

Client → Browser / Appⁿ | Alexa etc.

Server (Backend)
⇒ m/c.

Module-2.

HTML
CSS
JS.

→ frontend.
(UI/UX)

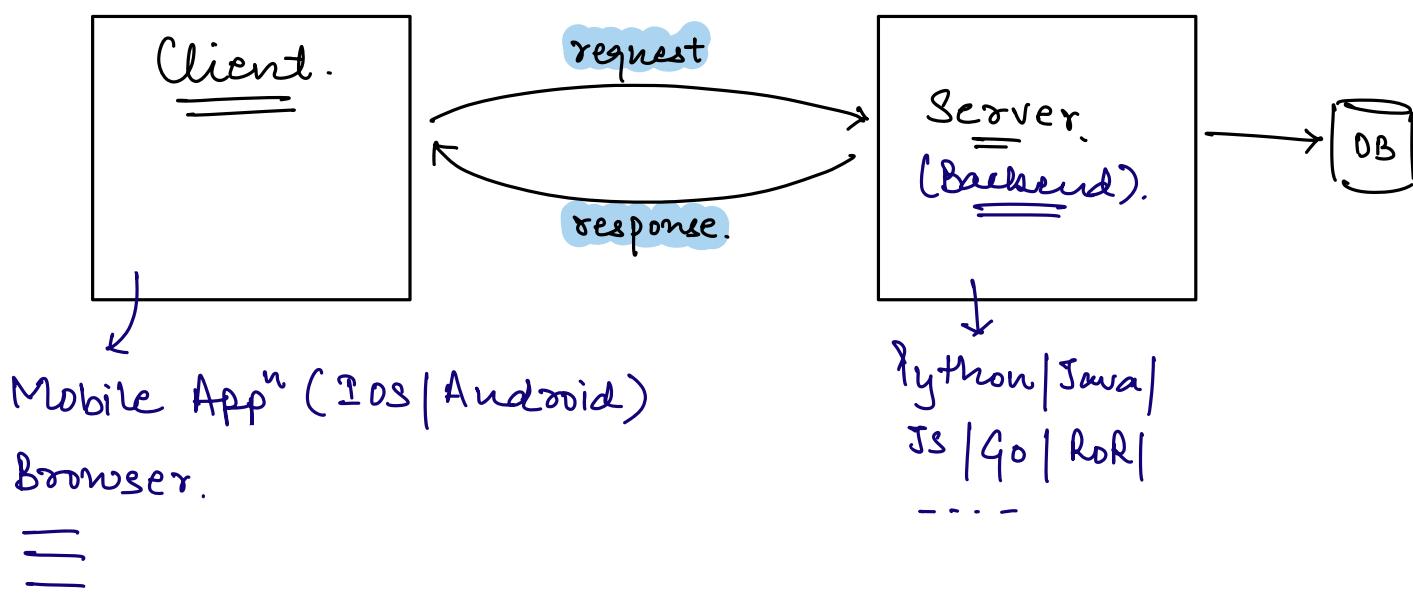
↓
Backend.

Server Side Development.

Agenda.

- Intro to Backend development.
- Client - Server Architecture.
- APIs.
 - ↳ REST APIs.
- JSON.

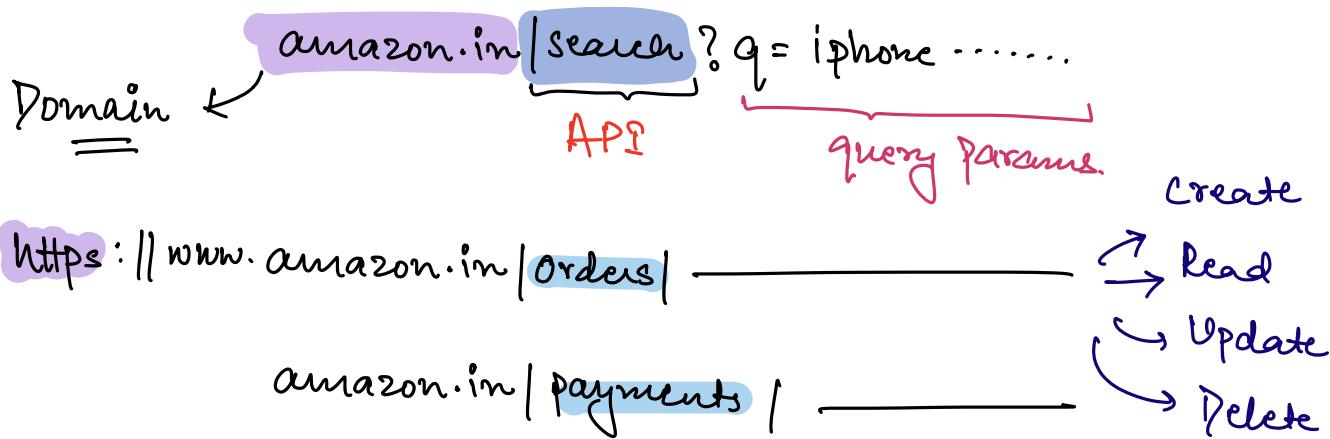
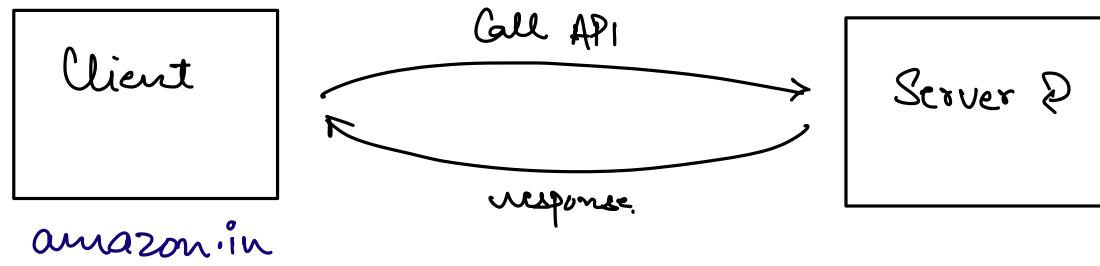
Client - Server Architecture.



Pagination

API.

↳ Application Programming Interface.

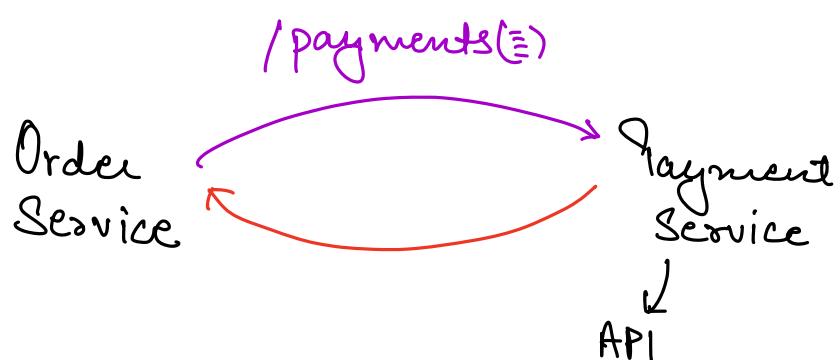
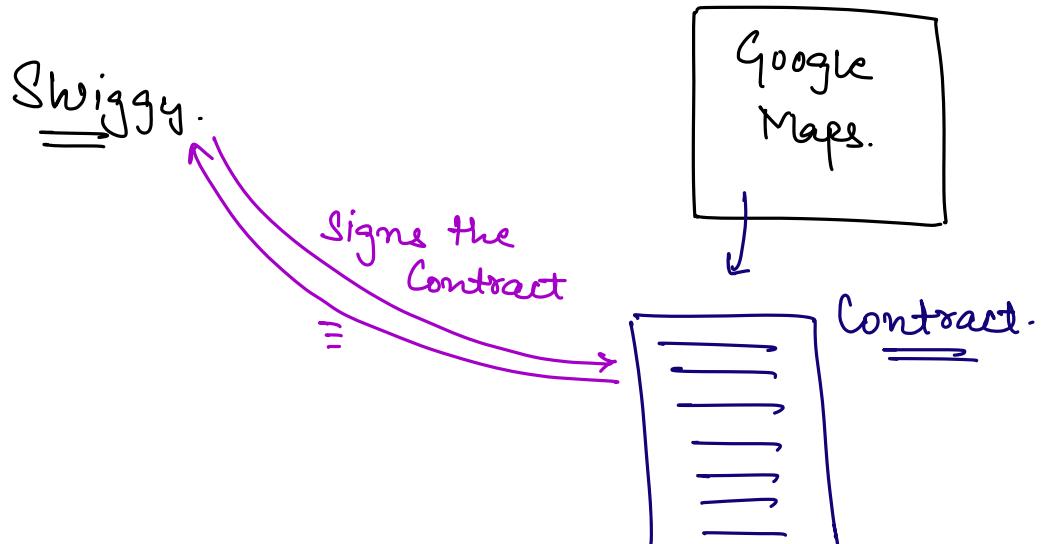
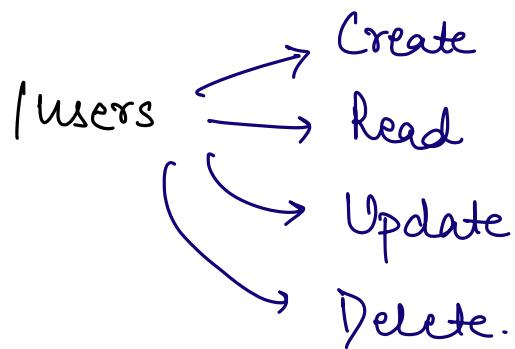
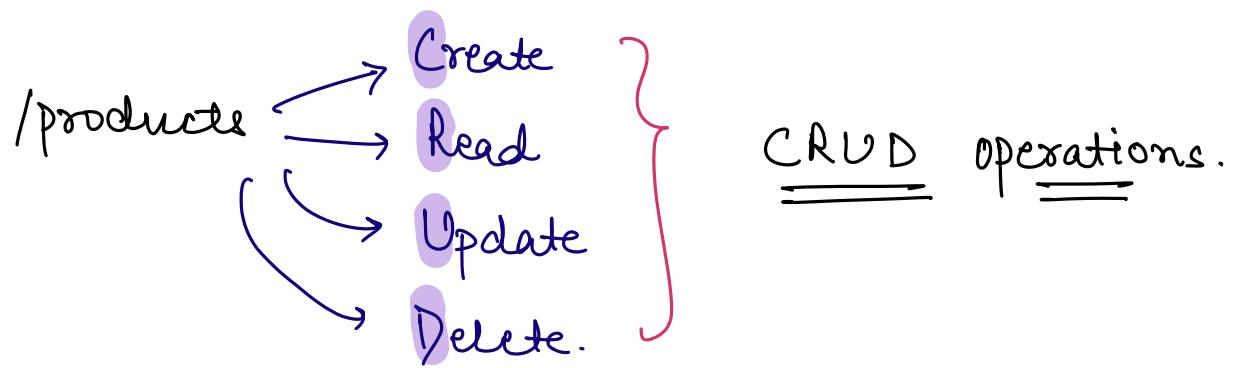


amazon.in/xyz
↳ **404** page Not found.
↓
Response Code

200 → Success.

4xx → Client side Error Code.

5xx → Server side Error.



REST Principles.

→ Representational State Transfer.

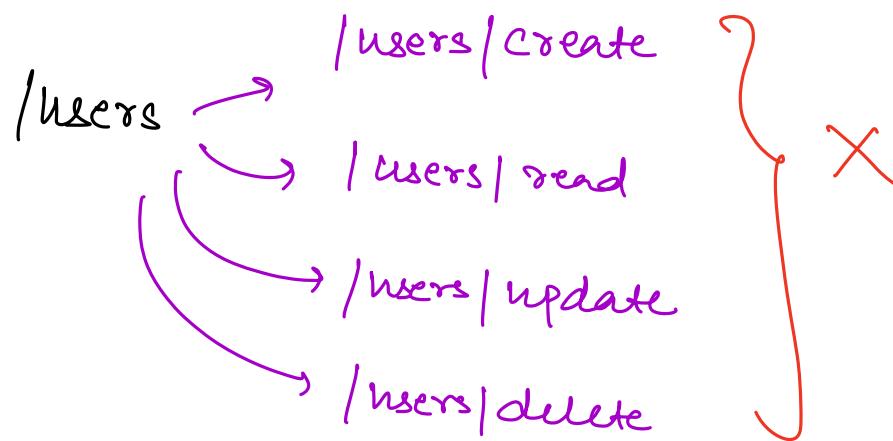
→ Guidelines on how to build API's.

⇒ APIs which follows 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 Methode.

GET : Read

POST : Create

PUT : Replace

PATCH : Partial Update

DELETE : ✓

GET /users

POST /users

PATCH /users

PUT /users

DELETE /users

⇒ follow http methode.