**MIDDLEWARES**

* **Definition:** Functions that process and modify incoming HTTP requests or outgoing responses (**Any function that is between your request and response which will help you to actually handle your request in a better way)**
* **Purpose:** Perform tasks like logging, authentication, data parsing, and error handling.
* **Access:** Have access to the request and response objects.
* **Flow Control:** Can terminate the request-response cycle or pass control to the next middleware using **next**.
* **Bridge:** Acts as a bridge between incoming requests and final responses.
* **Custom Functionality:** Allows developers to inject custom functionality into the request-response lifecycle.
* **Organization:** Helps organize and handle different aspects of web applications.

|  |
| --- |
| **Request → Middleware 1 → Middleware 2 → Route Handler → Response**  **↓ ↓**  **(next) (next)**  **↓ ↓**  **... ...** |

Types of Middleware

* Application-Level Middleware
* Router-Level Middleware
* Error-Handling Middleware
* Built-in Middleware
* Third-Party Middleware
* Session Middleware
* Logging Middleware
* Authentication Middleware
* Authorization Middleware
* CORS Middleware
* Custom Middleware
* Multer Middleware