Clarusway

Backend Workshop -5-

Workshop

Subject:

- Sequelize & Mongoose

Learning Goals

- Ability to work with RDB and NoSQL databases with Sequelize and Mongoose.

Introduction

- One of the things a backend developer needs to know is to work with SQL and NoSQL databases. For this purpose, studies will be carried out with Sequelize for SQL databases and Mongoose for NoSQL databases.

Prerequisites

- We will use the VSCode.
- At the same time, we need to install Nodejs, MongoDb on our computer.

Lets start

1. How Sequelize ORM works and its role in Node.js ?

Answer:

Sequelize ORM is a library for Node.js that provides Object-Relational Mapping (ORM) to handle tasks associated with relational databases. It abstracts database interactions, allowing developers to use JavaScript instead of SQL queries. Sequelize supports multiple dialects like PostgreSQL, MySQL, SQLite and MSSQL. In Node.js development, Sequelize plays a crucial role in managing data models and relationships. It offers features such as model definition, associations between models, synchronization, migrations, and transactions. Model definitions map directly to tables in the database while associations define relationships between models. Synchronization allows automatic update of database schema when models change. Migrations provide control over changes to the database schema. Transactions ensure atomicity of operations

2. What is Sequelize.sync() and how does it work ?

Answer:

-Sequelize.sync() is a method in Sequelize.js, a Node.js-based Object-Relational Mapping (ORM) library. It synchronizes the stated model with the database by generating new tables if none exist and updating existing ones if they have changed. The sync() function compares the current state of the models to the associated database tables. If discrepancies are discovered, the database is modified to match the models. This includes generating new tables or modifying old ones based on model modifications. To customize the behavior of the sync() method, options can be supplied to it. Passing force: true, for example, will drop tables before recreating them, effectively resetting the database. However, this should be done with caution because it results in data loss.

3. What is ODM ?

Answer:

Object Document Mapping, is a programming technique for converting data between incompatible type systems in MongoDB and Mongoose.ODM is a tool that allows you to interact with your MongoDB database using JavaScript objects.

ODM provides a way to interact with the documents in the database using JavaScript objects. With Mongoose, you can define a schema (a blueprint) for your objects and then use it to create instances of those objects. These instances can be saved, retrieved, updated and deleted in the database using mongoose methods.

It's like an ORM, but instead of SQL databases and tables, it uses MongoDB collections and documents.

4. How to connect to MongoDB using Mongoose ?

Answer:

```
const mongoose = require('mongoose')
mongoose.connect('mongodb://127.0.0.1:27017/dbname')
    .then(() =>
    .catch((err) =>
```

5. What is a Mongoose model, give a simple example ?

Answer:

A Mongoose model is a wrapper of the Mongoose schema. A Mongoose schema defines the document's properties, default values, types of data, validators, etc. In contrast, a Mongoose model provides an interface for the database to CRUD, and so on.

```
const UserSchema = new mongoose.Schema({
  name: String,
  email: String,
});
//Then, create a model from this schema:
const User = mongoose.model('User', UserSchema);
// create user
let newUser = new User({name: "Alex", email: "alex@example.com"});
// newUser.save();
```

Clarusway

Backend
Teamwork
-5-

Teamwork

Subject:

- Express project structure

Learning Goals

- Learning how organize Express.js project structure for better productivity.

Introduction

- The organization helps to maintain consistency, especially in a larger team. Consistency in the project structure equates to the predictability of where the code can be expected, which in turn helps in the productivity of the whole team. In this teamwork, we will learn how to organize an Express project to be used by a team of developer in order to enhance productivity and maintainability.

Prerequisites

We will use the VSCode At the same time, we need to install Nodejs and MongoDb on our computer.

Lets start

First of all, divide the task for this study. and do it together step by step during teamwork time.

- first select a project (e.g. blogapp)
- The team leader or volunteer should eliminate the project folder structure of the selected project and turn it into a single file.
- Those who did not participate in the first phase of the team that came together during the teamwork hour should change the project structure to the recommended project structure step by step.
- While doing this, discuss the purpose and details of each code block with your teammates.

Optional

- The project, which has become a single file, is shared by teamlead in its repo.
- Agree with the team on the desired project structure.
- Assign tasks to each team member.
- Each individual in the team should do their part and request a pull to the team leader.
- Let the team lead review what has been done and perform the merging process.

☺ **Thanks for Attending** ✍

Clarusway