Authentication and Authorization

Authentication

Usually, authentication by a server entails the use of a username and password. Other ways to authenticate can be through cards, retina scans, voice recognition, and fingerprints. Registers in the system earlier or is registered by someone else (system admin)

● When you enter your ATM card into the ATM machine, the machine asks you to enter your pin. After you enter the pin correctly, the bank then confirms your identity that the card really belongs to you and you're the rightful owner of the card. By validating your ATM card pin, the bank actually verifies your identity, which is called authentication.

● Students of a particular university are required to authenticate themselves before accessing the student link of the university's official website. This is authentication.

● One example of a multi-factor authentication supporting online service is that of PayPal. They currently offer at least two different multi-factor options. One option involves a credit card-sized device that produces on-demand a one-time-use six-digit PIN. The second option sends an SMS text message to your cell phone with a six-digit PIN. In either case, the PIN is used alongside your name and password credentials to gain access to your PayPal account.

Authorization (access control)

● The process of verifying and confirming employees ID and passwords in an organization is called authentication, but determining which employee has access to which floor is called authorization. Let's say you are traveling and you're about to board a flight. When you show your ticket and some identification before checking in, you receive a boarding pass which confirms that the airport authority has authenticated your identity. But that's not it. A flight attendant must authorize you to board the flight you're supposed to be flying on, allowing you access to the inside of the plane and its resources.

● Authorization determines exactly what information the students are authorized to access on the university website after successful authentication.

Authentication vs Authorization

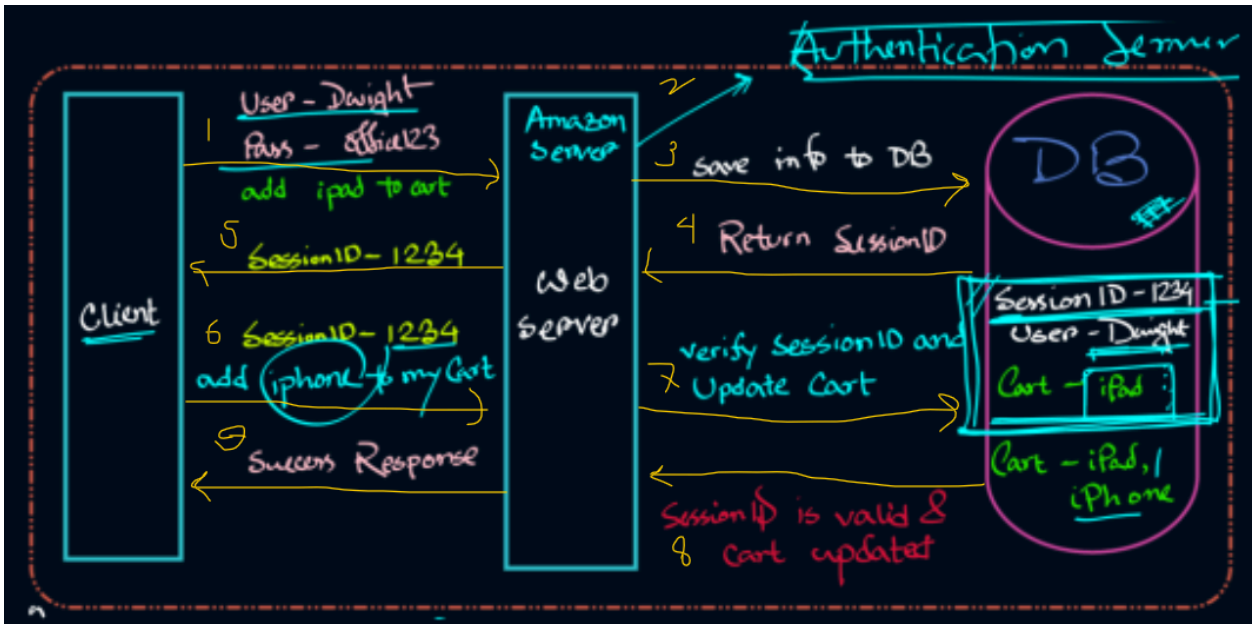
If authentication is who you are, authorization is what you can access and modify. In simple terms, authentication is determining whether someone is who he claims to be. Authorization, on the other hand, is determining his rights to access resources.

Factors of Authentication

Session Based Authentication

Steps to create a session between client and web server:

1. client sends userId and pass to the web server to authenticate
2. web server receives it & pass it to the auth server to cross check whether the person exists or not.
3. web server will store an entry in the database. it'll contain the session id, user info etc
4. DB will send the session id back to the web server, and web server sends back it to the client.
5. after that, whenevr clients sends requests, it'll pass the session id as well
6. DB verify the id and sends the success respond back to the web server



** SessionID is stored in cookies

Limitations of Session Based Authentication

1. Distributed System
2. Performance Issues
3. Cookie Fraud
4. Cannot set cookie for cross domain: It poses issues when APIs are served from a different domain to mobile and web devices.

Browser Storage

Local and Session Storage:

https://medium.com/@shahbazmehmood61/exploring-browser-storage-options-a-comprehensive-guide-012d58a655a7

Cookie:

https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
Session Cookie: Validity for a session only. Can change the duration of session
Permanent Cookie: Validity→ Infinite
3rd Party Cookie: kono kichu store hoy na, just history related info thake, no personal info. ager browsing history er upor base kore ekta refined result show korbe

Difference

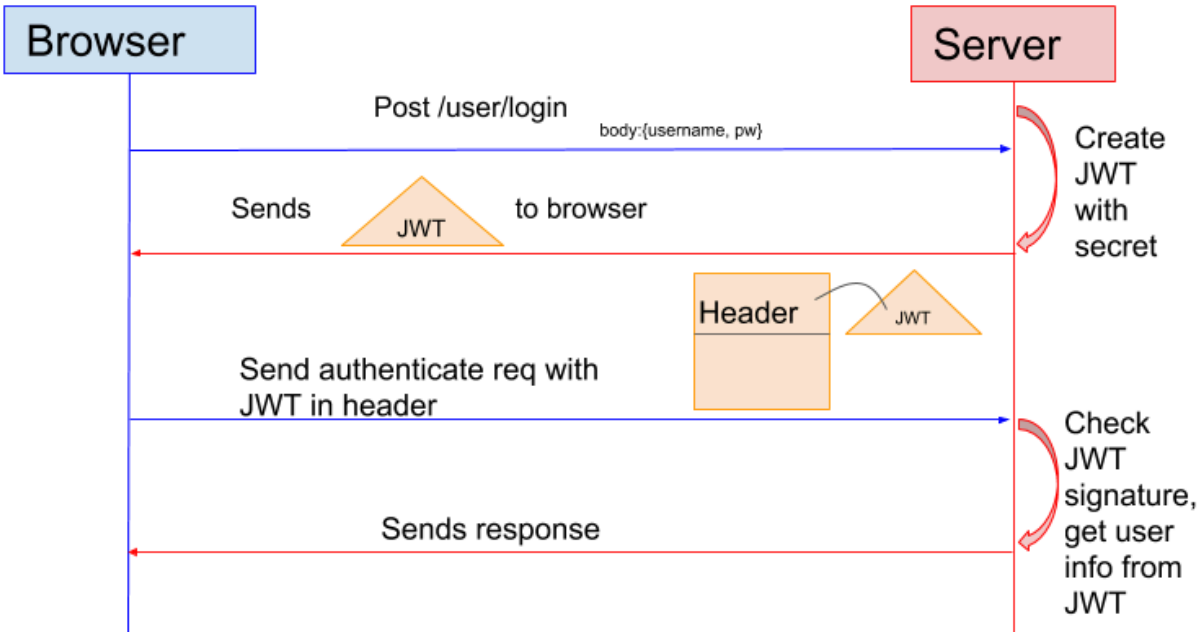
Points	Local	Session	Cookie
Storage Limit	5 MB	5-10 MB	4KB
Accessibility	Both Client & Server can access	Both Client & Server can access	Server can, but if HTTPOnly is active, then Client can access it
Validity	Lifetime	Browser/Tab close	Depends on the type
HTTP request	Not sent with HTTP requests, so it's good for storing data that doesn't need to be transferred to the server	Not sent with HTTP requests, so it's good for storing data that doesn't need to be transferred to the server	Automatically sent with each HTTP request to the server, making them useful for tracking sessions or managing authentication.
Example	Saving user preferences like theme settings or cart items that should persist even after the browser is closed.	Keeping form data or user selections that should only persist until the user navigates away or closes the browser tab.	Storing user login tokens, session IDs, or tracking information that must be sent to the server with every request.

Token Based Authentication

[The Ins and Outs of Token-Based Authentication]

[SessionID store kori cookie te, but JWT store kori local storage e keno?]
SessionID store kortam cookie te, jehetu size choto chilo, 4KB, header er satheo jodi pathai tao pera hobe na temon. Kintu JWT store kori local storage e, bcz eta continuously data store kortei thake. Tai cookie te store kora feasible na. Size limit exceed korte pare.

JWT encrypt korar jonno use kora hoy RSA/HMAC-SHA256
RSA: public key crypto graphy. rsa creates a signature based on a public and private key.
HMAC-sha256: a keyed hash function that combines a secret key with the input message before applying the SHA-256 hash function. This provides an additional layer of security by ensuring that the integrity and authenticity of the message can only be verified by someone who has access to the secret key. HMAC-SHA256 is commonly used for message authentication and integrity checking in various protocols and applications, including digital signatures, VPNs, and secure messaging.



Steps in gist:
○ User Requests Access with Username / Password
○ Application validates credentials
○ Application provides a signed token/JWT with secret to the client
○ Client stores that token and sends it along with every HTTP request
○ Server verifies token and responds with data. Here, only the server that has that secret key can decrypt the JWT

JWT Structure