Express - 5

Middlewares

What is a middleware ?

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

They can

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

Purpose of middleware

- Code Reusability: Middleware allows you to reuse code across different routes.
- Request Handling: Modify incoming requests or outgoing responses.
- Error Handling: Handle errors that may occur during request processing.
- Authentication/Authorization: Verify users and restrict access to routes.
- Logging: Log requests and responses for debugging and analytics.

Different Types of Middleware

- Application-level middleware - Logging
- Router-level middleware - Authentication
- Error-handling middleware
- Built-in middleware
- Third-party middleware

Creating a logger middleware

```
const logging = (req, res, next) => {
    console.log(`${req.method} ${req.url}`);
    next();
}

export default logging;
```

Creating an error handler middleware

```
const errorHandler = (error, req, res, next) => {
    res.status(error.status).json({message: error.message})
    next();
}

export default errorHandler
```

```
if(userIndex === -1){
  const error = new Error(`A user with the id of ${id} is not found`);
  error.status = 404;
  return next(error)
}
```