**10-06-2025**

package connection;

import org.bson.Document;

import com.mongodb.client.FindIterable;

import com.mongodb.client.MongoClient;

import com.mongodb.client.MongoClients;

import com.mongodb.client.MongoCollection;

import com.mongodb.client.MongoDatabase;

import com.mongodb.client.model.Filters;

public class MongoDB {

public static void main(String[] args) {

// Connect to MongoDB server

MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017");

// Get the "vit" database

MongoDatabase database = mongoClient.getDatabase("vit");

// Access the "products" collection

MongoCollection<Document> collection = database.getCollection("products");

// Delete the document with \_id = 1

collection.deleteOne(Filters.eq("\_id", 1));

System.out.println("Document deleted.");

// Try to retrieve it to verify deletion

FindIterable<Document> documents = collection.find();

for (Document document : documents) {

System.out.println(document.toJson());

}

// Close the connection

mongoClient.close();

}

}

**package** connection;

**import** org.bson.Document;

**import** com.mongodb.client.MongoClient;

**import** com.mongodb.client.MongoClients;

**import** com.mongodb.client.MongoCollection;

**import** com.mongodb.client.MongoDatabase;

**import** com.mongodb.client.model.Filters;

**public** **class** MongoDB {

**public** **static** **void** main(String[] args) {

// Connect to MongoDB server

MongoClient mongoClient = MongoClients.*create*("mongodb://localhost:27017");

// Get the "operators" database

MongoDatabase database = mongoClient.getDatabase("operators");

// Access the "sales" collection

MongoCollection<Document> collection = database.getCollection("sales");

// Delete all documents where item is "Cappuccino"

collection.deleteMany(Filters.*eq*("item", "Cappuccino"));

System.***out***.println("All 'Cappuccino' items deleted.")

// Close the connection

mongoClient.close();

}

}

**package** connection;

**import** org.bson.Document;

**import** com.mongodb.client.MongoClient;

**import** com.mongodb.client.MongoClients;

**import** com.mongodb.client.MongoCollection;

**import** com.mongodb.client.MongoDatabase;

**import** com.mongodb.client.model.Filters;

**import** com.mongodb.client.FindIterable;

**public** **class** MongoDB {

**public** **static** **void** main(String[] args) {

// Connect to MongoDB server

MongoClient mongoClient = MongoClients.*create*("mongodb://localhost:27017");

// Get the "operators" database

MongoDatabase database = mongoClient.getDatabase("operators");

// Access the "sales" collection

MongoCollection<Document> collection = database.getCollection("sales");

// Delete all documents where item is "Cappuccino"

collection.deleteMany(Filters.*eq*("item", "Cappuccino"));

System.***out***.println("All 'Cappuccino' items deleted.");

// Retrieve and print a subset of documents (skip 1, limit 2)

FindIterable<Document> allDocuments = collection.find().skip(1).limit(2);

**for** (Document doc : allDocuments) {

System.***out***.println(doc.toJson());

}

// Close the connection

mongoClient.close();

}

}

package connection;

import org.bson.Document;

import com.mongodb.client.MongoClient;

import com.mongodb.client.MongoClients;

import com.mongodb.client.MongoCollection;

import com.mongodb.client.MongoDatabase;

import com.mongodb.client.model.Filters;

import com.mongodb.client.FindIterable;

public class MongoDB {

public static void main(String[] args) {

// Connect to MongoDB server

MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017");

// Get the "operators" database

MongoDatabase database = mongoClient.getDatabase("operators");

// Access the "sales" collection

MongoCollection<Document> collection = database.getCollection("sales");

// Delete all documents where item is "Cappuccino"

collection.deleteMany(Filters.eq("item", "Cappuccino"));

System.out.println("All 'Cappuccino' items deleted.");

// Retrieve and print a subset of documents (skip 1, limit 2)

FindIterable<Document> selectedDocuments = collection.find().skip(1).limit(2);

int index = 0;

for (Document doc : selectedDocuments) {

if (index % 2 == 0) {

System.out.println("Remaining Document: " + doc.toJson());

}

index++;

}

// Close the connection

mongoClient.close();

}

}

package connection

import org.bson.Document;

import com.mongodb.client.MongoClient;

import com.mongodb.client.MongoClients;

import com.mongodb.client.MongoCollection;

import com.mongodb.client.MongoDatabase;

import com.mongodb.client.FindIterable;

import org.bson.conversions.Bson;

import static com.mongodb.client.model.Sorts.descending;

public class Sorting {

public static void main(String[] args) {

// Connect to MongoDB server

MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017");

// Get the database

MongoDatabase database = mongoClient.getDatabase("yourDatabaseName"); // replace with your DB name

// Get the collection

MongoCollection<Document> collection = database.getCollection("sampleCollection");

// Sort documents by First\_Name in descending order

Bson sort = descending("First\_Name");

FindIterable<Document> documents = collection.find().sort(sort);

// Print sorted documents

for (Document document : documents) {

System.out.println(document.toJson());

}

// Close the connection

mongoClient.close();

}

}

package connection;

import org.bson.Document;

import com.mongodb.client.MongoClient;

import com.mongodb.client.MongoClients;

import com.mongodb.client.MongoCollection;

import com.mongodb.client.MongoDatabase;

import com.mongodb.client.FindIterable;

import org.bson.conversions.Bson;

import static com.mongodb.client.model.Sorts.descending;

public class Sorting {

public static void main(String[] args) {

// Connect to MongoDB server

MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017");

// Get the database (replace with actual DB name)

MongoDatabase database = mongoClient.getDatabase("yourDatabaseName");

// Get the collection

MongoCollection<Document> collection = database.getCollection("sampleCollection");

// Sort documents by First\_Name in descending order

Bson sort = descending("First\_Name");

FindIterable<Document> documents = collection.find().sort(sort);

// Print sorted documents

for (Document document : documents) {

System.out.println(document.toJson());

}

// Close the connection

mongoClient.close();

}

}