

Clarusway



Backend Workshop -7-

Workshop

Subject:

- NodeJS Interview Question

Learning Goals

- Preparing for NodeJS Interview Questions

Introduction

- Preparing for interview questions that may arise during job application processes

Prerequisites

- Answer by discussing among yourselves.

Lets start

1. Explain Middleware in Express.js.

Answer: Middleware functions in Express.js are functions that have access to the request and response objects and can modify them. They are used to perform tasks such as logging, authentication, parsing request data, and more. Middleware functions are executed in the order they are defined in the application.

2. What is routing in Express.js, and how is it implemented?

Answer: Routing in Express.js refers to the process of determining how an application responds to a client request to a particular endpoint (URL). In Express, you can define routes using the `app.get()`, `app.post()`, `app.put()`, and `app.delete()` methods to specify the HTTP method and the URL pattern to which the application should respond.

3. How does error handling work in Express.js?

Answer: Express.js provides a built-in error handling mechanism using middleware. You can define error-handling middleware functions that take four parameters (`err`, `req`, `res`, `next`). When an error occurs in your application, you can call `next(err)` to pass the error to the error-handling middleware, where you can log, format, or respond to the error appropriately.

4. What is the purpose of the Express.js `app.use()` method?

Answer: The `app.use()` method is used to mount middleware functions in Express.js. It can be used to specify middleware that should be executed for every request, regardless of the HTTP method or URL pattern. It's often used for tasks like setting up CORS headers, serving static files, or handling sessions.

5. Explain the role of the Express.js response object.

Answer: The response object (`res`) in Express.js is used to send HTTP responses to the client. It provides methods for setting response headers, sending data back to the client, and handling HTTP status codes. For example, you can use `res.send()`, `res.json()`, or `res.status()` to send various types of responses to the client.

6. What is Sequelize, and what is its primary purpose?

Answer: Sequelize is an Object-Relational Mapping (ORM) library for Node.js that simplifies interactions with relational databases. It allows you to perform database operations using JavaScript objects and provides a

structured way to work with databases.

7.What is pagination, and how can you implement it in a database query with Sequelize or Mongoose?

Answer: Pagination is the practice of dividing query results into manageable chunks. In Sequelize, you can use the limit and offset options to implement pagination. In Mongoose, you can use methods like skip() and limit() to achieve pagination.

8.How do you use the 'where' clause in Sequelize to create search queries?

Answer: In Sequelize, you can use the where clause to filter records in a query. For example, to find users with a specific name, you can use User.findAll({ where: { name: 'John' } }).

9.Explain JSON Web Tokens (JWT) and their role in web applications.

Answer: JWT is a compact, self-contained way of representing information between parties as a JSON object. In web applications, JWTs are commonly used for authentication and authorization. They are issued by the server and can be used to verify the identity of the user, access control, and more.

10.What are some common security considerations when using JWT for authentication?

Answer: Some common security considerations when using JWT for authentication include ensuring the token is stored securely, using HTTPS, setting reasonable token expiration times, and verifying the token's signature to prevent tampering. Additionally, it's important to keep sensitive information out of the token payload.





Backend Teamwork -7-

Teamwork

Subject:

- Hotel Api Project

Learning Goals

- Consolidating Pizza Api

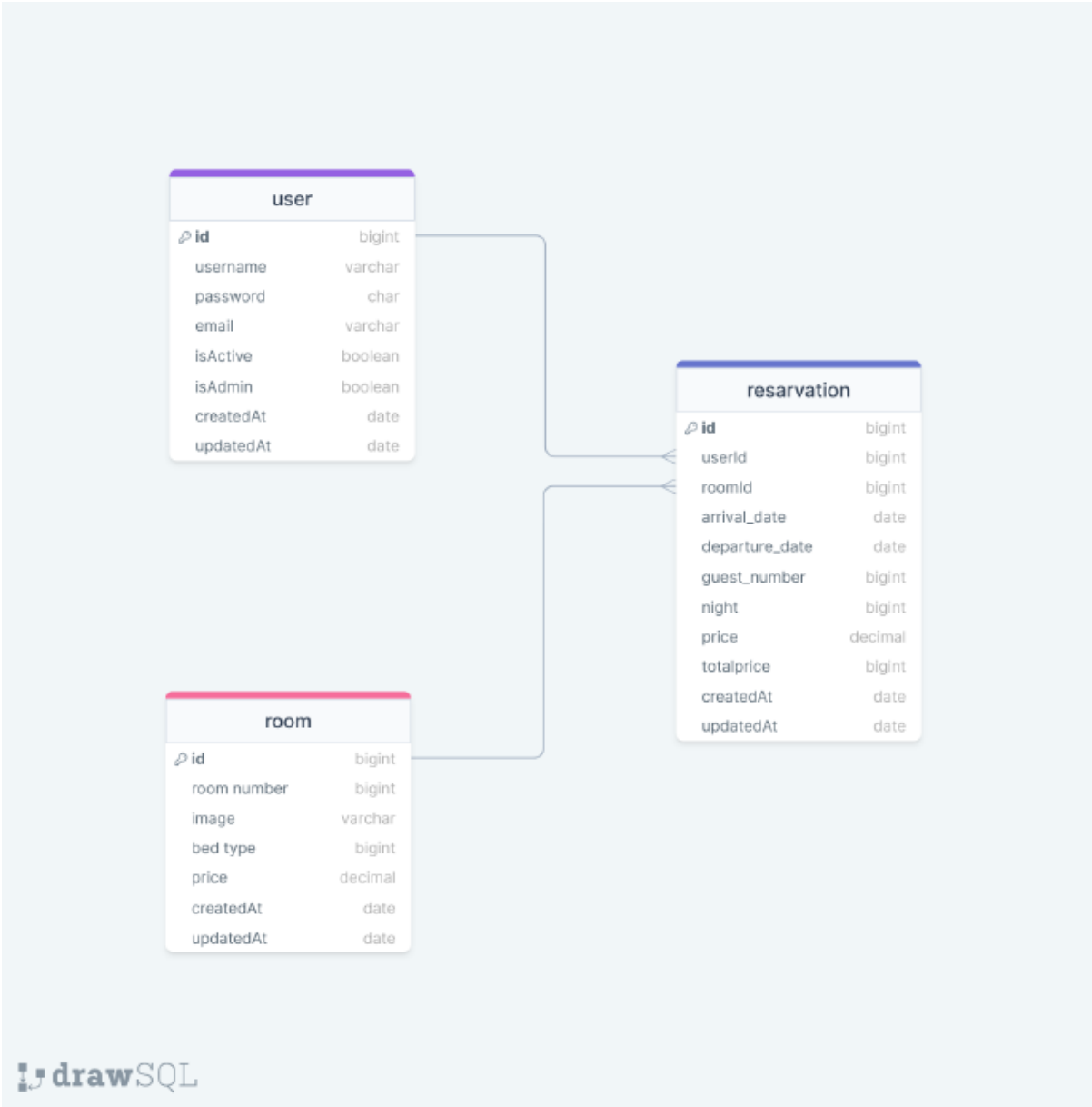
Introduction

-This document focuses on the consolidation of the Pizza API . 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



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 🙌

