

2025년 상반기 K-디지털 트레이닝

MongoDB Java POJO 연동

[KB] IT's Your Life

✓ POJO 매핑

- Document로 실제 데이터를 처리하는 경우 매우 번거로움
- Java POJO 클래스로 작업 직원 필요

✓ build.gradle

```
dependencies {  
    compileOnly("org.projectlombok:lombok:1.18.32")  
    annotationProcessor("org.projectlombok:lombok:1.18.32")  
    testCompileOnly("org.projectlombok:lombok:1.18.32")  
    testAnnotationProcessor("org.projectlombok:lombok:1.18.32")  
    ...  
}
```

Database.java

```
package app;

...
import org.bson.codecs.configuration.CodecProvider;
import org.bson.codecs.configuration.CodecRegistry;
import org.bson.codecs.pojo.PojoCodecProvider;
import static com.mongodb.MongoClientSettings.getDefaultCodecRegistry;
import static org.bson.codecs.configuration.CodecRegistries.fromProviders;
import static org.bson.codecs.configuration.CodecRegistries.fromRegistries;

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

    static {
        CodecProvider pojoCodecProvider = PojoCodecProvider.builder().automatic(true).build();
        CodecRegistry pojoCodecRegistry = fromRegistries(getDefaultCodecRegistry(), fromProviders(pojoCodecProvider));

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

Database.java

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

public static MongoDBDatabase getDatabase() {
    return database;
}

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

public static <T> MongoCollection<T> getCollection(String colName, Class<T> clazz) {
    MongoCollection<T> collection = database.getCollection(colName, clazz);
    return collection;
}
}
```

Todo.java

```
package app.domain;

...
import org.bson.types.ObjectId;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class Todo {
    private ObjectId id;
    private String title;
    private String desc;
    private boolean done;
}
```

Todo.java

```
package app.domain;

...
import org.bson.types.ObjectId;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class Todo {
    private ObjectId id;
    private String title;
    private String desc;
    private boolean done;
}
```

App.java

```
package app;

import app.domain.TODO;
import com.mongodb.client.MongoCollection;
import org.bson.conversions.Bson;
import org.bson.types.ObjectId;

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

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

public class App {
    public static void main(String[] args) {
        MongoCollection<TODO> collection = Database.getCollection("todo", TODO.class);
```


App.java

```
// insertOne
Todo newtodo = new Todo(null, "POJO", "POJO 테스트 확인", false);
collection.insertOne(newtodo);

// insertMany
List <Todo> newTodos = Arrays.asList(
    new Todo(null, "POJO2", "POJO2 테스트 확인", false),
    new Todo(null, "POJO3", "POJO3 테스트 확인", true),
    new Todo(null, "POJO4", "POJO4 테스트 확인", false)
);

collection.insertMany(newTodos);
```

App.java

```
// find()
List<Todo> todos = new ArrayList<>();
collection.find().into(todos);

for(Todo todo : todos) {
    System.out.println(todo);
}

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

Todo todo = collection.find(query).first();
System.out.println("==> findByIdResult : " + todo);

Database.close();
}
}
```

- ✓ update와 delete는 기존 방식과 동일