

Lecture 8. Middlewares In Nodejs using Express.js

11 April 2025 20:07

Middleware in Express.js – Summary for Revision

What is Middleware?

Middleware functions are functions that have access to the **request (req)**, **response (res)**, and the **next middleware function (next)** in the application's request-response cycle.

They are used to:

- Modify req and res objects
- End the request-response cycle
- Call the next middleware in the stack

Types of Middleware

1. Application-level Middleware
2. Router-level Middleware
3. Built-in Middleware
4. Error-handling Middleware
5. Third-party Middleware

Syntax

js

Copy code

```
app.use((req, res, next) => {  
  // Code to execute  
  next(); // Pass control to next middleware  
});
```

Example 1: Application-level Middleware

Logs request info for every route:

js

Copy code

```
const express = require('express');  
const app = express();  
app.use((req, res, next) => {  
  console.log(`${req.method} ${req.url}`);  
  next(); // Continue to next middleware or route  
});  
app.get('/', (req, res) => {  
  res.send('Home Page');  
});
```

Example 2: Router-level Middleware

Used with `express.Router()`:

js

Copy code

```
const express = require('express');  
const router = express.Router();  
router.use((req, res, next) => {
```

```
console.log('Router middleware triggered');
next();
});
router.get('/user', (req, res) => {
  res.send('User Page');
});
app.use('/api', router);
```

Example 3: Built-in Middleware

To parse JSON payloads:

```
js
Copy code
app.use(express.json());
app.post('/data', (req, res) => {
  res.send(req.body); // Automatically parsed
});
```

Example 4: Error-handling Middleware

Must have 4 arguments:

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

Example 5: Third-party Middleware (e.g., morgan for logging)

```
js
Copy code
const morgan = require('morgan');
app.use(morgan('dev'));
```

Key Points

- Order matters – middleware runs in the order they're defined.
- next() must be called unless the middleware ends the response.
- Middleware can be reused across routes or apps.
- Helps with DRY code – reusable logic like auth, logging, etc.