

# Introduction

Friday, 4 August 2017  
10:52 am

## Things

- Anything in blue hyperlinks are assumed knowledge
- Project 60%, Final exam 40%
- Week 13 is revision
- HINT: Deploy before the last week

## HTTP Return Codes Cheat Sheet

1XX	Hold on	Wait for a final response
2XX	Here you go	Success
3XX	Go away	Redirect. E.g. the content was moved.
4XX	You fucked up	Error caused by client
5XX	I fucked up	The server failed to fulfil a request

# Lecture 1 Web Fundamentals

Friday, 4 August 2017

5:35 pm

*What happens when you enter a URL into the browser?*

Parsing a URL

Protocol://domain.tld/path/file.ext

- Protocol is often HTTP(S)
- TLD: top level domain

Check browser cache

- Local browser cache
  - Has it already requested something in the past?
  - Problem with keeping things up to date
  - Can check by asking server if something has been modified since...

Resolve IP to DNS

- DNS propagates, iterative
- Using UDP

Establish TCP connection

- TCP is sent and received in order
- UDP is sent, but receipt is not guaranteed and the result doesn't matter

HTTP Request

- Has information about
  - Host: domain.tld
  - User agent (browser, operating system)
  - Language
  - ...
- Broken into two parts
  - Head: contains metadata such as cookie and connection information
  - Body: Contain the data to be sent with the transmission (optional)
- Several types of web requests
  - GET: A simple request header ONLY
  - POST: Contains data in the bod,,; e.g. forms data, file upload

Web servers

- Usually the first recipient of the HTTP request is NOT the service that provides the response (security)
- Load balances may interface to multiple web servers
- DDOS mitigation services

Application servers

- App servers process the HTTP request

Application databases

- SQL: relational
- NoSQL: non-relational

### Slave/Master Databases

- Typically, one master (can read and write)
- Can have many slaves (only read)
- More slaves, more capacity
- **Slave drift**: time delay when replicating information from master to slave. Needs to be managed because race conditions.

### Rendering a HTTP response

#### Returning a HTTP response

- Header
  - HTTP code to signal OK
  - Cache control: should the browser store this for next time?
- Body
  - Any data

#### Parsing a HTML Response

- 200: Success
- 301: Moved permanently
- 302: Moved temporarily
- 4XX: Client error
- 5XX: Server error

#### Parse HTML into DOM

- DOM: Document object model
- Think of HTML as the content
  - CSS is how you style the content

#### Fetching DOM Resources

#### Browser rendering

### Web Hosts

#### Web server

- Mainly hosted on Linux

#### Web server software

- Apache, Nginx (engineX), IIS

#### Summary

- Parse the URL
- Check the cache
- Resolve the IP
- Establish TCP connection
- Send http request
- Web server receives the request and sends to app server
- App server process the request and send back HTTP response
- Client receives HTTP response and parse to HTML
- Parse HTML to DOM
- Render DOM to screen

# Lab 02

Friday, 11 August 2017

3:48 pm

## **Make the request**

- Parse the URL - `subdomain.domain.gTLD.ccTLD`
- Check cache - JS/CSS/Images - HTTP 304 Not Modified
- Identify network - DHCP - IP/Subnet Mask/ Default gateway/NameServer (DNS)
- Resolve IP using DNS - UDP Port 53 - Iterative searching
- Establish TCP connection - Transport Layer (segment) - HTTP Port 80/ HTTPS Port 443
- Send HTTP Request - Header + Body - CRUD

## **Response**

- LB establish TCP connection and Receive HTTP Request (DDoS mitigation services)
- LB software parses HTTP header, then routes and forwards request to an app server
- App server (Python, PHP, Ruby, JavaScript etc) executes back-end website logic, probably connecting to one or more databases (SQL, NOSQL)
- App server renders a response

## **Parse a Web Response**

- Parse HTML into DOM
- Initiate further connections to
  - Fetch assets (CSS, JS, Images)
  - Establish WebSocket connections
  - Load browser plugins (AdBlock)
- Render page

## **IDEAS**

- Generating movie scripts using Markov chains
- Multiplayer tic-tac-toe
- Multiplayer dots and boxes