JWT AUTH

- Used to identify authorized users.

Q. What is JWT Web Token?

A.
- **Open Standard:** means anywhere, anytime & anyone can use JWT.

- Secure data transfer

- It's digitally Signed: Info. is verified & trusted.

- There is no alteration of data.

- **Compact:** because JWT can be sent via URL, post req. & HTTP header

- Fast transmission makes JWT more usable.

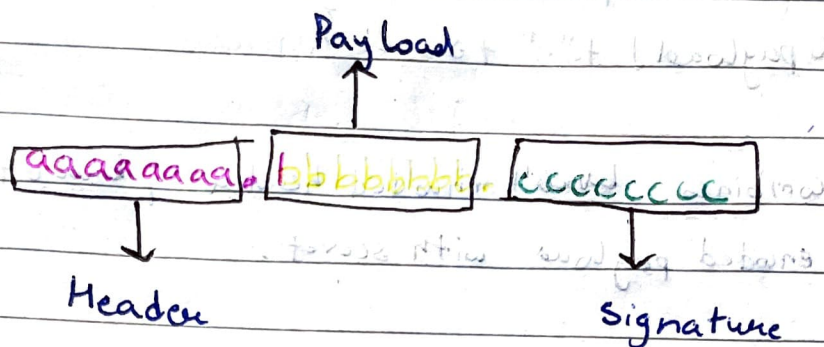- **Self Contained:** because JWT itself holds user info.

- Avoids querying Db more than one.

- JWT is Useful for :-

↳ Authentication   ↳ Secure data transfer

- JWT Token struct

Payload ↑

| aaaaaaaa | bbbbbbbb | cccccccc |

         Header   ↓   Signature ↓

1o Header

↳

    contains algo. like RSA / HMACSHA 256 & info.
    type of token

    {

        "alg" : "" Algo. Like RSA / HMA
        "Type" : "" Type of JWT

    }

2. Pay load

↳

    login / user credintals

    {

        "loginname" : "Ray"
        "password" : "123#"
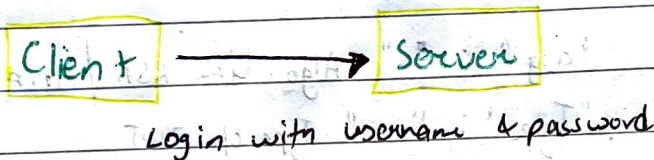
    }

3. Signature

{ base 64 urlencoded (header) + "." + base 64 urlencoded
(payload) + "." + Secret }

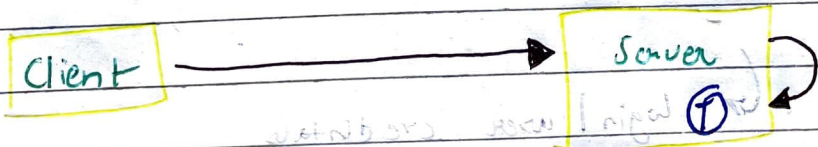- Combines base64 encoded header, base64
  encoded payload with secret.
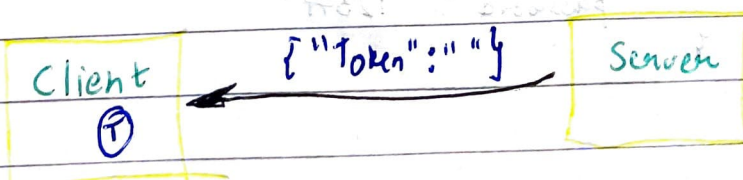
- More security.

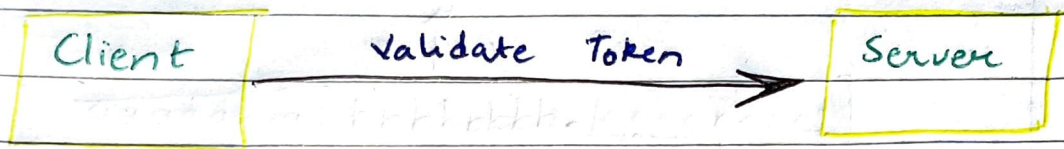**Q.** How does JWT?

A. **Step 1 :** Client logs in with crendientials

Client ──────➤ Server

Login with username & password

Step 2 : Server generates a JWT token at server side

Client ──────────────────➤ Server ⟲ ①

Step 3 : Server returns a token in response

Client ⓣ ◄────── { "Token":" " } Server

Step 4 : Client sends a copy of the token to validate the token

Client	Validate Token ⟶	Server

Step 5 : server checks it valid /not

Client		Server	⟶
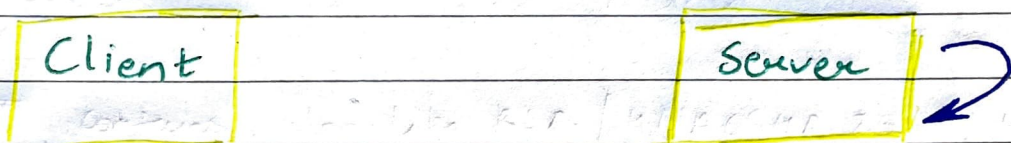
Step 6 : Sends a msg to client

Client	⟵ { "Message" : "sucess" }	Server