# Delete Documents item Latte.

import org.bson.Document;

import com.mongodb.client.MongoClient;

import com.mongodb.client.MongoClients;

import com.mongodb.client.MongoCollection;

import com.mongodb.client.MongoDatabase;

public class Tester {

public static void main(String[] args) {

// Creating a Mongo client

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

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

// Get the collection

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

// Delete one document where item = "Lattes"

collection.deleteOne(Filters.eq("item", "Lattes"));

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

System.out.println("\*\*\*Documents\*\*\*");

// Select all documents

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

for (Document document : documents) {

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

}

}

}

# Delete Document item Cappuccino.

import org.bson.Document;

import com.mongodb.client.MongoClient;

import com.mongodb.client.MongoClients;

import com.mongodb.client.MongoCollection;

import com.mongodb.client.MongoDatabase;

public class Tester {

public static void main(String[] args) {

// Creating a Mongo client

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

// Get the collection

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

// Find all documents

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

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

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

System.out.println("\*\*\*Documents\*\*\*");

// Select all documents

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

for (Document document : documents) {

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

}

}

}

# Using skip after deleting item Cappuccino

import org.bson.Document;

import com.mongodb.client.MongoClient;

import com.mongodb.client.MongoClients;

import com.mongodb.client.MongoCollection;

import com.mongodb.client.MongoDatabase;

public class Tester {

public static void main(String[] args) {

// Creating a Mongo client

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

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

// Get the collection

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

// Delete one document where item = "Cappuccino"

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

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

System.out.println("\*\*\*Documents after skipping 2\*\*\*");

// Find all documents and skip the first 3

FindIterable<Document> documents = collection.find().skip(2);

for (Document document : documents) {

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

}

}

}

# Using skip and limit after deleting item Latte

import org.bson.Document;

import com.mongodb.client.MongoClient;

import com.mongodb.client.MongoClients;

import com.mongodb.client.MongoCollection;

import com.mongodb.client.MongoDatabase;

public class Tester {

public static void main(String[] args) {

// Creating a Mongo client

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

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

// Get the collection

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

// Delete one document where item = "Lattes"

collection.deleteOne(Filters.eq("item", "Lattes"));

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

System.out.println("\*\*\*Documents after skipping 2 and limiting to 1\*\*\*");

// Find all documents and skip the first 3

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

for (Document document : documents) {

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

}

}

}

# Only printing even (2nd,4th,6th..)documents.

import org.bson.Document;

import com.mongodb.client.MongoClient;

import com.mongodb.client.MongoClients;

import com.mongodb.client.MongoCollection;

import com.mongodb.client.MongoDatabase;

public class Tester {

public static void main(String[] args) {

// Creating a Mongo client

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

// Connecting to the database

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

// Getting the sales collection

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

// Fetch all documents

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

// Print only even-numbered documents

int index = 0;

for (Document document : allDocuments) {

if ((index + 1) % 2 == 0) {

System.out.println(document);

}

index++;

}

}

}

# Sorting in Ascending and Descending order

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.BasicDBObject;

public class Tester {

public static void main(String[] args) {

// Create Mongo client and connect to database

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

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

// Get the collection

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

// Sort in Descending order by "item"

System.out.println("\*\*\*Descending Order by item\*\*\*");

FindIterable<Document> allDocuments = collection.find()

.sort(new BasicDBObject("item", -1));

for (Document document : allDocuments) {

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

}

// Sort in Ascending order by "item"

System.out.println("\*\*\*Ascending Order by item\*\*\*");

allDocuments = collection.find()

.sort(new BasicDBObject("item", 1));

for (Document document : allDocuments) {

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

}

}

}

import com.mongodb.client.\*;

import org.bson.Document;

public class Tester {

public static void main(String[] args) {

// Create Mongo client and connect to database

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

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

// Get the collection

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

// Sort in Descending order by "item" using Sorts.descending()

System.out.println("\*\*\*Descending Order by item\*\*\*");

FindIterable<Document> allDocuments = collection.find()

.sort(Sorts.descending("item"));

for (Document document : allDocuments) {

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

}

// Sort in Ascending order by "item" using Sorts.ascending()

System.out.println("\*\*\*Ascending Order by item\*\*\*");

allDocuments = collection.find()

.sort(Sorts.ascending("item"));

for (Document document : allDocuments) {

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

}

}

}