
EXPRESS

1: Introduction to Express.js

1.1. What is Express.js?

- 1.1.1. Introduction to Node.js and Express.js
- 1.1.2. Key Features of Express.js

1.2. Setting Up a Development Environment

- 1.2.1. Installing Node.js and npm
- 1.2.2. Creating Your First Express Application

2: Building Basic Express Applications

2.1. Hello World with Express

- 2.1.1. Creating Routes and Handling Requests
- 2.1.2. Sending Responses to Clients

2.2. Express Middleware

- 2.2.1. Understanding Middleware Functions
- 2.2.2. Creating Custom Middleware

3: Routing in Express

3.1. Route Handling

- 3.1.1. Defining Routes for Different HTTP Methods
- 3.1.2. Route Parameters and Wildcards

3.2. Route Middleware

- 3.2.1. Applying Middleware to Specific Routes
- 3.2.2. Route Groups and Sub-Routers

4: Express and Template Engines

4.1. Introduction to Template Engines

- 4.1.1. Popular Template Engines (e.g., EJS, Pug)



- 4.1.2. Integrating Template Engines with Express

4.2. Rendering Dynamic Views

- 4.2.1. Passing Data to Views
- 4.2.2. Creating Reusable Templates

5: Middleware and Error Handling

5.1. Advanced Middleware Concepts

- 5.1.1. Error Handling Middleware
- 5.1.2. Third-Party Middleware (e.g., body-parser)

5.2. Error Handling in Express

- 5.2.1. Handling Errors with Custom Middleware
- 5.2.2. Error Response Formats

6: Working with Databases

6.1. Using Databases in Express

- 6.1.1. Connecting to Relational Databases (e.g., MySQL, PostgreSQL)
- 6.1.2. Connecting to NoSQL Databases (e.g., MongoDB)

6.2. CRUD Operations

- 6.2.1. Implementing Create, Read, Update, and Delete
- 6.2.2. Database Queries and Data Retrieval

7: Authentication and Authorization

7.1. User Authentication

- 7.1.1. User Registration and Login
- 7.1.2. Session and Token-Based Authentication

7.2. Authorization and Access Control

- 7.2.1. Role-Based Access Control
- 7.2.2. Middleware for Authentication and Authorization

8: RESTful APIs with Express

8.1. API Design Principles

- 8.1.1. RESTful API Guidelines
- 8.1.2. Structuring API Endpoints

8.2. Creating RESTful Routes

- 8.2.1. Implementing CRUD Operations
- 8.2.2. Handling JSON Data and Responses

9: Real-Time Applications with Express

9.1. WebSockets and Socket.io

- 9.1.1. Real-Time Communication Concepts
- 9.1.2. Integrating Socket.io with Express

9.2. Building a Chat Application

- 9.2.1. Implementing Real-Time Chat Functionality
- 9.2.2. Broadcasting Messages

10: Deployment and Scaling

10.1. Deployment Options

- 10.1.1. Hosting Express Apps on Various Platforms
- 10.1.2. Setting Up Production Environment

10.2. Scaling Express Apps

- 10.2.1. Load Balancing and Scaling Techniques
- 10.2.2. Monitoring and Performance Optimization

11: Express.js Best Practices

11.1. Code Quality and Organization

- 11.1.1. Coding Standards and Style
- 11.1.2. Structuring Express Applications

11.2. Security Considerations

- 11.2.1. Securing Express Apps
- 11.2.2. Handling Security Vulnerabilities

12: Final Project and Course Review

12.1. Project Proposal and Planning

- 12.1.1. Designing a Full-Featured Express Application
- 12.1.2. Implementing Advanced Express Features

12.2. Implementation and Presentation

- 12.2.1. Building the Express Project
- 12.2.2. Final Project Presentation