Express-Error Middleware

Error Middleware in Express

In Express.js, error middleware is used to handle errors that occur during the execution of routes. It's a crucial part of building robust and reliable web applications.

Purpose of Error Middleware

The primary goal of error middleware is to catch and handle unexpected errors that might not be anticipated by developers. This ensures that your application remains stable and doesn't crash when an error occurs.

Example of Error Middleware in Express

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

app.use((err, req, res, next) => {
  console.error(err); // Log the error for debugging purposes
  res.status(500).send({ message: 'Internal Server Error' });
});

// Example route that might throw an error
app.get('/users', (req, res) => {
  // Simulating an error by trying to access a non-existent property
  const user = req.body.user;
  if (!user) {
    throw new Error('User not found');
  }
  res.json(user);
});

app.listen(3000, () => {
  console.log('Server listening on port 3000');
});
```

In this example:

- We define an error middleware function that logs the error and returns a 500 Internal Server Error response to clients.
- The `next` function is not called in this middleware, as we're handling the error ourselves.
- We have an example route `/users` that might throw an error if the `user` property is missing from the request body.

Best Practices for Writing Error Middleware

1. **Log errors**: Log errors for debugging purposes to help you identify and fix issues.
2. **Provide a meaningful response**: Return a response with a suitable status code (e.g., 500 Internal Server Error) and provide additional information, if necessary.
3. **Don't call `next()`**: When handling an error in middleware, don't call the `next()` function, as this will propagate the error to the next middleware or route handler.

By following these best practices and using error middleware effectively, you can build more robust and reliable Express.js applications.

Curated by Brajesh Kumar