

# Middleware

---

- Middleware applies before routes.
- Any thing we want to perform before routes are to be done using middleware.
- jab bhi server request accept krta hai waha se tab rout tk pahuchne se pahle bich me hi rok ke agr kuch perform krna chhate ho to ye element middleware keh;lata hai.
- Whenever the server accepts a request, and you want to stop it in the middle to do something before it reaches the route, that part is called middleware.

## # Ways to Use Middleware in Node.js

In Node.js, middleware can be used in several ways depending on the framework and context. Here are the main approaches:

### 1. Application-Level Middleware

Applied to the entire application, executes for every request:

```
app.use((req, res, next) => {
  console.log("Time:", Date.now());
  next();
});
```

### 2. Router-Level Middleware

Applied to specific routers:

```
const router = express.Router();
router.use((req, res, next) => {
  console.log("Router middleware");
  next();
});
```

### 3. Route-Specific Middleware

Applied to individual routes:

```
app.get("/user", authMiddleware, (req, res) => {
  res.send("User page");
});
```

## 4. Error-Handling Middleware

Special middleware with four parameters for handling errors:

```
app.use((err, req, res, next) => {
  console.error(err.stack);
  res.status(500).send("Something broke!");
});
```

## 5. Third-Party Middleware

Using external middleware packages:

```
const morgan = require("morgan");
app.use(morgan("dev"));
```

## 6. Multiple Middleware in Sequence

Chaining multiple middleware functions:

```
app.get("/user", [auth, validate, sanitize], controller);
```

## 7. Conditional Middleware

Applying middleware based on conditions:

```
app.use((req, res, next) => {
  if (req.path.startsWith("/api")) {
    return apiMiddleware(req, res, next);
  }
  next();
});
```

## 8. Path-Specific Middleware

Middleware for specific paths:

```
app.use("/admin", adminMiddleware);
```

---

Each approach serves different purposes in structuring your application's request-response cycle efficiently.