

2025년 상반기 K-디지털 트레이닝

MongoDB Java 연동 (기본)

[KB] IT's Your Life

 수업시 사용했던 프로젝트 build.gradle 파일에 다음과 같이 추가하세요.

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

 배포한 logback.xml 파일을 resources 폴더에 배치하세요.

```
<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>
```

- ✔ **몽고DB 커넥션 uri를 작성하고 todo_db 데이터베이스 추출 코드를 완성하세요.**

```
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 = _____;
        String db = _____;
        try (MongoClient client = _____){
            MongoDatabase database = _____);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

✓ ConnectionTest

```
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();
        }
    }
}
```

- ✔ 앞에서 작성한 MongoDB 연결 코드를 재사용을 위한 Database 클래스로 정의하고자 한다. 다음 코드를 완성하세요.

```
package app;

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

    static {
        // 데이터베이스 연결 및 database 참조 설정
    }

    public static void close() { // 데이터베이스 닫기
    }

    public static MongoDB getDatabase() { // 데이터베이스 참조 얻기
    }

    public static MongoClient getCollection(String colName) { // 전달된 컬렉션이름의 컬렉션 리턴하기
    }
}
```

✓ Database.java

```
package app;
...
public class Database {
    static MongoClient client;
    static MongoDB database;

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

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

    public static MongoDB getDatabase() {
        return database;
    }
}
```

✓ Database.java

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


✓ Document를 이용하여 다음과 같은 정보를 갖는 문서 하나를 추가하는 코드를 완성하세요.

- title: "MongoDB"
- desc: "MongoDB 공부하기"
- done: false

```
package sec02;

public class InsertOneTest {
    public static void main(String[] args) {

    }
}
```

✓ InsertOneTest.java

```
package sec02;
...

public class InsertOneTest {
    public static void main(String[] args) {
        MongoClient<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();
    }
}
```

- ✓ Document를 이용하여 다음과 같은 정보를 갖는 문서 두개를 insertMany()로 추가하는 코드를 완성하세요.

- document1

- title: "MongoDB1"
- desc: "MongoDB 공부하기1"
- done: false

- document2

- title: "MongoDB2"
- desc: "MongoDB 공부하기2"
- done: true

```
package sec02;

public class InsertManyTest {
    public static void main(String[] args) {

    }
}
```

✓ InsertManyTest.java

```
package sec02;
...
public class InsertManyTest {
    public static void main(String[] args) {
        MongoClient
```

✓ InsertManyTest.java

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

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

Database.close();
}
```

✓ 루프문을 사용하여 10개의 user 데이터를 users 컬렉션에 insertMany로 추가하세요.

○ 문서 필드

- name: user_i
- age: i
- created: 실제 실행된 시간

→ i는 10에서 20사이의 값

```
package sec02;

public class InsertMany2Test {
    public static void main(String[] args) {

    }
}
```

✓ InsertManyTest2.java

```
package sec02;
...
public class InsertMany2Test {
    public static void main(String[] args) {
        MongoClient<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);
        }

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

✓ **todo 컬렉션에 저장된 모든 문서를 출력하세요.**

```
package sec03;

public class FindTest {
    public static void main(String[] args) {

    }
}
```


✓ FindTest.java

```
package sec03;
...

public class FindTest {
    public static void main(String[] args) {
        MongoClient<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();
    }
}
```

- ✅ **todo 컬렉션에 저장된 문서 하나의 id 값을 준비하고, 이 값으로 문서 하나를 찾아 출력하세요.**

```
package sec03;

public class FindOneTest {
    public static void main(String[] args) {

    }
}
```

✓ FindOneTest.java

```
package sec03;
...

public class FindOneTest {
    public static void main(String[] args) {
        MongoClient<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();
    }
}
```