

1. What are middleware functions in Express.js, and how do they work?

Ans: In Express.js, middleware functions are special functions that handle tasks like modifying requests, sending responses, or moving to the next function in line. When a request comes in, these functions are executed in the order they are defined. They can perform tasks like logging, authentication, or data parsing. If a middleware function doesn't finish the request, it calls `next()` to pass control to the next middleware, ensuring the request is processed step by step until a response is sent.

2. What is JWT, and how does it work?

Ans: JWT (JSON Web Token) is a secure way to transmit information between a client and a server as a small, encoded string. It contains three parts: a header (which specifies the type of token and algorithm), a payload (which holds the data, like user information), and a signature (which ensures the token hasn't been altered). When a user logs in, the server creates a JWT and sends it to the client, who then includes it in requests to access protected resources. The server checks the JWT to verify the user's identity and grant access.

3. How do you securely store JWT on the client-side?

Ans : To securely store a JWT on the client side, the best approach is to use an HTTP-only cookie so that JavaScript can't access it, which helps protect against attacks.

4. How does token expiration work in JWT?

Ans: In JWT, token expiration is managed using the `exp` (expiration) claim in the token's payload. This claim specifies the exact time (in Unix timestamp format) when the token will expire. Once the expiration time is reached, the token becomes invalid, and the server will reject any requests made with it.