Clarusway

Backend
Workshop
-6-

Workshop

Subject:

- JWT & Permissions and Logging & Documentation

Learning Goals

- Working on JWT & Permissions and Logging & Documentation

Introduction

- In the realm of web and software development, securing user access and facilitating effective communication are pivotal. This document explores JWT & Permissions, which govern user authentication and authorization, as well as Logging & Documentation, the cornerstones of monitoring and understanding applications. By the end, you'll be equipped to enhance security and application insights while fostering clear and efficient communication

Prerequisites

- We will use the VSCode.

Lets start

1. Can you explain the concept of middleware in Node.js and how it is employed for managing user permissions using JWT?

Answer: Middleware is used for managing user permissions using JWT (JSON Web Tokens) in the following way:

JWT Authentication Middleware: You create a middleware to check the validity of JWT tokens sent with incoming requests. This middleware verifies the token's signature and expiration, ensuring that the request is from an authenticated user. If the token is valid, it extracts user information from the token payload and attaches it to the request object for further use.

Permission-Based Middleware: Once the user is authenticated, additional middleware functions are created to check user permissions for specific routes or actions. These middleware functions examine the user information attached to the request object and determine if the user has the necessary permissions for the requested action.

Applying Middleware to Routes: Middleware functions are applied to specific routes or route handlers that require authentication and authorization. The JWT authentication middleware ensures that a user is authenticated, and the permission-based middleware verifies if they have the required permissions for the specific route.

2. How can Node.js be used to implement JWT-based authentication and authorization in a web application?

Answer: Node.js can be used to implement JWT-based authentication and authorization in a web application through these key steps:

Install Required Packages: Start by installing packages like jsonwebtoken for creating and verifying JWT tokens, and choose a web framework like Express.js for building your web application.

User Registration and Login: Create user registration and login functionality to validate user credentials.

JWT Token Creation: Upon successful login, generate a JWT token. This token typically contains user information (e.g., user ID, roles, permissions) and is signed with a secret key to ensure its authenticity.

Token Sending to Client: Send the JWT token as a response to the client, often as part of the login response.

Client Token Storage: Store the JWT token securely on the client side, usually in a cookie or local storage.

Route Protection with Middleware: Develop middleware functions to protect routes that require authentication and authorization. Verify the JWT token from the client's request using the secret key.

User Information Extraction: After token verification, extract user information from the token's payload and attach it to the request object for use in route handlers.

Authorization Logic: Implement authorization logic within route handlers to check user roles or permissions. This determines whether the user can access specific resources or perform actions.

Handling Unauthorized Access: If a user lacks the necessary permissions, return an unauthorized (HTTP 401 or 403) response.

Token Renewal (Optional): Implement token renewal for issuing a new JWT when the old one expires, usually through refresh tokens or re-authentication.

3. What is the purpose of structured logging, and how can it benefit the debugging and monitoring of a Node.js application?

Answer:Structured logging aims to organize log data in a consistent format, such as JSON or key-value pairs, to facilitate analysis and understanding. It benefits debugging and monitoring in a Node.js application by providing clear, searchable, and context-rich logs, making it easier to identify and resolve issues quickly

4. In the context of Node.js, what logging and documentation practices should be followed to enhance the overall maintainability and collaboration on a project?

Answer: Logging:

Use Structured Logging: Employ structured logs with clear key-value pairs or JSON format to improve searchability and analysis. Log Relevant Information: Include timestamps, log levels, and relevant contextual information in logs. Centralized Logging: Consider centralizing logs in a dedicated system for easier monitoring and analysis. Error Handling: Log errors and exceptions with relevant details to aid in debugging. Use Log Levels: Employ different log levels (e.g., info, error, debug) for various types of messages. Log Critical Paths: Log key execution paths and user actions to trace issues effectively. Documentation:

Clear API Documentation: Provide comprehensive and up-to-date documentation for APIs, specifying endpoints, request parameters, and response structures. Code Comments: Include meaningful comments in the code to explain complex or critical sections. Readme Files: Maintain informative readme files that describe the project's purpose, setup, and usage. Coding Standards: Enforce coding standards and practices for consistency. Version Control: Use version control tools (e.g., Git) and maintain descriptive commit messages. Collaborative Tools: Utilize collaboration tools, such as issue trackers and project boards, to manage tasks and enhance teamwork.

Clarusway

Backend
Teamwork
-6-

Teamwork

Subject:

- Hotel Api Project

Learning Goals

- Consolidating Pizza Api and Flight Api projects

Introduction

- This document focuses on the consolidation of the Pizza API and Flight API projects. By merging these two distinct systems, we aim to optimize resources, enhance efficiency, and deliver a more cohesive and effective solution.

Prerequisites

- The folder structure of the Pizza Api project should be used.
- At the same time, we need to usage of Erd diagram.

Lets start

alt text

We want a Hotel Api project from you. You can act in accordance with the erd diagram we shared in this project. For convenience, you can use the pizza api file structure.

☺ **Thanks for Attending** ✍

Clarusway
