KB금융그룹 | 국민의 평생 금융파트너

2024년 상반기 K-디지털 트레이닝

MongoDB Java 연동

[KB] IT's Your Life

KB 국민은행

MongoDB Java 연동

프로젝트 생성

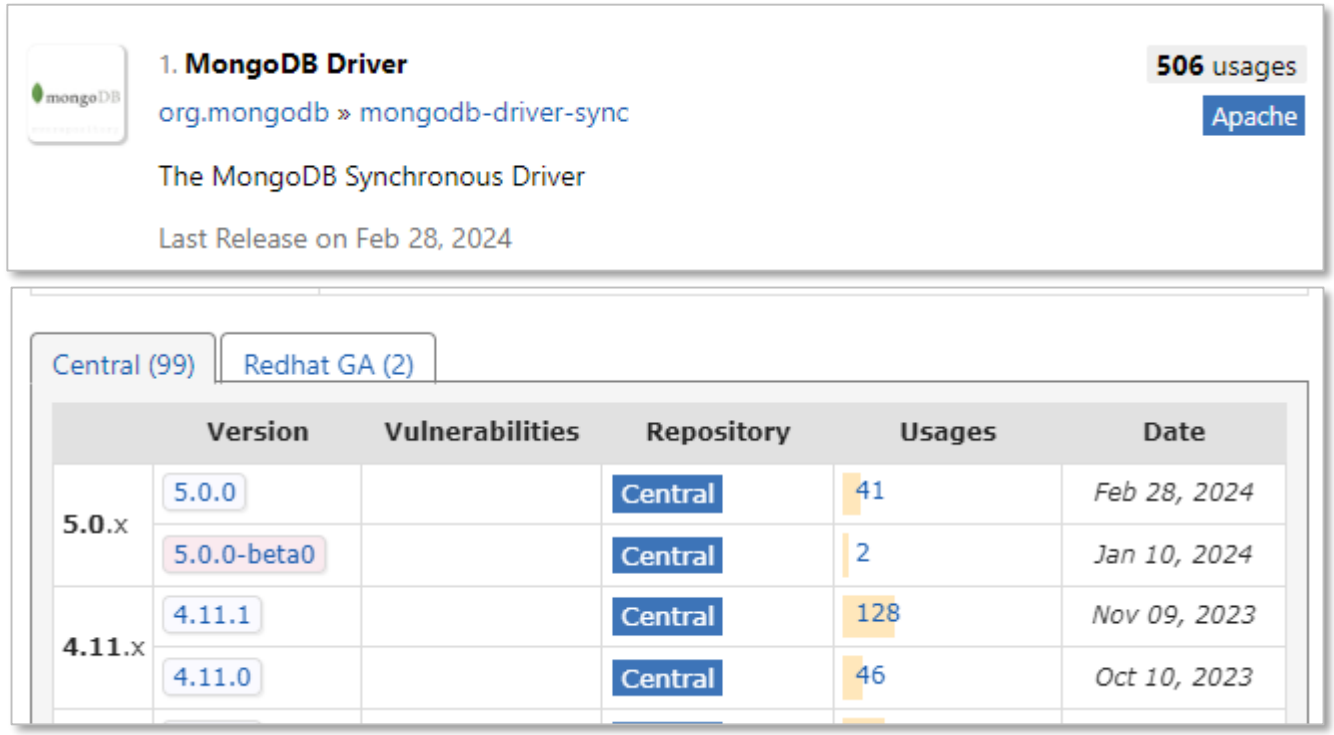
- Name: java_ex
- Location: C:₩KB_Fullstack₩06_MongoDB
- Build System: Gradle
- Gradle DSL: Groovy

- Advanced Settings
  - GroupId: org.scoula
  - ArtifactId : java_ex

MongoDB Java 연동

MongoDB Java 드라이버

- mvnrepository.com
  - mongodb java 검색



1. **MongoDB Driver**

org.mongodb » mongodb-driver-sync

The MongoDB Synchronous Driver

Last Release on Feb 28, 2024

506 usages

Apache

Central (99) | Redhat GA (2)

	Version	Vulnerabilities	Repository	Usages	Date
5.0.x	5.0.0		Central	41	*Feb 28, 2024*
	5.0.0-beta0		Central	2	*Jan 10, 2024*
4.11.x	4.11.1		Central	128	*Nov 09, 2023*
	4.11.0		Central	46	*Oct 10, 2023*

```
implementation 'org.mongodb:mongodb-driver-sync:5.0.0'
```

MongoDB Java 연동

build.gradle

```
dependencies {
    implementation 'ch.qos.logback:logback-classic:1.2.11'
    implementation 'org.mongodb:mongodb-driver-sync:5.0.0'
    …
}
```

resources/logback.xml

```
<configuration>
    <appender name="CONSOLE"
              class="ch.qos.logback.core.ConsoleAppender">
        <encoder>
            <pattern>
                %-4relative [%thread] %-5level %logger{30} - %msg%n
            </pattern>
        </encoder>
    </appender>
    <logger name="org.mongodb.driver.connection" level="INFO" additivity="true"/>
    <root level="INFO">
        <appender-ref ref="CONSOLE" />
    </root>
</configuration>
```

MongoDB 연결하기

```
String uri = "mongodb://127.0.0.1:27017";
String db = "todo_db";

try (MongoClient client =MongoClients.create(uri)){
    MongoDatabase database = client.getDatabase(db);
} catch(Exception e) {
    e.printStackTrace();
}
```

ConnectionTest.java

```
package sec01;

import com.mongodb.client.MongoClient;
import com.mongodb.client.MongoClients;
import com.mongodb.client.MongoDatabase;

public class ConnectionTest {

    public static void main(String[] args) {
        String uri = "mongodb://127.0.0.1:27017";
        String db = "todo_db";

        try (MongoClient client =MongoClients.create(uri)){
            MongoDatabase database = client.getDatabase(db);
        } catch(Exception e) {
            e.printStackTrace();
        }
    }
}
```

MongoDB Java 연동

Database.java

```
package app;

import com.mongodb.ConnectionString;
import com.mongodb.client.MongoClient;
import com.mongodb.client.MongoClients;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;

public class Database {
    static MongoClient client;
    static MongoDatabase database;

    static {
        ConnectionString connectionString = new ConnectionString("mongodb://127.0.0.1:27017");
        client = MongoClients.create(connectionString);
        database = client.getDatabase("todo_db");
    }
```

Database.java

```
    public static void close() {
        client.close();
    }

    public static MongoDatabase getDatabase() {
        return database;
    }

    public static MongoCollection<Document> getCollection(String colName) {
        MongoCollection<Document> collection = database.getCollection(colName);
        return collection;
    }
}
```

MongoDB Java 연동

추가

collection.InsertOne(Document)

```
Document document = new Document();

document.append("title", "MongoDB");
document.append("desc", "MongoDB 공부하기");
document.append("done", false);

InsertOneResult result = collection.insertOne(document);
System.out.println("==> InsertOneResult : " + result.getInsertedId());
```

InsertOneTest.java

```
package sec02;

import app.Database;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.result.InsertOneResult;
import org.bson.Document;

public class InsertOneTest {
    public static void main(String[] args) {
        MongoCollection<Document> collection = Database.getCollection("todo");

        Document document = new Document();
        document.append("title", "MongoDB");
        document.append("desc", "MongoDB 공부하기");
        document.append("done", false);

        InsertOneResult result = collection.insertOne(document);
        System.out.println("==> InsertOneResult : " + result.getInsertedId());
        Database.close();
    }
}
```

MongoDB Java 연동

추가

collection.insertMany(List)

```
List<Document> insertList = new ArrayList<>();
Document document1 = new Document();
Document document2 = new Document();

document1.append("title", "Dune2 영화보기");
document1.append("desc", "이번 주말 IMAX로 Dune2 영화보기");
document1.append("done", false);

document2.append("title", "Java MongoDB 연동");
document2.append("desc", "Java로 MongoDB 연동 프로그래밍 연습하기");
document2.append("done", false);
insertList.add(document1);
insertList.add(document2);

InsertManyResult result = collection.insertMany(insertList);
System.out.println("==> InsertManyResult : " + result.getInsertedIds());
```

InsertManyTest.java

```
package sec02;

import app.Database;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.result.InsertManyResult;
import org.bson.Document;

import java.util.ArrayList;
import java.util.List;

public class InsertManyTest {
    public static void main(String[] args) {
        MongoCollection<Document> collection = Database.getCollection("todo");

        List<Document> insertList = new ArrayList<>();

        Document document1 = new Document();
        Document document2 = new Document();
```

InsertManyTest.java

```
        document1.append("title", "Dune2 영화보기");
        document1.append("desc", "이번 주말 IMAX로 Dune2 영화보기");
        document1.append("done", false);

        document2.append("title", "Java MongoDB 연동");
        document2.append("desc", "Java로 MongoDB 연동 프로그래밍 연습하기");
        document2.append("done", false);

        insertList.add(document1);
        insertList.add(document2);

        InsertManyResult result = collection.insertMany(insertList);
        System.out.println("==> InsertManyResult : " + result.getInsertedIds());

        Database.close();
    }
}
```

InsertManyTest2.java

```
package sec02;

import app.Database;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.result.InsertManyResult;
import org.bson.Document;
…

public class InsertMany2Test {
    public static void main(String[] args) {
        MongoCollection<Document> collection = Database.getCollection("users");

        List<Document> insertList = new ArrayList<>();
        for(int i = 10; i < 21; i++) {
            Document document = new Document();
            document.append("name", "user_" + i);
            document.append("age", i);
            document.append("created", new Date() );
            insertList.add(document);
        }
```

InsertManyTest2.java

```
        InsertManyResult result = collection.insertMany(insertList);
        System.out.println("==> InsertManyResult : " + result.getInsertedIds());

        Database.close();
    }
}
```

비교연산자

이름	설명
eq	지정된 값과 같은 값을 찾습니다.
ne	지정된 값과 같지 않은 모든 값과 일치합니다.
lt	지정된 값보다 작은 값과 일치합니다.
lte	지정된 값보다 작거나 같은 값을 찾습니다.
gt	지정된 값보다 큰 값을 찾습니다.
gte	지정된 값보다 크거나 같은 값과 일치합니다.
in	배열에 지정된 값 중 하나와 일치합니다.
nin	배열에 지정된 값과 일치하지 않습니다.

- static 메서드 임포트로 간소화해서 사용
  - import static com.mongodb.client.model.Filters.eq;

MongoDB Java 연동

조회

단순 조회

```
FindIterable<Document> doc = collection.find();

Iterator itr = doc.iterator();
while (itr.hasNext()) {
    System.out.println("==> findResultRow : " + itr.next());
}
```

id 조회

```
String id = "...";
Bson query = eq("_id", new ObjectId(id));

Document doc = collection.find(query).first();
System.out.println("==> findByIdResult : " + doc);
```

FindTest.java

```
package sec03;

import app.Database;
import com.mongodb.client.FindIterable;
import com.mongodb.client.MongoCollection;
import org.bson.Document;

import java.util.Iterator;

public class FindTest {
    public static void main(String[] args) {
        MongoCollection<Document> collection = Database.getCollection("todo");

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

        Iterator itr = doc.iterator();
        while (itr.hasNext()) {
            System.out.println("==> findResultRow : "+itr.next());
        }

        Database.close();
    }
}
```

FindOneTest.java

```
package sec03;
…
import org.bson.conversions.Bson;
import org.bson.types.ObjectId;

import static com.mongodb.client.model.Filters.eq;

public class FindOneTest {
    public static void main(String[] args) {
        MongoCollection<Document> collection = Database.getCollection("todo");

        String id = "666a6296f4fe57189cd03eea";
        Bson query = eq("_id", new ObjectId(id));

        Document doc = collection.find(query).first();
        System.out.println("==> findByIdResult : " + doc);
        Database.close();
    }
}
```

수정

- updateOne(쿼리, 수정내용)

```
Bson query = eq("_id", new ObjectId(id));

Bson updates = Updates.combine(
        Updates.set("name", "modify name"),
        Updates.currentTimestamp("lastUpdated")
);

UpdateResult result = collection.updateOne(query, updates);
System.out.println("==> UpdateResult : " + result.getModifiedCount());
```

UpdateOneTest.java

```
package sec04;
…

public class UpdateOne {
    public static void main(String[] args) {
        MongoCollection<Document> collection = Database.getCollection("users");

        String id = "666a6b65a25a4c74fddfedc2";
        Bson query = eq("_id", new ObjectId(id));

        Bson updates = Updates.combine(
                Updates.set("name", "modify name"),
                Updates.currentTimestamp("lastUpdated"));

        UpdateResult result = collection.updateOne(query, updates);
        System.out.println("==> UpdateResult : " + result.getModifiedCount());

        Database.close();
    }
}
```

MongoDB Java 연동

수정

- updateMany(쿼리, 수정내용)

```
Bson query = gt("age", age);

Bson updates = Updates.combine(
        Updates.set("name", "modify name"),
        Updates.currentTimestamp("lastUpdated"));

UpdateResult result = collection.updateMany(query, updates);
System.out.println("==> UpdateManyResult : " + result.getModifiedCount());
```

UpdateManyTest.java

```
package sec04;
…
import static com.mongodb.client.model.Filters.gt;

public class UpdateMany {
    public static void main(String[] args) {
        MongoCollection<Document> collection = Database.getCollection("users");

        int age = 16;
        Bson query = gt("age", age);

        Bson updates = Updates.combine(
                Updates.set("name", "modify name"),
                Updates.currentTimestamp("lastUpdated"));

        UpdateResult result = collection.updateMany(query, updates);
        System.out.println("==> UpdateManyResult : " + result.getModifiedCount());

        Database.close();
    }
}
```

MongoDB Java 연동

삭제

deleteOne(쿼리)

```
Bson query = eq("_id", new ObjectId(id));

DeleteResult result = collection.deleteOne(query);
System.out.println("➔ DeleteResult : " + result.getDeletedCount());
```

deleteMany(쿼리)

```
Bson query = gt("age", age);

DeleteResult result = collection.deleteMany(query);
System.out.println("==> DeleteManyResult : " + result.getDeletedCount());
```

DeleteOneTest.java

```
package sec05;

…

public class DeleteOneTest {
    public static void main(String[] args) {
        MongoCollection<Document> collection = Database.getCollection("users");

        String id = "666a6b65a25a4c74fddfedc2";
        Bson query = eq("_id", new ObjectId(id));

        DeleteResult result = collection.deleteOne(query);
        System.out.println("==> DeleteResult : " + result.getDeletedCount());

        Database.close();

    }
}
```

DeleteManyTest.java

```
package sec05;

…

public class DeleteManyTest {
    public static void main(String[] args) {
        MongoCollection<Document> collection = Database.getCollection("users");

        int age = 15;
        Bson query = gt("age", age);

        DeleteResult result = collection.deleteMany(query);
        System.out.println("==> DeleteManyResult : " + result.getDeletedCount());


        Database.close();

    }
}
```