

Express.js Middleware Beginner Notes

What is Middleware?

1. Middleware is just a function that runs/executes on every request and response in an Express app.
2. It acts as a bridge between the user's request and the main function (route handler or controller).
3. Middleware can validate or check the user's request:
 - If the request is valid, it goes to the main function.
 - If not valid, the request will be rejected or sent back to the user without going to the main function.
4. You can use multiple middleware functions in a single code block.

What Can Middleware Do?

Middleware functions in Express can:

- Execute any code
- Make changes to the request (`req`) or response (`res`) objects
- End the request-response cycle
- Call the next middleware in the stack using `next()`

Using app.use()

The `app.use()` function in Express.js is used to add middleware to the applications request-processing pipeline.

Example:

```
app.use(myMiddleware);
```

This means the middleware will run before any route handler and can modify the request or perform actions like logging.

Middleware Function Parameters

Middleware functions in Express always work with three parameters:

1. req the request object
2. res the response object
3. next a function that calls the next middleware in line

Example:

```
function myMiddleware(req, res, next) {  
  console.log('Hello from middleware!');  
  next();  
}
```

Multiple Middleware Functions

You can add multiple middleware functions in a single route.

Example:

```
function first(req, res, next) {  
  console.log('First middleware');  
  next();  
}  
  
function second(req, res, next) {  
  console.log('Second middleware');  
  next();  
}  
  
app.get('/test', first, second, (req, res) => {  
  res.send('Test Page');  
});
```

What happens here?

1. First, first() middleware runs and logs "First middleware".

2. Then, second() runs and logs "Second middleware".

3. Finally, the response 'Test Page' is sent to the user.