

Client \Rightarrow Browser / Appⁿ / Alexa etc.

Server (Backend)

\hookrightarrow m/c.

Module-2.

\downarrow
HTML
CSS
JS. \rightarrow frontend.
(UI/UX)

\downarrow
Backend.

Server Side Development.

Agenda.

\rightarrow Intro to Backend development.

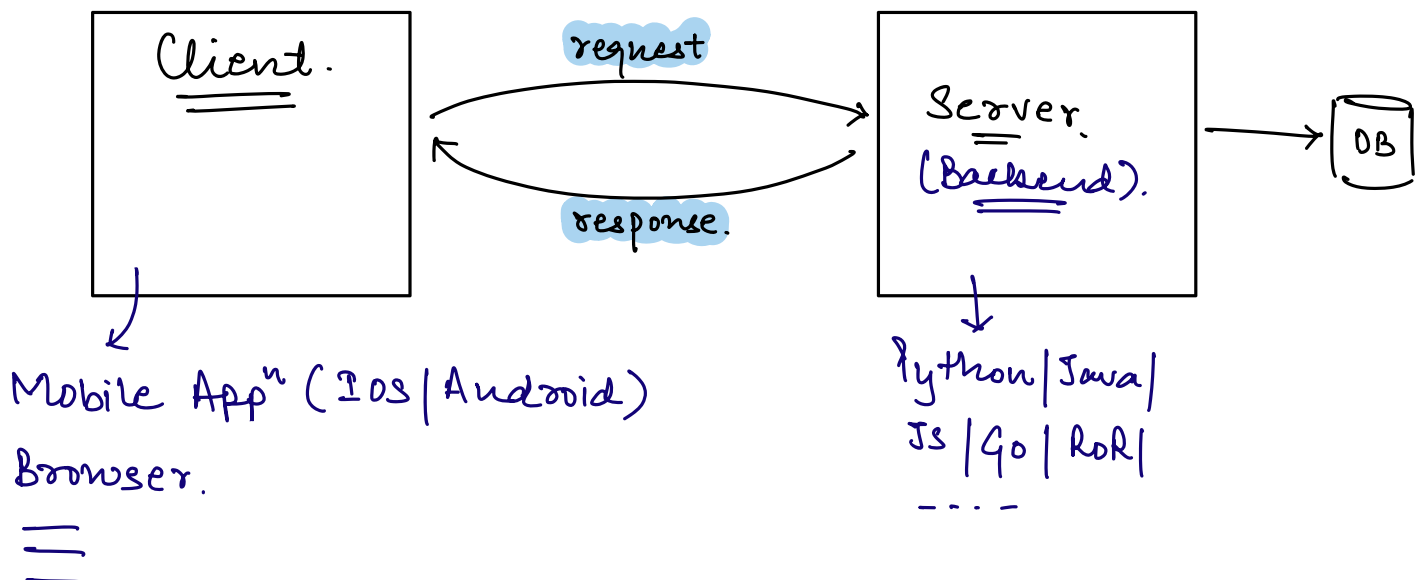
\rightarrow Client - Server Architecture.

\rightarrow APIs.

\hookrightarrow REST APIs.

\rightarrow JSON.

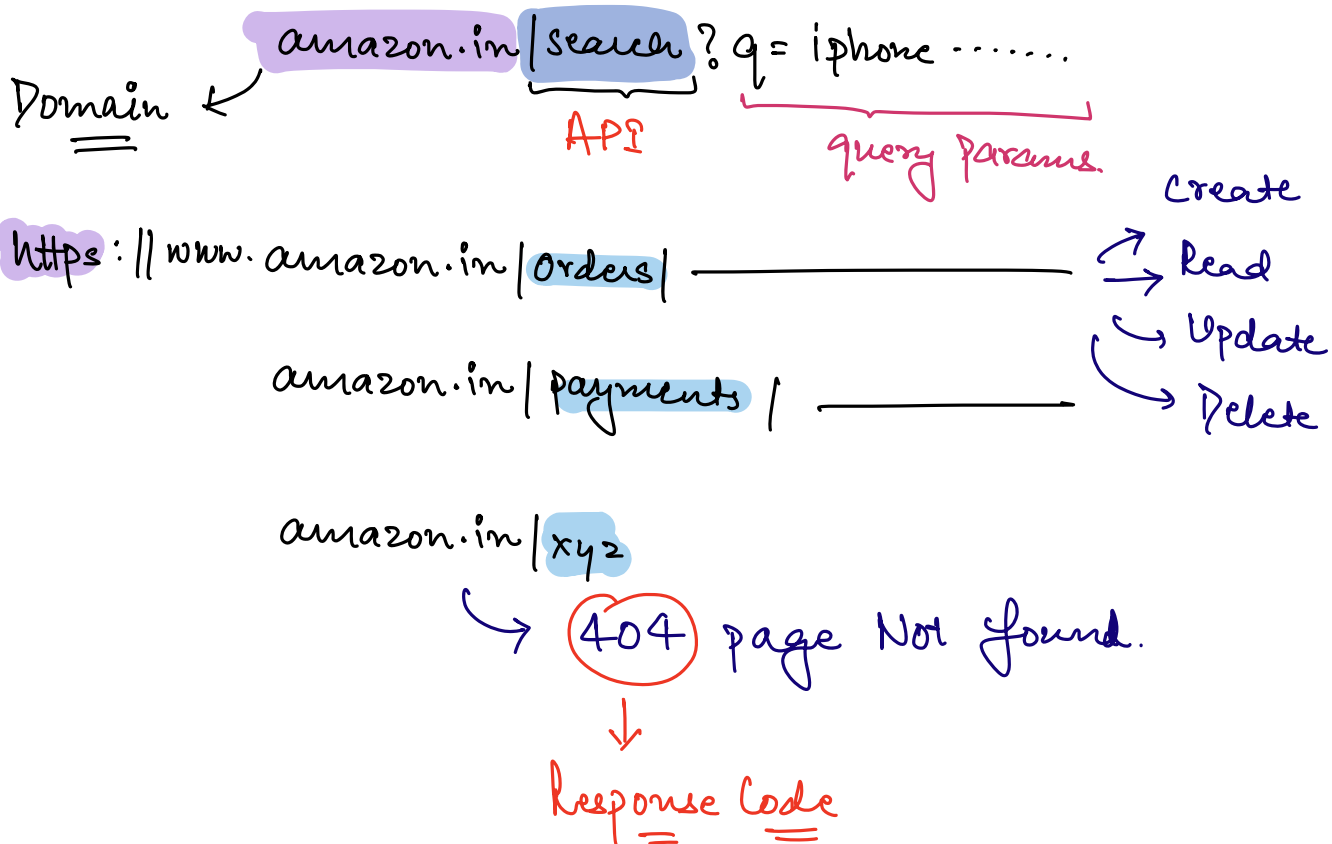
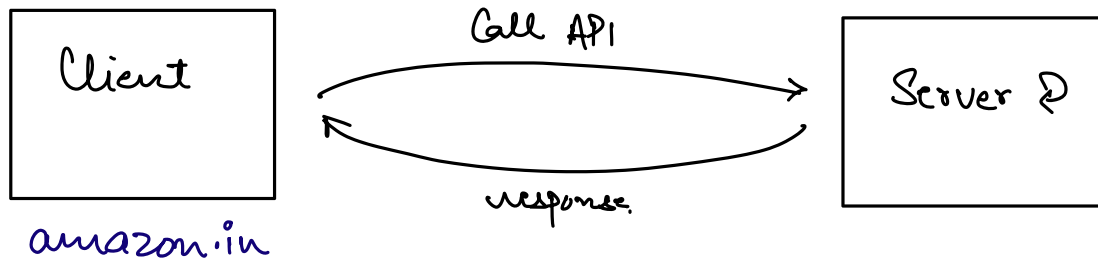
Client - Server Architecture.



Pagination

API.

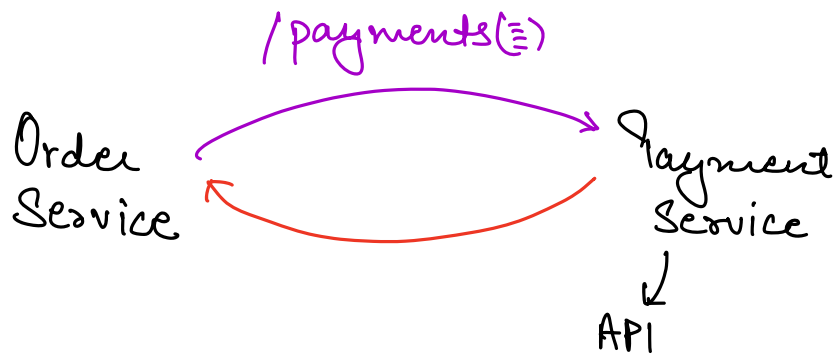
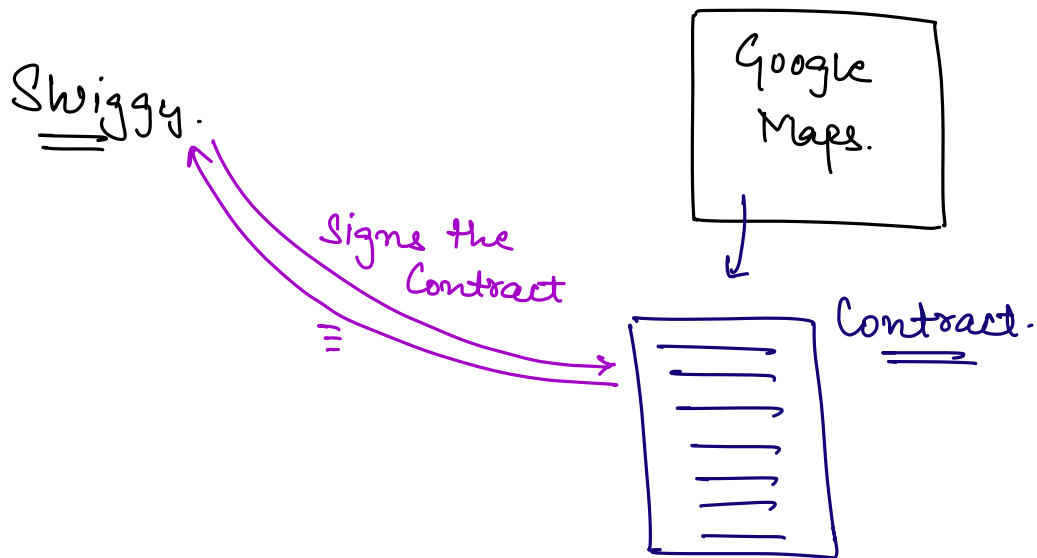
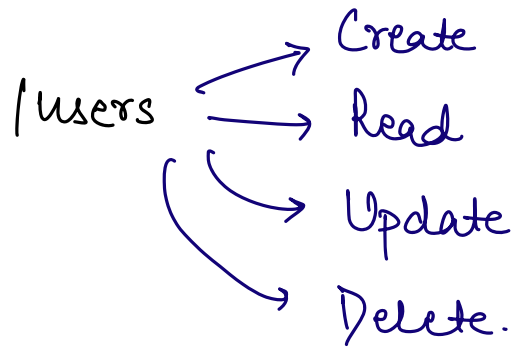
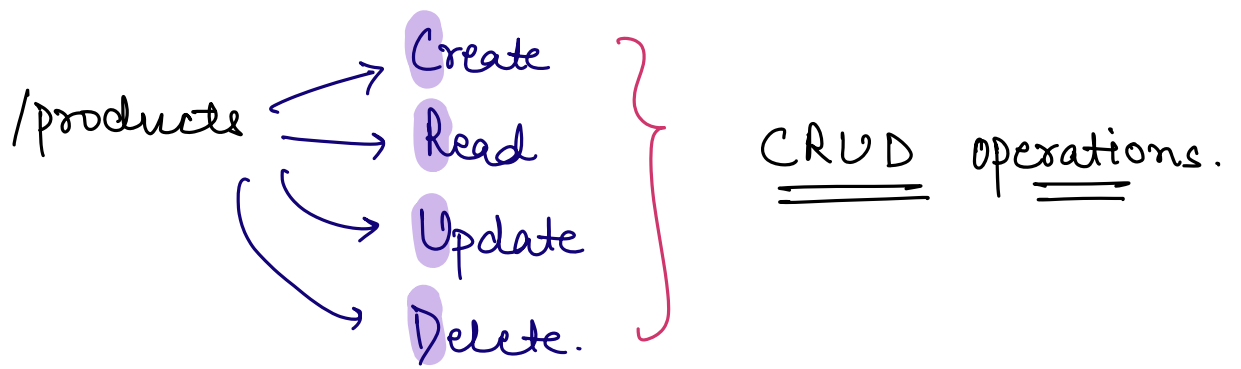
↳ Application Programming Interface.



200 → Success.

4xx → Client side Error Code.

5xx → Server side Error.



REST Principles.

→ Representational State Transfer.

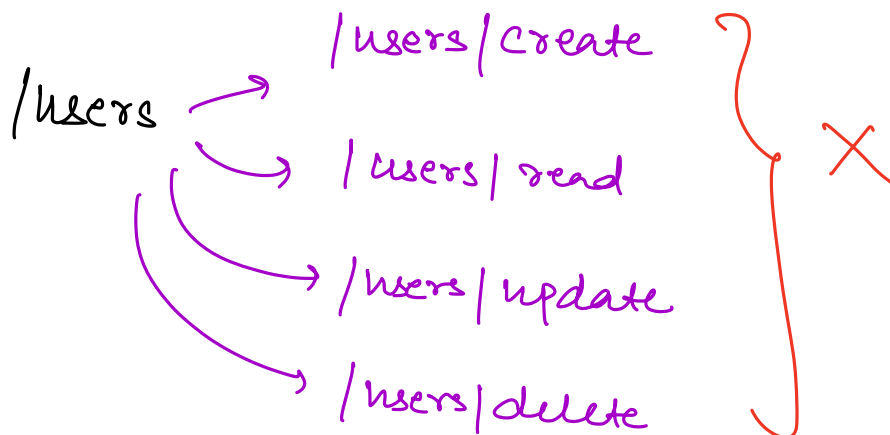
→ Guidelines on how to build API's.

⇒ APIs which follow REST principles are called as REST API's.

⇒ All the API's should be structured around the resources they work on.
Entities.

/orders

⇒ The type of action shouldn't be part of the API endpoint rather it should be defined by the HTTP method.



HTTP Methods.

GET : Read

POST : Create

PUT : Replace

PATCH : Partial Update

DELETE : ✓

GET /users

POST /users

PATCH /users

PUT /users

DELETE /users

⇒ follow HTTP methods.