MERN Stack Training / Web Full Stack / Javascript Full Stack

Duration: 140 hours

MongoDB: NoSQL Database to maintain the data

Express.js: Backend services

React.js: Frontend

Node.js: Runtime environment

Contents:

* Applications
* Fundamentals of Computer Programming
* Algorithms & Pseudocode
* Linux
* Git & Git Hub
* HTML
* CSS
* Javascript & Typescript
* React.js
* Node.js
* Express.js (REST Api’s)
* MongoDB
* DevOps

Applications: They are computer programs which can perform tasks for the users, there are 2 types of applications

1. Standalone
2. Distributed

Standalone applications: These are applications which you can used only after running in the machine

Two types

* Desktop based: This can be installed on your desktop
* Mobile based: This can be installed on your mobiles

ex: Word document, Mobile based applications

Distributed applications: These are the applications which you can access over the internet

ex: Banking services, Twitter, Facebook, Gmail, Food Ordering Systems, Shopping cart applications

How do we access distributed applications

We can access them using Browser or Client applications

Browser: It is an application which helps to enter URL and use the distributed application

Client applications: These are the applications that can internally use the URL to access distributed application

ex: Mobile applications, ATM machine programs, Swiping Machines programs

URL: Uninform Resource Locator, it is a name given to the applications to easily access it, in real time the applications use IP address & Port numbers, but users can’t remember it hence, URL will be given

Browser:

Browser is an application which helps to access any websites through URL, there are many browsers

1. Google Chrome
2. Edge
3. Mozilla
4. Opera
5. Older browsers like
   1. World wide web browser
   2. Mosaic
   3. Internet explorer

Distributed Applications:

These are the applications that can be accessed over the internet, these applications will be running in some remote machines, there are mainly 2 types of Distributed applications

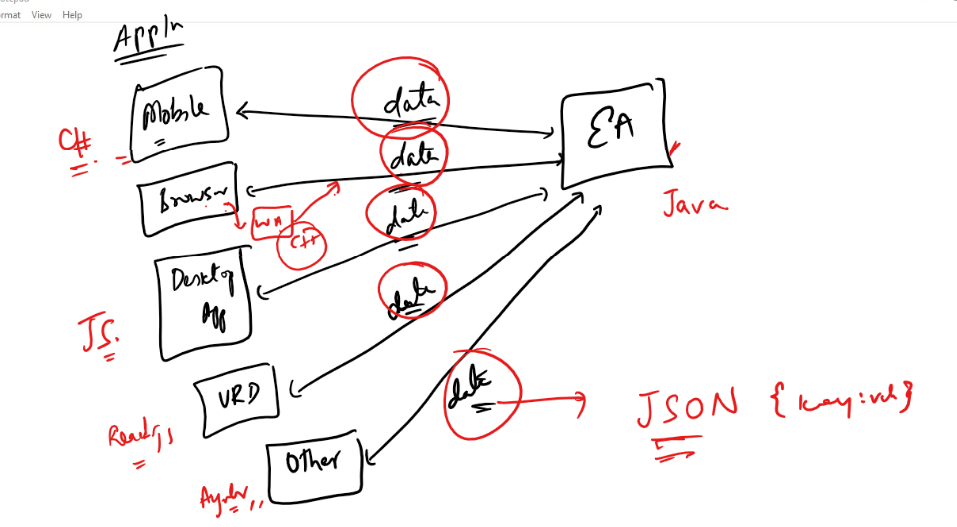
1. Web applications
2. Enterprise wide applications

Web application:

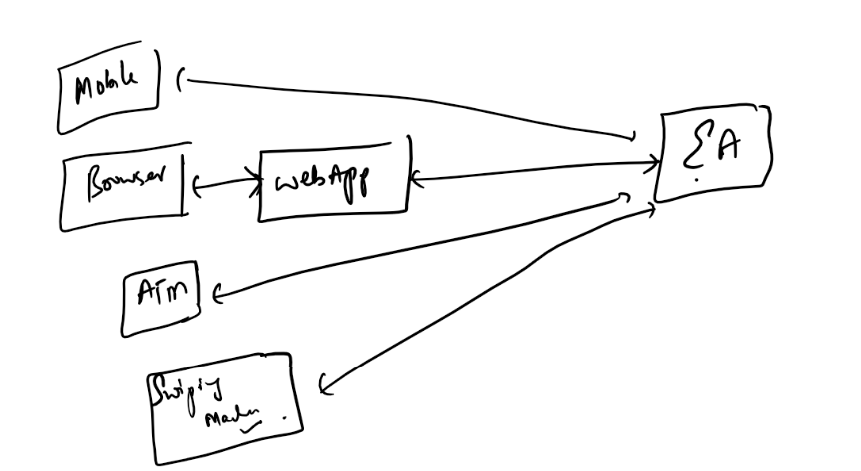
A web application takes request from the browser & returns the response in the format the browser understands i.e., HTML, CSS & Javascript

Enterprise application:

These are business related applications which can serve various types of applications written in different languages.



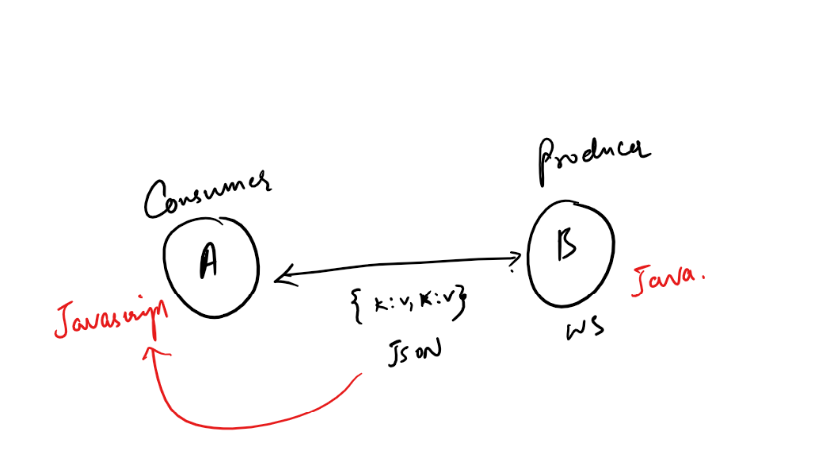
The enterprise application return data in a format which every application can convert to their respective types, suppose Java wants to share data to C#, then Java wouldn’t share data in Java instead it shares in one common format JSON that will be converted to C# by C# applications, same can be converted to Javascript by Javascript applications, same can be converted to Python by Python applications.



How does these enterprise applications share/exchange data to other applications

Enterprise applications will have webservices (ReST based Webservices) to share the data to any applications.

REST WebServices: These are online services or API’s which are made available on the internet so that any applications can use their URLs to consume the data, usually data will be in JSON format which is converted to the format the consuming application is written in



What are the common formats the webservices can use to share the data

1. JSON (More widely used): Easy to understand & maintain hence its used widely almost 99.9% of the time
2. XML
3. TEXT
4. HTML

Web applications: These applications will have web pages, there are 2 types of web pages

1. Static web pages: These are the pages which are having same contents for all the users, it doesn’t change but its common for all the users

ex: Wikipedia

1. Dynamic web pages: These are the pages whose contents are different for different users, these pages change their contents at runtime

ex: Gmail pages, Facebook pages and so on

Technologies/Languages used to create Static & Dynamic web pages

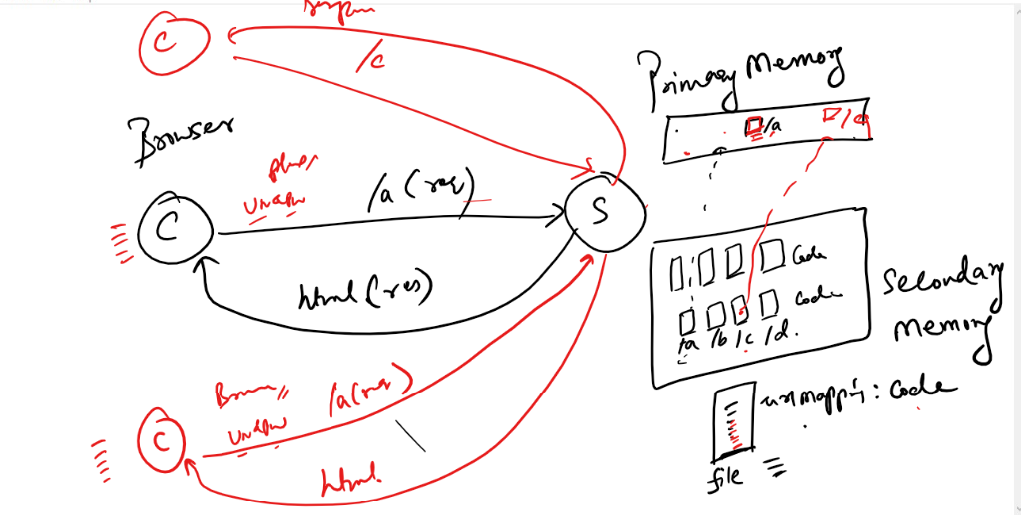
Static pages are created with below languages

1. HTML
2. CSS
3. Javascript

Dynamic pages are created with various technologies

1. C#: ASP .NET
2. Java: Servlets & JSP
3. Python: Django
4. Javascript: React.js, Angular.js, Ember.js

How the dynamic pages are created



Here when client sends the request to the server, it sends the URL to the server, the server is going first check in the primary memory whether any instance/entity/object can handle the request if the entity is not present then server checks in some configuration file to know which is the entity to handle the URL, it loads that entity to the primary memory & it executes the code to generate response, here response will be HTML because client is browser, however it can generate different HTML response for different clients for same URL.

ex: If you login to gmail then you see different page and others see different page.

What is the language browser understands

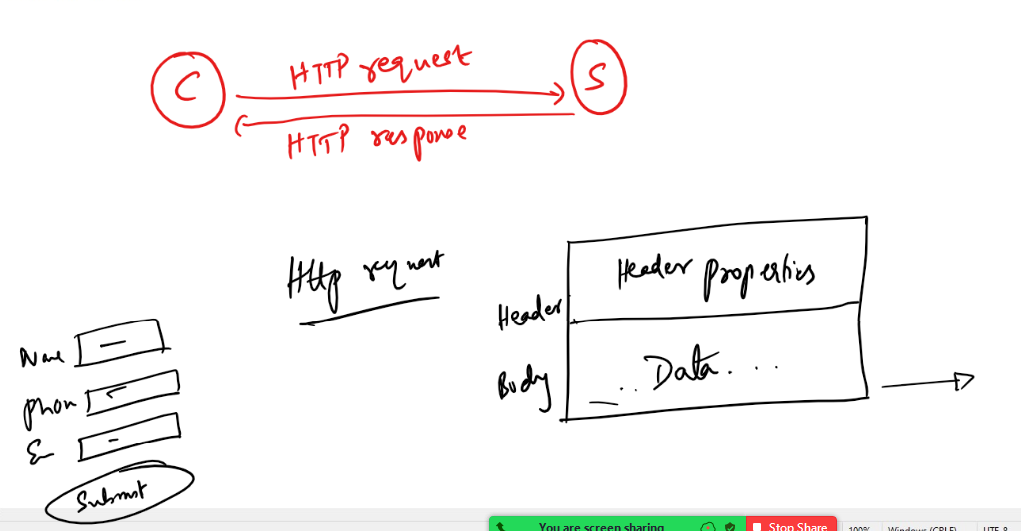
1. HTML
2. CSS
3. Javascript

What is the language client & server uses to communicate

HTTP: Stands for Hyper Text Transfer Protocol, it is the language client & server uses to communicate, they communicate using request & response, since they use HTTP they are also called as HTTP request & HTTP response.

Architecture of HTTP request & HTTP response

HTTP Request



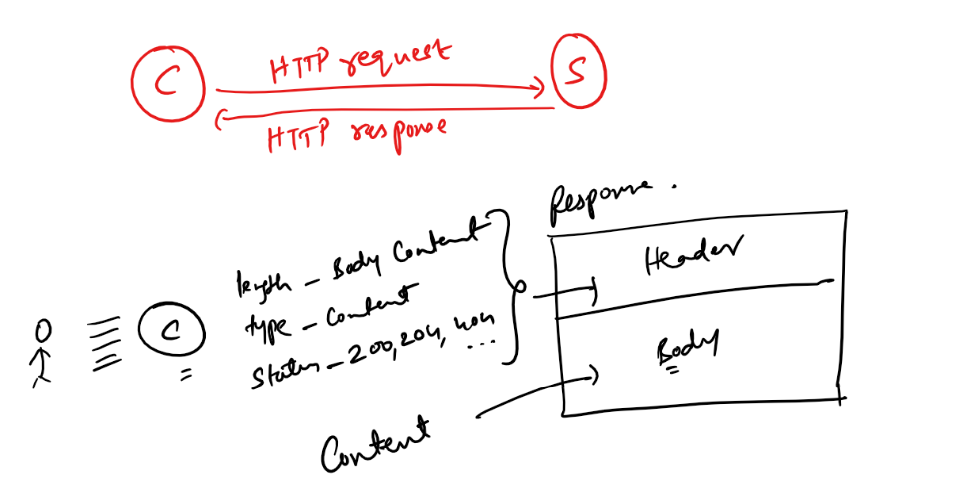
When client sends the request an HTTP request will be sent, the HTTP request will be divided into two sections

1. Header
2. Body

Header: It will have header properties like URL of the server resource, content type like text, json, xml and so on

Body: it will have content/data

HTTP Response



The Response will have 2 sections header & body

Header of the response will have header properties of response like content type, content length, time take to produce response, status codes like (200, 404, 405) and so on

Note: status code plays an important role for the client programs, because client programs can understand whether the request is succeeded or failed.

If status codes are in 2xx series then its success

If status codes are in 4xx series then its failed

HTTP status codes

These are some standard codes for client & servers used while communicating so that they can make decisions based on the status code

200: Ok

201: Created

404: Resource not found

401: Unauthorized access

400: Bad request