COLÉGIO COTEMIG

DESENVOLVIMENTO WEB 2 (BACKEND)

PROF. FERNANDO – SET 2023

COLÉGIO COTEMIG

MONGODB + TYPESCRIPT

Primeiros Passos

Primeiros Passos

- **Esse material tem como objetivo auxiliar na criação de uma função em TypeScript que retorne uma conexão em um banco de dados MongoDB, sendo que, a partir dessa conexão permita a execução de comandos de persistência de dados.**

  - **Instalar a biblioteca do MongoDB para Node.js:**

    ```
    npm install mongodb@4.0.0
    ```

  - **Criar o diretório “db” na pasta “src”.**
  - **Criar o módulo “index.ts” na pasta “db”.**

Primeiros Passos

- **src/db/index.ts:**

```
// importa a classe MongoClient
import { MongoClient } from "mongodb";

// Cria a string de conexão
const uri = "mongodb://admin:admin@127.0.0.1:27017/devweb2";

// Retorna uma conexão ativa
const getMongoConn = async (): Promise<MongoClient> => {
    const client = new MongoClient(uri); // Cria objeto da conexão
    const conn = await client.connect(); // Conecta na instância
    return conn; // Retorna a conexão
}

// Exporta a função de conexão
export default getMongoConn;
```

Primeiros Passos

- **Criar em “src”, o módulo “index.ts”:**
- **src/index.ts:**

```
// importa a classe MongoClient
import { MongoClient } from "mongodb";

// Importa a função de conexão
import getMongoConn from "./db";

const main = async () => {
    // Variável de conexão
    let conn: MongoClient | null = null;
    try {
        conn = await getMongoConn(); // Cria a conexão
        const db = conn.db(); // Retorna o banco de dados
        const pessoas = db.collection("pessoas"); // Retorna a coleção
```

Primeiros Passos

- **src/index.ts (continuação):**

```
// Insere um documento
let docRet = await pessoas.insertOne({
    nome: "Teste",
    idade: 18
});

// Atualiza um documento
await pessoas.updateOne({
    _id: docRet.insertedId // Filtro
}, {
    $set: {
        nome: "Teste Alterado" // Atualização
    }
});
```

Primeiros Passos

- **src/index.ts (continuação):**

```
// Busca todos os documentos
const data = pessoas.find();
console.log(await data.toArray());
console.log("Número de Documentos: ", await data.count());

// Exclui um documento
await pessoas.deleteOne({
    _id: docRet.insertedId
});
```

Primeiros Passos

- **src/index.ts (continuação):**

```
    } catch (err) {
        console.log(err);
    } finally {
        if (conn) {
            await conn.close(); // Fecha a conexão
        }
    }
}

main(); // Chama a função principal do módulo
```

COLÉGIO COTEMIG

MONGODB + TYPESCRIPT

ObjectId

ObjectId

- **É a função responsável por retornar um objeto que contém um identificador único. Esse objeto retornado é utilizado principalmente para preencher o atributo _id dos documentos, visando dar um única identificação para cada.**
- **Ele encapsula um valor hexadecimal que é composto por:**
  - **Um carimbo de data e hora em segundos;**
  - **Um valor aleatório; e;**
  - **Um contador incremental que é iniciado com um valor aleatório.**
- **Exemplos de utilização:**

```
let novoId = new ObjectId(); // Cria um novo id

let idQualquer = new ObjectId("507f191e810c19729de860ea"); // Um id
hexadecimal já existente
```

ObjectId

- **Exemplos de utilização:**

```
let novoId = new ObjectId(); // Cria um novo id

let carimboDataHora = novoId.getTimestamp(); // Retorna um objeto Date com a data e hora do id

console.log(carimboDataHora); // Exemplo: 2023-09-11T14:22:06.000Z
```

- **o método getTimestamp() retorna um objeto do tipo Date nativo do JS / TS. Com ele é possível recuperar em qual momento o id foi gerado, conforme pode ser visto no exemplo acima.**

COLÉGIO COTEMIG

MONGODB + TYPESCRIPT

Operadores de Query

Operadores de Query

- **Existem diversos operadores de query que podem ser utilizados para filtragem de documentos em uma coleção.**
- **Segue link da documentação oficial:**

  https://www.mongodb.com/docs/v4.4/reference/operator/query/

- **Exemplo:**

```
// WHERE nome = 'Teste' AND idade = 18
let docs = await pessoas.find({
     $and: [
          { nome: "Teste" },
          { idade: 18 }
     ]
}).toArray();
console.log(docs);
```

COLÉGIO COTEMIG

MONGODB + TYPESCRIPT

Prática 16 - Desafio

Prática 16 - Desafio

- Criar um programa que declare um array contendo 5 objetos representando países, sendo que cada país deve ter os seguintes atributos: nome e população.
- Persistir os objetos do array no banco de dados devweb2 no mongoDB na coleção países.
- No mesmo projeto, criar um servidor web Express com a rota /paises utilizando o verbo GET. Nessa rota deve ser passado o parâmetro populacao via query.
- Caso o parâmetro não seja informado ou não seja um número, a rota deve retornar um status 400 com uma mensagem orientativa.
- Caso o parâmetro seja informado, a rota deverá conectar ao mongoDB e recuperar todos os países que tem a população maior ou igual ao parâmetro informado. A resposta da rota deverá ser um array json de países.

Referências

- MongoDB Node.js Database Interaction. **W3Schools**. Disponível em: <https://www.w3schools.com/mongodb/mongodb_nodejs_connect_database.php>. Acesso em: 06 de set. de 2023.
- Query and Projection Operators. **MongoDB**. Disponível em: <https://www.mongodb.com/docs/v4.4/reference/operator/query/>. Acesso em: 11 de set. de 2023.

Let´s Go