

***Q) How next() works in middlewares?***

***Ans.*** Yes, that's correct! When `next()` is executed in a middleware, it signals to Express that the current middleware has completed its processing and that Express should continue processing the request by moving on to the next middleware in the chain. If there are no more middlewares to process, Express then uses its internal routing table to match the request to the appropriate route handler based on the HTTP method and URL.

Here's a step-by-step breakdown of how the process works:

1. Request is received by Express.
2. Express begins processing the request through the defined middlewares in the order they were added using `app.use()` or `app.METHOD()`, where METHOD can be `get`, `post`, `put`, etc.
3. When a middleware calls `next()`, control is passed to the next middleware in the chain. If there are no more middlewares to execute, or if a middleware does not call `next()`, the process moves to the next step.
4. After all middlewares have been executed, Express uses its internal routing table to match the request's HTTP method and URL to the appropriate route handler.
5. Once the matching route handler is found, Express executes that route handler function, passing in the `req` (request) and `res` (response) objects.
6. The route handler processes the request, performs any necessary actions, and sends a response back to the client.

So, to summarize, the middlewares are executed first, and if `next()` is called in each middleware, it allows the request to move through the middleware chain. After all middlewares are executed, Express uses its routing mechanism to find the correct route handler, and then the logic within the route handler is executed.