

Middlewares

NodeJS

What are Middlewares?

- Middleware functions are code executed between the server receiving a request and the final route handler sending a response.
- They form a pipeline or "chain" that the request travels through.

Example flow of execution:

Request → Middleware 1 (Logger) → Middleware 2 (Authentication) → Final Route Handler → Response

Functions of Middleware

- Run code (e.g., logging, data processing)
- References (Modify the request (req) and response (res) objects)
- Response (End the request/response cycle by sending a response)
- Release (Call next() to pass control to the next middleware in the stack)

Common Use Cases

Authentication/Authorization

Checking if a user is logged in or has permissions.

Request Logging

Recording details about every incoming request.

Data Validation

Ensuring incoming data meets required criteria (e.g., sanitizing inputs).

Error Handling

Catching and centralizing errors across the application.

Creating Custom Middleware

```
function customMiddleware() {  
  return (req, res, next) => {  
    ...place logic of middleware here  
    next();  
  };  
};  
  
module.exports = {  
  customMiddleware,  
};
```

Using Middleware

Application-Level Middleware

- Executed for every request that the server processes

Route-Level Middleware

- Executed for every request to a particular route; or
- Executed for every request that matches a route

Router-Level Middleware

- Executed for every request to a specific router

Application-Level Middleware

```
const app = express();  
  
app.use((req, res, next) => { ... logic here });  
app.use(customMiddleware());  
  
app.get("/", (req, res) => { ... });  
app.get("/:id", (req, res) => { ... });  
app.post("/", (req, res) => { ... });  
app.post("/:id", (req, res) => { ... });
```

Route-Level Middleware

```
const app = express();
```

```
app.get("/", (req, res, next) => {  
  ... middleware logic here  
  next();  
});
```

```
app.get("/", customMiddleware(), (req, res) => { ... });  
app.get("/:id", (req, res, next) => {  
  ... middleware logic here  
  next();  
}, (req, res) => { ... });
```


Router-Level Middleware

```
const router = express.Router();

router.use((req, res, next) => { ... logic here });
router.use(customMiddleware());

router.get("/", (req, res) => { ... });
router.get("/:id", (req, res) => { ... });
router.post("/", (req, res) => { ... });
router.post("/:id", (req, res) => { ... });

module.exports = router;
```