MongoDB and Mongoose

Muhammad Zahir Junejo

Lecture - Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
  - Please review Code of Conduct (in Student Undertaking Agreement) if unsure
- No question is daft or silly - **ask them!**
- Q&A session at the end of the lesson, should you wish to ask any follow-up questions.
- Should you have any questions after the lecture, please schedule a mentor session.
- For all non-academic questions, please submit a query: www.hyperiondev.com/support

Lecture Objectives

1. Significance of databases in data storage and management.
2. Introduction to MongoDB, a popular NoSQL database.
3. Interacting with MongoDB using the MongoDB shell and basic commands.
4. Executing queries, updating, and deleting data in MongoDB.

Lecture Objectives (Cont'd)

5. The significance of database interaction in web applications.
6. Introduction to Mongoose as an ODM library for MongoDB.
7. Setting up Mongoose by installing, importing, and connecting to the database.
8. Defining schemas and creating models for structured data storage.

Understanding Databases

- Databases are organized collections of data, structured for easy storage, retrieval, and manipulation.
- They are essential for storing and managing vast amounts of information efficiently.
- Databases are the backbone of applications, powering everything from simple websites to complex systems.

Intro to MongoDB

- MongoDB is a popular NoSQL database that stores data in a flexible, document-oriented format.
- It's designed for scalability, high availability, and ease of development.
- MongoDB is widely used for a variety of applications, including web, mobile, and IoT.

Installing MongoDB

- Server: https://www.mongodb.com/try/download/community
- Shell: https://www.mongodb.com/try/download/shell
  - Extract the zip
  - Rename to MongoDB Shell
  - Move folder to C:/Program Files or parent directory of MongoDB Server folder
  - Add path of bin folder to environment variables
  - Run mongosh command in CMD/Powershell/Terminal

Creating Schemas in MongoDB

- **Schema** is a crucial concept in databases that defines the structure and organization of data within a collection or table.
- Schemas ensure data consistency, integrity, and allow applications to understand how data is stored and retrieved.
- Schema in Relational DB:
  - Defines structure using tables, rows, and columns.
- Schema in MongoDB:
  - Defines data structure using documents (JSON-like objects).
  - No strict schema requirements, allowing flexibility.

```
{
  "name": "John Doe",
  "age": 25,
  "email": "john@example.com"
}
```

Getting Started with MongoDB Console

- Starting MongoDB Shell:
  - Open a terminal and type mongosh to launch the MongoDB shell.
- Basic Commands:
  - show dbs: Show available databases.
  - use database_name: Switch to a specific database. Creates the database if it does not exist.
- Inserting Documents:
  - db.collection_name.insertOne({...}): Insert a single document.
  - db.collection_name.insertMany([...]): Insert multiple documents.

Querying Data

- Basic Queries:
  - db.collection_name.find({...}): Retrieve documents that match the query.
- Query Operators:
  - $eq, $gt, $lt and more for complex queries.
- Projection:
  - Limit the fields returned by specifying fields to include/exclude.
- Example:

```
db.users.find({ age: { $gt: 20 } }, { name: 1, age: 1 })
```

Updating and Deleting Data

- Updating Documents:
  - db.collection_name.updateOne({...}, {...}): Update the first matching document.
  - db.collection_name.updateMany({...}, {...}): Update multiple documents.

- Deleting Documents:
  - db.collection_name.deleteOne({...}): Delete the first matching document.
  - db.collection_name.deleteMany({...}): Delete multiple documents.

Why Database Interaction?

- Database Interaction is crucial for web applications to store, retrieve, and manipulate data.
- Databases provide a structured way to manage information efficiently.
- Seamless interaction between the application and the database ensures data consistency and reliability.

Introduction to Mongoose

- Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js.
- Simplifies the process of working with MongoDB by providing a structured schema and data validation.
- Facilitates interaction with the MongoDB database in a more intuitive manner.

Setting Up Mongoose

- Installation:
  - Install Mongoose using npm or yarn:

```
npm install mongoose
```

```
const mongoose = require('mongoose');
mongoose.connect('mongodb://localhost/mydatabase', {
  useNewUrlParser: true,
  useUnifiedTopology: true
});
```

Defining Schemas

- Schema Definition:
  - Define a schema for your data using mongoose.Schema.
  - Specify fields, their types, and optional configurations.
- Creating Models:
  - Use the schema to create a model using mongoose.model().
  - Models represent collections in the database.

```
const userSchema = new mongoose.Schema({
  name: String,
  age: Number,
  email: String
});
const User = mongoose.model('User', userSchema);
```

CRUD Operations

- Creating Documents:
  - Create new documents using the model's constructor and save() method.
- Reading Documents:
  - Use methods like find(), findOne(), or queries to retrieve data.
- Updating Documents:
  - Use methods like updateOne(), updateMany(), or findOneAndUpdate().
- Deleting Documents:
  - Use methods like deleteOne(), deleteMany(), or findOneAndDelete().

Error Handling

```
try {
  const result = await newUser.save();
} catch (error) {
  console.error('An error occurred:', error.message);
}
```

Questions and Answers

Thank You!