Agenda.

→ REST API Principles.

→ HTTP Methods.

→ Http Status Codes.

→ Synchronous (vs) Asynchronous Programming

REST APIs.

↳ Representation State Transfer.

/getOrder/1234 ✗ ⇒ GET /orders/1234

/users/create ✗ ⇒ POST /users

DELETE /orders/1234

2 API's on Order → 1st: Complete Order Object

→ 2nd: Only few details of Order Object

GET ~~/orders/1234~~ → 1st: /getCompleteOrderDetails/1234

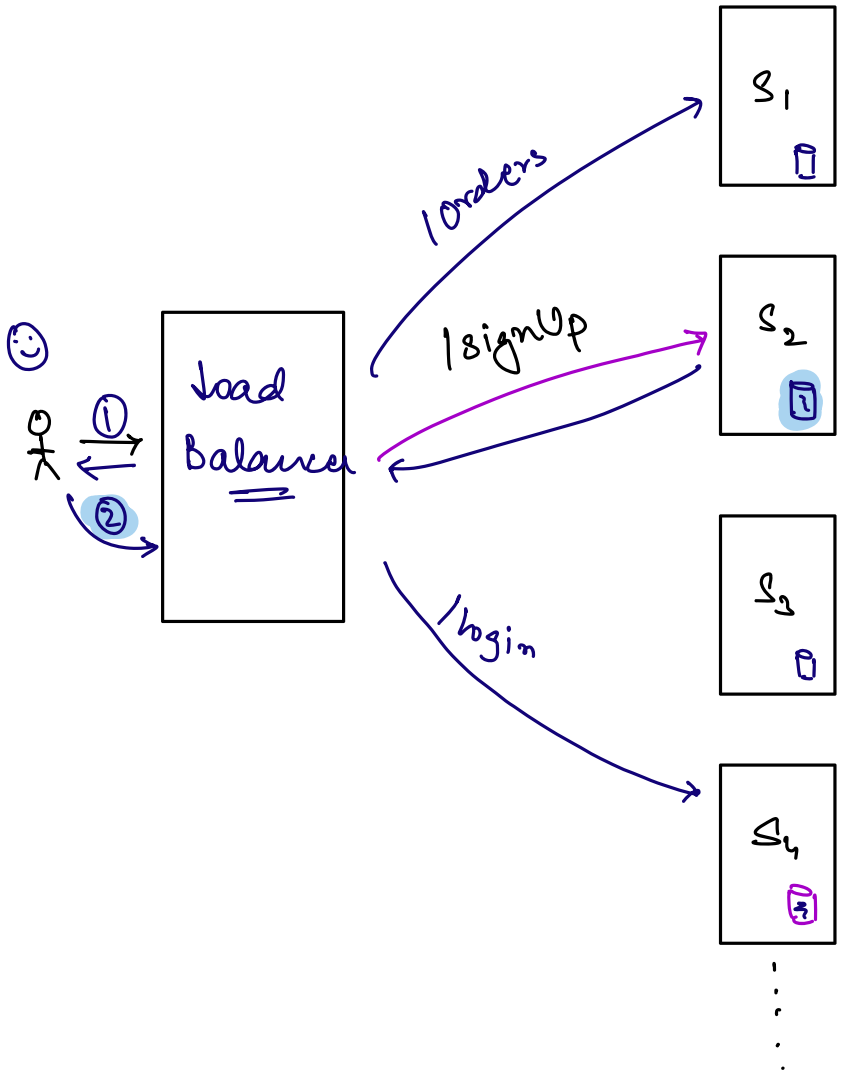
→ 2nd: /getPartialOrderDetails/1234

REST APIs should be stateless.

⇒ APIs should be independent.

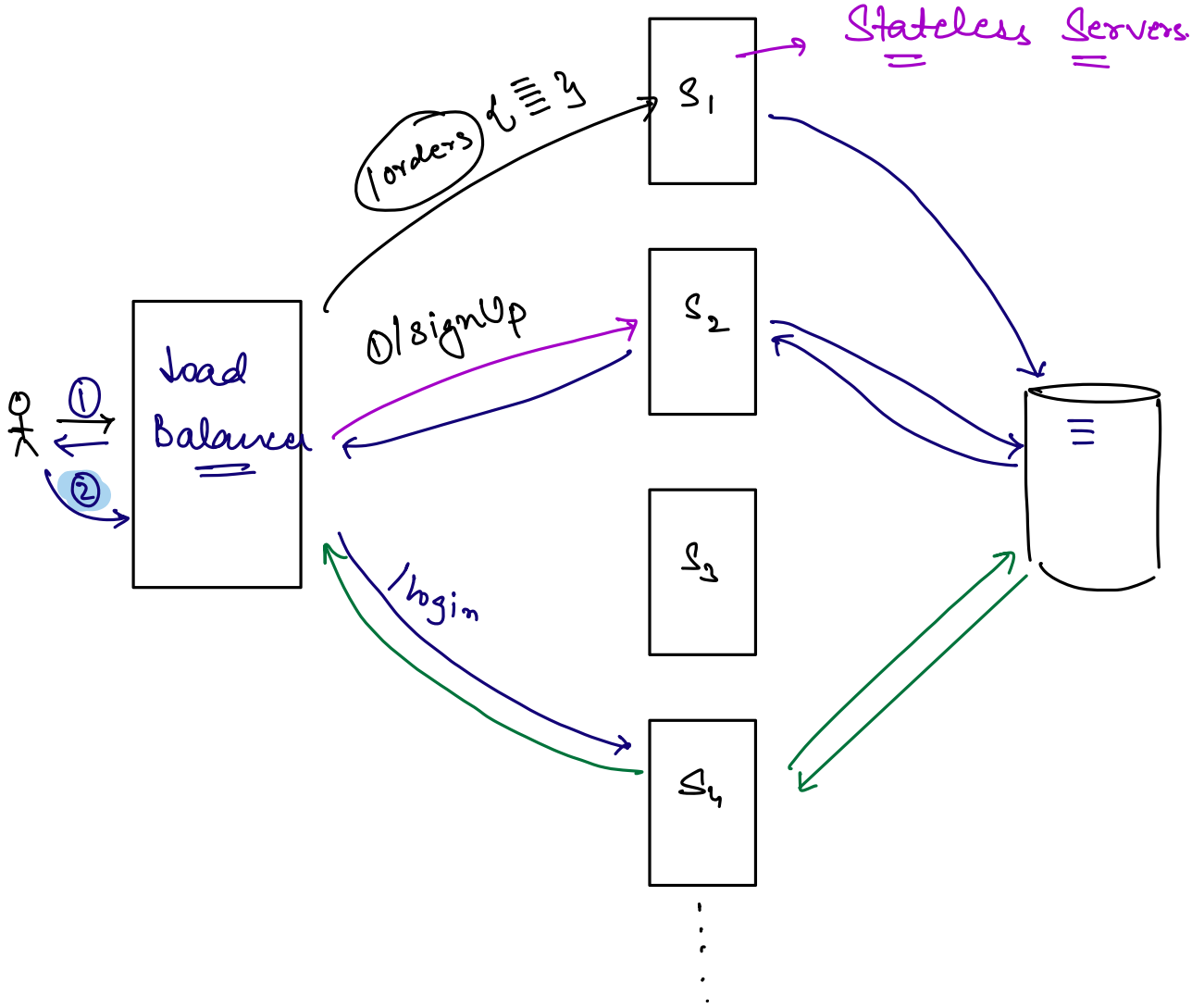
⇒ Each API should contain all the data / input whatever is required to execute the API.

Server ≡ m/c.



⇒ Stateful Servers.

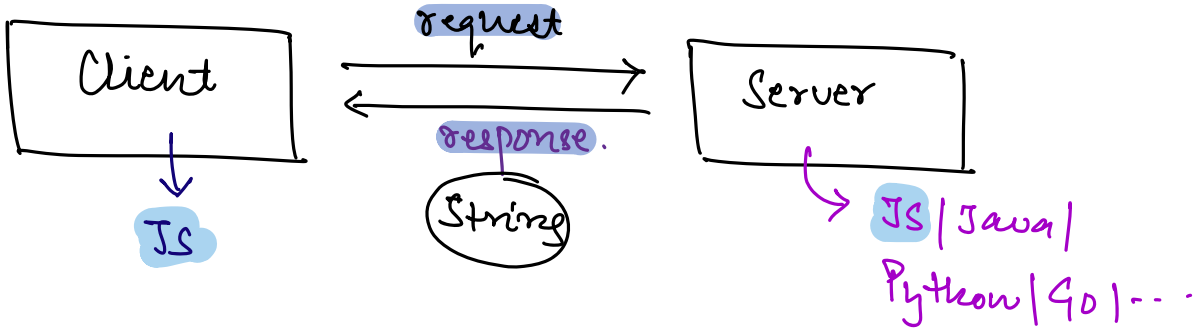
Data.



JSON

↳ Java Script Object Notation.

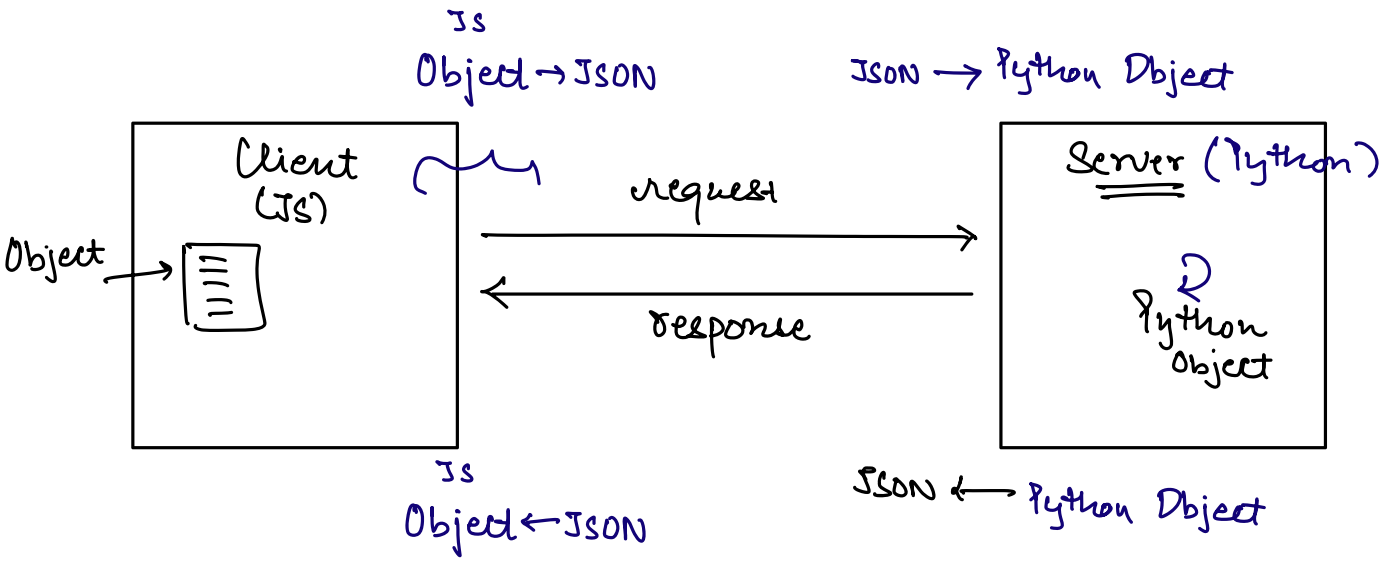
→ Common language based on which client & server can interact.



JSON : String. ⇒ Human readable Key-Value pair String.

```
"
{
    "id" : "1234",
    "description" : "iphone 14 pro",
    "price" : "150,000",
    "brand" : ——
              ——————
              ——————
              ——————
              ——————
}"
```

⇒ XML



Object → JSON : Serialization

→ Stringify

JSON → Object : DeSerialization

→ Parse

HTTP Methods.

GET : Read

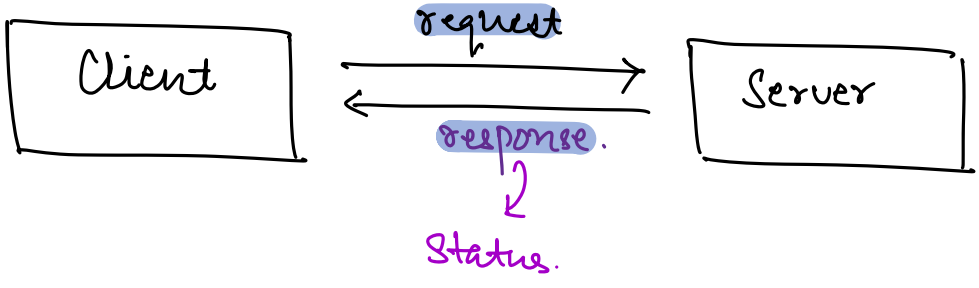
POST : Create

PUT : Replace

PATCH : Partial Update

DELETE : ✓

HTTP Status Code.



1xx : Informational Status Code

2xx : Successful.

- 200 : OK
- 201 : Created

3xx : Redirecting

4xx : Client Error.

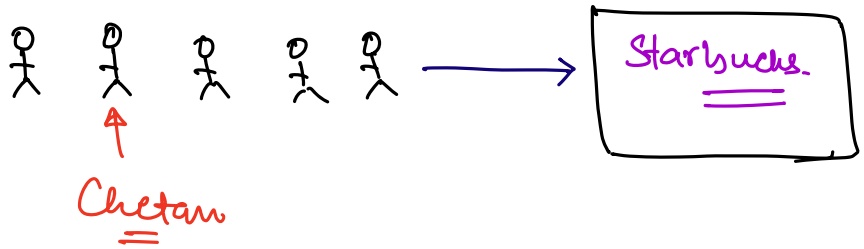
- 401 : Unauthorized
- 404 : Not found

5xx : Server Error.

- 500 : Internal Server Error
- 502 : Bad Gateway.

Synchronous (vs) ASynchronous Programming

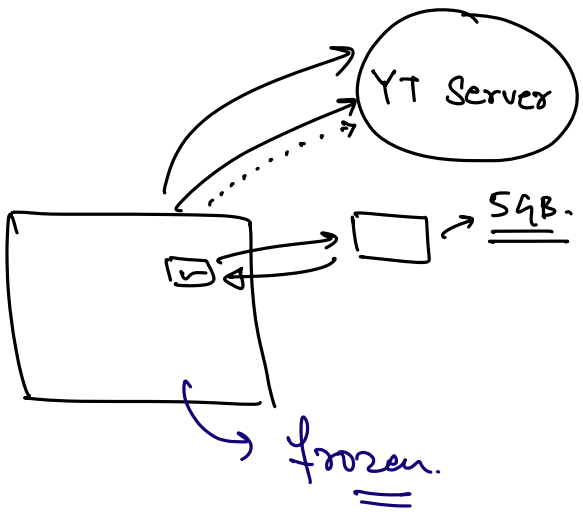
⇒ Coffee Shop.



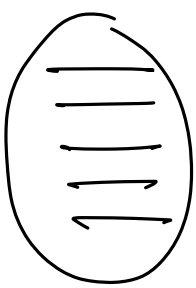
⇒ Synchronous.

↳ Blocking Execution

Youtube.



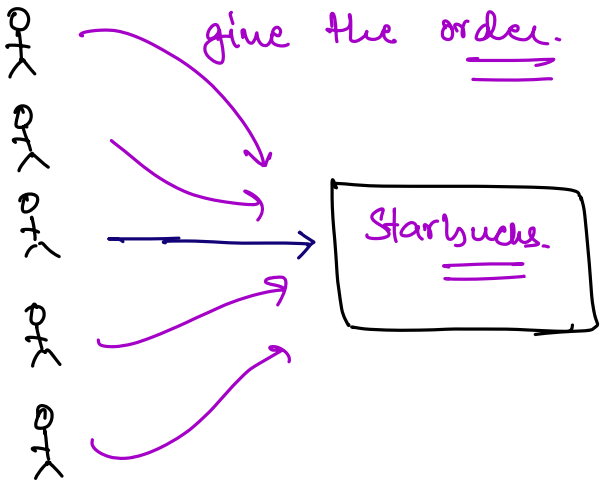
Heavy task.



Asynchronous.

↓

Non Blocking.



SetTimeOut → H/W.

(Read about SetTimeOut fun^)