# Notes from 9th August

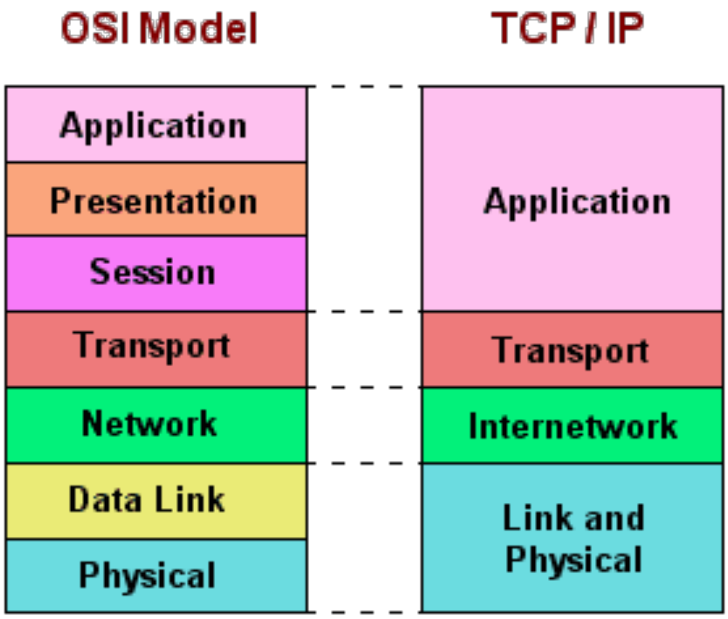
**Objectives:**

- Know basic terminology
- describe client/server
- explain http requests
- describe crud and how it relates to http verbs
- explain the make up of a url including parameters
- make basic http requests to a restful API to conduct crud operations
- recognise JSON XML and HTML
- Describe purposes of cookies

## Terminology

First version of internet: 1969 arpernet, method for universities to share papers etc.
log i
WWW - Tim Berners Lee , system for linking together web pages , big revelation- links
Intranet - Internal network for a system or company for internal communication
E-commerce: online shops, purchasing online and advertising online
B2B: business to business -> Gira/trello tools specifically for businesses to communicate with other businesses
B2C: Business to customer -> Amazon , stores in general
C2C: Facebook e.g.

Difference between a website and an web app, a web app is more interactive a website is information based



HTTP request - HTTP response part of the http protocol , allows for the server to request and respond with data

Tim berbers-lee chose HTTP, seven HTTP process— get post put patch delete select http finds the web page via the IP, works hand in hand with DNS.

Client GET —> www.google.com ——> server
Client<—the google home page< ——server
 POST —> A search term —> server
Client <— Results page <— server
put and patch update data on a server
put will edit entire sections of data
patch will partially update data
large data make patch much more efficient
delete does the obvious

URL - Uniform Resource Locator
two parts of a URL
the first being the protocol identifier - identifying the protocol e.g http:// or ftp://
then the resource locator - either by ip e.g 198.162.0.1 or www.google.com
after that you have paths for example www.google.com**/search/**…..
fragment - the end part of the URL, a section of a web page, that may have been bookmarked or anchored by the web dev, e.g linking within the same page

**CRUD**
Create read update delete
allows a user to interact with data on a web page
delete
DELETE /todos/5
GET /todos/5
GET /todos/ (List all)
POST /todo
PATCH/PUT /todo/4

**RESTful Roots**

- REST is a function to be able to interact with a server
- 1st root index: go and get all of the REST GET domain/fish
- new: GET ~/fish/new (gets the page that allows you to create a new fish)
- create: POST ~/fish (puts a new fish) used with ~/fish/new
- edit: GET ~/fish/:id/edit (takes to the page to edit a fish, carries the id of the fish to edit)
- update: PUT: ~/fish/:id
- show: SHOW ~/fish/:id (Gets one unique fish)
- delete: DELETE ~/fish/:id (Deletes a unique fish)

i'm only resting- play

Response codes
200 OK
201 Created
302 found

304 not modified
400 bad request
403 forbidden
404 not found
408 request timeout
500 internal server error
502 bad gateway
503 Service unavailable
http
JSON you can use ARRAYs and refer to them  list.0

git checkout, resets all changes to last commit