

Cloud Applications Architecture



Course 5 - Web Capabilities

Topics

(Technologies that help us build and consume services on the web)

- **HTTP**
- **Content**
- **Hosting**
- **CDNs**
- **Domains**
- **DNS**

URI/URL/URN

URI/URL/URN

Universal Resource Identifier: used to identity a resource (doesn't have to be internet related).

E.g. *ISBN 1234*

Universal Resource Locator: a more specific URI that also describes how to the get the resource over a given protocol.

E.g. *http://books.com/books/1234*

Universal Resource Name: a more specific and standardized URI.

E.g. *urn:isbn1234*

URL

<https://www.caacourse.com:443/courses?status=published>

Protocol

Subdomain

Domain

Port

Path

(Query) Parameters

HTTP

HTTP Concepts

Hypertext Transfer Protocol

Defines how messages should be structured.

Layer 7 (Application).

Read more from some well-known sources: [Mozilla](#), [Cloudflare](#)

Stateless protocol (requests are independent)

- As opposed to TCP which is stateful for example

Common actors: client (user-agent), server, proxies

HTTP Versions

HTTP/1.0

- Initial version, required a new TCP connection for each request

HTTP/1.1

- Allowed TCP connection reuse
- Still widely used today

HTTP/2

- Derived from Google's SPDY protocol
- Incremental changes over HTTP/1.1; most notable being Server Push

HTTP/3

- Entirely different than previous versions. Works over UDP

HTTP Methods

Safe methods (no change expected)

GET

HEAD

OPTIONS

These methods are also **idempotent**.

Not safe methods (change expected)

POST

PUT (Idempotent)

DELETE (Idempotent)

PATCH

It's still our job to use these methods as intended and expected

HTTP Headers

Commonly used headers

- **Content-type/Accept:** *application/json, text/plain, */**
- **Cache-Control:** *no-cache, max-age=<seconds>*
- **Authorization:** *Basic <base64 credentials>, Bearer <token>, etc*
- **Access-Control-Allow-Origin:** *https://caacourse.com* (needed for CORS)
- **Origin:** *https://caacourse.com*

Some headers are automatically handled by browsers (e.g. CORS).

A common practice is to prefix custom headers with **x-**, but it's not really necessary nor encouraged.

Watch [this](#) video for some great capabilities headers offer us.

HTTP Header Fields

Can be grouped as

- **General:** Request URL, Request Method, Status Code, Remote Address
- **Request:** Cookies, Accept-xx, Content-Type, Content Length, Authorization, User-Agent
- **Response:** Server, Set-Cookie, Content-Type, Content-Length, Date

Content

Types of Content

Images

Video

User data

Text

Binary

Applications (i.e. html, css, js)

Static

Hello!

Dynamic

Hello, *John*!

Content Storage

Common cloud services

- Amazon S3 (Simple Storage Service)
- Google Cloud Storage
- Azure Storage Account (different sub-services: Blob, File, Queue)

Content Caching

Content can be cached at multiple levels

- Server
- CDN (Content Distribution Network)
- Browser

CDN and Browser caching is usually controlled through Header Cache-Control

Both static and dynamic content can be cached, but with different **TTL** (time to live).

Various techniques to refresh the content such as purging and cache-busting.

Content Distribution

Common issues

- Latency (for global distribution)
 - Solved by storing the content across multiple locations, closer to more users
- Paid content
 - Storage services usually support **signed requests/URLs**.

Website Hosting

Website Hosting

“Static” Websites

I.e. no processing is required to serve the website (html, css, js files).

Cheap/Free hosting. Can leverage storage services (e.g. S3).

Jamstack

(These also include react/angular/vue etc. apps)

“Dynamic” Websites

I.e. The server generates the website files when requested.

E.g. Wordpress sites.

Require a server with dedicated CPU.

Content Delivery Networks (CDN)

What is a CDN?



Edge caching

Source: [Cloudflare](#)

Why CDNs?

Improve load times

- Content is physically closer to the end-user

Optimize costs

- For content heavy applications, traffic is one major cost driver

Reduce pressure on critical resources

- Our system only loads the content into the CDN

Improved security

- Most CDNs provide features such as DDoS protection

CDN Examples

- [AWS CloudFront](#)
- [Google Cloud CDN](#)
- [Azure CDN](#)
- [Cloudflare CDN](#)
- [Akamai CDN](#)

Some providers also support edge computing. E.g. [Lambda@Edge](#)

Domains

Domains

Domains, subdomains, domain registrars

Relative paths, local development, virtual hosts.

Multiple environments (can be deployed differently): dev, stage, prod.

Domain Name System (DNS)

DNS

Translates **domains/hostnames** to **IP addresses**.

Multiple layers. Companies might even have internal DNS.

DNS Actors:

- DNS Recursor
- Root nameserver
- TLD nameserver
- Authoritative nameserver

DNS Records

- **NS (Name Server)**
 - Indicates the DNS server handling DNS requests
- **A Record**
 - Points the domain to the server (actual IP address)
 - 172.217.17.238 -> google.com
- **CNAME (Canonical Name)**
 - Points an “alias” (sub)domain to the real (sub)domain
 - www.google.com -> google.com
- **MX (Mail Exchange)**
 - @ -> mxa.mailgun.org (these also usually have a priority, e.g. 10)
- **TXT**
 - General purpose. Usually used to verify ownership of the domain

DNS Services

AWS Route53

Google Cloud DNS

Google DNS (8.8.8.8) (Public)

Azure DNS

Cloudflare DNS

Cloudflare 1.1.1.1 (Public)

DNS TTL

Clients usually cache the association between a domain and its IP address.

The caching time is given by the DNS Record TTL.

Relevant especially when we change DNS records.

Summary