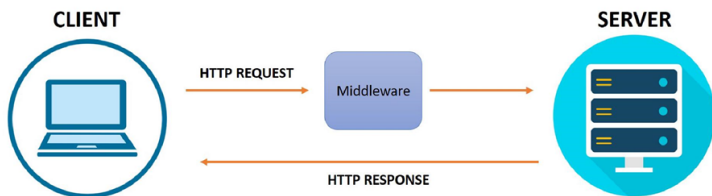


Node.js - Adding Middleware



Middleware functions allow you to handle and modify requests and responses objects, end response cycle and call next middleware.

Very common example of middleware is cors module. To add CORS support, simply install it, require it and put this line:

```
app.use(cors());
```


Simple Middleware Example

A basic logging middleware that logs the request method and URL for every request:

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

// Simple logging middleware
app.use((req, res, next) => {
  console.log(`${req.method} ${req.url}`);
  next(); // Passes control to the next
middleware function
});

app.get('/', (req, res) => {
  res.send('Hello, World!');
});

app.listen(3000, () => {
  console.log('Server is running on
```

```
http://localhost:3000');  
});
```

Here, `app.use()` applies middleware globally to all routes. The `next()` function passes control to the next middleware or route handler.

Advanced Middleware

Example

An authentication and authorization middleware, where a token is checked, and permissions are verified.

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

// Token verification middleware
function authenticateToken(req, res, next) {
  const token = req.header('Authorization');

  if (!token) {
    return res.status(401).json({ message:
'Access denied. No token provided.' });
  }

  // Mock token validation
  if (token === 'VALID_TOKEN') {
```

```

    req.user = { id: 1, role: 'admin' }; // Mock
    user information
    next(); // Passes control to the next
    middleware or route
  } else {
    res.status(403).json({ message: 'Invalid
    token.' });
  }
}

// Authorization middleware for role-based access
function authorizeRole(role) {
  return (req, res, next) => {
    if (req.user && req.user.role === role) {
      next();
    } else {
      res.status(403).json({ message: 'Forbidden:
      Access denied. Insufficient permissions.' });
    }
  };
}

// Routes with middleware
app.get('/admin', authenticateToken,
```

```
authorizeRole('admin'), (req, res) => {  
  res.send('Welcome, admin!');  
});  
  
app.get('/user', authenticateToken, (req, res) =>  
{  
  res.send('Welcome, authenticated user!');  
});  
  
app.listen(3000, () => {  
  console.log('Server is running on  
http://localhost:3000');  
});
```

user route testing with Curl

To test this, use two different curl commands:

Without API Key (should be forbidden)

```
curl http://localhost:1234/user
```

This should return `{"message": "Access denied. No token provided."}` with a 403 status.

With API Key (should be allowed)

```
curl -H "Authorization: VALID_TOKEN"  
http://localhost:1234/user
```

This should return `Welcome, authenticated user!`.

admin route testing with Curl

To test this, use two different curl commands:

Without API Key (should be forbidden)

```
curl http://localhost:1234/admin
```

This should return `{"message": "Access denied. No token provided."}` with a 403 status.

With API Key (should be allowed)

```
curl -H "Authorization: VALID_TOKEN"  
http://localhost:1234/admin
```

This should return `Welcome, admin!`.

AUNG THU OO

I'm Seeking Job **OPPORTUNITIES**

Senior Developer Role

Bringing over 15 years of software industry experience, I use my full-stack skills to deliver scalable, user-friendly web and mobile applications that align with both user needs and business goals.



My Skills

- Next.js
- React.js
- Node.js
- Flutter
- Laravel

VISIT MY WEBSITE & EMAIL

<https://aungthuoo.github.io>
koaungthuoo@gmail.com