istemciler genellikle bilgisayarlarımızda çalıştırdığımız tarayıcılardır, sunucular ise uzak bilgisayarlarda sürekli olarak açık tutulan programlardır. Tarayıcılar web sunucularına HTTP isteğinde bulunur, sunucular ise bunlara HTTP cevabı ile karşılık verir. Bir sunucuya aynı anda birden çok HTTP isteği gelebilir.

HTTP İstekleri

1. GET/POST/PUT/DELETE: İstemcinin sunucudan istekleri
2. Host: İstekte bulunulan dosyaların barındırıldığı uzak bilgisayarın web adresi
3. User-agent: Kullanıcının tarayıcı ve bilgisayar bilgileri (Örnek; Mozilla/5.0)
4. Accept: Kabul edilecek format (Örnek; HTML, JSON)

HTTP İstek Metodları

- GET: Sunucudan veri almak için kullanılır ve özellikle kullanıcı şifresi gibi hassas içeriklerin barındırıldığı isteklerde kullanılması tavsiye edilmez.
- POST: Host'a bir veri kümesi veya sadece bir veri göndermek için kullanılır.
- PUT: Sunucu tarafında zaten mevcut olan bir kaynağın içeriğini değiştirmek için kullanılır. Programlarımızda veritabanında kayıtlı bulunan bir verinin tamamını ya da bir kısmını değiştirmek için bu metodu kullanırız.
- DELETE: Kayıtlı olan bir kaynağın tamamen silinmesi için kullanılır.

HTTP Cevapları

1. HTTP Durum Kodu: Protokol versiyonu, durum kodu ve ona karşılık gelen mesaj
2. Date: Gelen istek sonucunda sonucunun istenilen nesneleri kendi dosya sisteminden çekip istemciye sunduğu anın tarihidir
3. Server:Sunucu ile ilgili bilgidir. İstek mesajında “User-agent” bilgisine benzer
4. Content-Type: Sunulan nesnelerin türüdür

HTTP Durum Kodları

HTTP durum kodları genel kategorilere ayrılır ve bu kategorilere göre ilk sayılar değişir;

- 1xx: Bilgi mesajları
- 2xx: Başarılı
- 3xx: Yeniden yönlendirme
- 4xx: İstemci hatası
- 5xx: Sunucu hatası

Bazı durum kodları:

- 200 OK: İstek başarılı
- 401 Unauthorized: Yetkisiz erişim
- 404 Not Found: Dosya bulunamadı
- 500 Internal Server Error: Sunucu içerisinde hata

Node.js

Node.js, V8 isimli bir javascript motoru üzerinden çalışır. V8, Google tarafından geliştirilen, Chrome web browserlarının da üzerinde çalıştığı C, C++ ve javascript dilleri ile kodlanan açık kaynak kodlu bir motordur.

Node.js; bir javascript kodunun sadece tarayıcılarda değil aynı zamanda bilgisayarınızda bağımsız şekilde çalışacak bir uygulama şeklinde kullanmak istenmesinden ortaya çıkmıştır.

Node.js'in kullanılmasında öne çıkan bazı nedenleri;

- Non-Blocking Teknolojisi: bir uygulama üzerinde bir işlem yaparken işlemlerin birbirini beklemediği, asenkron olarak gerçekleştiği anlamına gelir. Bu da zaman ve kaynak kullanımı konusunda avantaj sağlar.
- NPM (Node Package Manager): uygulamamızı daha hızlı ve verimli hale getirmek için geliştiricilerinin yazılımlarını açık kaynak kodlu olarak paylaştığı, topluluk tarafından oluşturulmuş bir kütüphanedir.

Node Package Manager

Node Package Manager ya da Node Packaged Modules, Isaac Z. Schlueter tarafından tamamen javascript dili kullanılarak geliştirilmiştir.

Npm temel olarak komut satırı üzerinden çalışan, üçüncü parti yazılımları yüklemeyi sağlayan bir uygulamadır. Npm ile temel olarak yapabileceğimiz şeyler ise;

- Otomatik ya da manuel olarak paketleri yükleme
- Sistemdeki paketleri silmek
- Sistemdeki paketleri listeleme
- Sistemdeki paketleri update etmek

Java 8 ve Java 11

Java 8 ile Java 11'in hala bu kadar popüler olmasının temel nedenlerinden biri, uzun süreli destek sürümü olmasıdır. Bu politika kullanıma sunulduğundan beri yalnızca Java 8 (2014) ve Java 11 (2018) uzun süreli destek sürümü olarak piyasaya çıkmıştır.

Hiçbir kuruluş uzun süreli desteklenmeyen bir Java sürümüne dayanan bir sistemi üretime geçirmedikten dolayı diğer Java sürümlerine kıyasla Java 8 ile Java 11'in popülerliğini sağlamıştır. Oracle Java 8'e 2014'ten 2019'a kadar ücretsiz destek vermiştir. 2019'dan sonra ise Aralık 2030'a kadar ücretli destek vermektedir.