Java Backend Bootcamp. (Java bra ns)

The Big picture

Java Basics

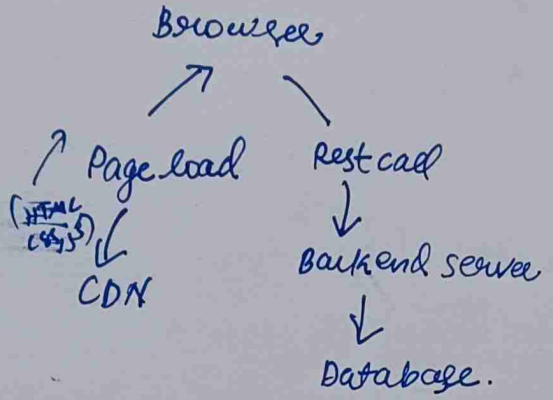
Java Collections

Advanced Java

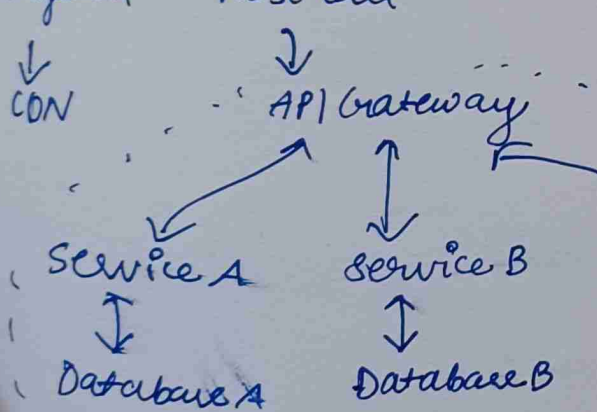
Maven

Three-tier architecture

→ Front-end
→ mid-tier
→ Database.



Microservices:



CDN → "content delivery network"

↓

Distributed servers that deliver web content to users based on their geographic location

CDN ← Origin server (main server)
Edge server (copy of origin server)

Web request:-

Browser → DNS server → corresponding IP address

DNS → "Domain name system"

Content to Browser

IP of closest CDN

↳ cache (found) → returns cached version

cache (not found) → sends request to origin server.

CDN Edge server

Web page:-

HTML page, CSS assets, images, JS assets, Fonts

Rest requests:-

* JS → constructs HTTP request.
* Adds endpoint API URL and HTTP method.
* Includes necessary parameters or data.
* Server gets request and returns response.
* Java Script receives response and processes it.
* Response manifests as a UI change.

Capable backend:-

* Intercept requests
* Extract data from requests
* process the request
* Pull necessary data from database
* Process the data and prepare response
* Return the response.

Handling requests:-

HTTP → HyperText Transfer protocol
↓
(message sent to client by server)

HTTP Request:-

* Request line ('GET /index.html HTTP/1.1')
* Headers : Meta info about the request
* Body : Data to be sent with the request.

HTTP Response:-

* Status line : First line Eg: 'HTTP/1.1 200 OK'
* Headers : Meta information about the response.
* Body : Actual data being sent.

HTTP:-
Stateless protocol.

Cookies → Allow multiple requests to be "tied together"
* Sent to the client in the set-cookie header of a response.
* sent back to the server in the cookie header of subsequent requests.

Git → helps to work with the file of many people at same time.

Git → manages versions of our code locally or on a remote services

Github → potential "main" remote ~~server~~ server.

Github:-

Pull request Workflow

* fork the repository.
* clone the repository.
* create a new branch.
* Make changes
* Push the changes.
* Create a pull request.
* Review and discuss
* Merge or decline.
* Update and delete the branch.

Three-tier Architecture:-

* Presentation Layer. - (user interface) HTML, CSS, JS.
* Application Logic Layer. - (Business Logic of the application)
* Data Layer. - (interface for the data stored in the database)

Front-end ⇒ user interface
back-end ⇒ application logic, calculations, databases and server-side tasks.