Saturday, 7 October 2023

MongoDb

MongoDB is a popular NoSQL database system

- **Database**: Think of a database like a big digital warehouse where you store your information. This warehouse can hold various types of things.
- **Collections**: In MongoDB, collections are like different sections within the warehouse where you keep related items. For instance, you might have a "Books" collection for storing information about books and a "Movies" collection for storing data about movies.
- **Documents**: Now, within each collection, you have documents, which are like individual items on the shelves. Each document can have different properties or characteristics, just like different books in a library.

Let's say you have a "Books" collection in your MongoDB database:

- Document 1 (Book 1):
    - Title: "To Kill a Mockingbird"
    - Author: "Harper Lee"
    - Genre: "Fiction"
- Document 2 (Book 2):
    - Title: "1984"
    - Author: "George Orwell"
    - Genre: "Dystopian"
- Document 3 (Book 3):
    - Title: "The Catcher in the Rye"
    - Author: "J.D. Salinger"
    - Genre: "Coming-of-age"

Here, each document represents a specific book with its unique information. The "Books" collection is like a bookshelf containing all these different books. You can think of each document as a page in a catalog, where each page represents one item (book) and contains details about that item.

So, in MongoDB:

- Database is like a digital warehouse.
- Collections are like sections in the warehouse for organizing related items.
- Documents are like individual items on the shelves within those sections, each with its unique attributes.

You can add, update, retrieve, or remove these documents from the collections in your MongoDB database, just as you would interact with items in a real warehouse or catalog.

How to Install MongoDb ?

Windows:

1. Download the MongoDB installer from the MongoDB website.
2. Run the installer and follow the installation wizard.

macOS (using Homebrew):

1. Install Homebrew if you don't have it.
2. Open your terminal and run:

bash Copy code

```
brew tap mongodb/brew
brew install mongodb-community
```

Ubuntu Linux:

1. Open your terminal.
2. Run these commands:

sql Copy code

```
sudo apt update
sudo apt install mongodb
```

Remember to check MongoDB's official documentation for any updates or specific instructions for your system.

How to Use MongoDb ?

Once the mongoldb is installed properly in your system. You can open your terminal or command prompt and type “mongosh”

After running this command if you see the results like following then means you can

```
Current Mongosh Log ID: 65219d3faaf97675f78fa058
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverS
electionTimeoutMS=2000&appName=mongosh+1.5.0
Using MongoDB:          5.0.7
Using Mongosh:          1.5.0

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
```

now start using the mongoldb commands on your terminal

Note: This green url that you get after typing mongosh is known as connection url. This the url that will help the external services like mongoldb or any other to connect to your database. Connection url is like a key to connect to your database. “Mongosh” is the command used to start mongoldb in terminal and get the connection url

MongoDb Commands: https://github.com/Ashish1322/mentor-kart-files-batch-4/blob/main/backend%20Tutorials/first-backend/mongodbCommands.js

What is Mongoose ?

Imagine MongoDB as a place to store data, like a digital filing cabinet. It's designed to handle large amounts of data and is well-suited for applications where data doesn't fit neatly into traditional tables (like in SQL databases).

Now, think of Mongoose as an assistant that helps you organize and interact with the data in that filing cabinet. It provides a structured way to define your data models, create, read, update, and delete records in the database, and perform various operations on the data.

In simple terms, you can use Node.js with Mongoose to connect to a MongoDB database, define the structure of your data (called schemas), and then use JavaScript code to perform actions like adding new data, reading data, updating it, or deleting it from the MongoDB database. Mongoose simplifies the process and makes it more organized and developer-friendly compared to interacting with MongoDB directly.

How to work with mongoose

Following is the GitHub repo in which we learnt how we can connect to mongoldb database using mongoose and how to run queries so following is the GitHub repo

https://github.com/Ashish1322/mentor-kart-files-batch-4/tree/main/backend%20Tutorials/mongodbTutorial