KB금융그룹 | 국민의 평생 금융파트너

2024년 상반기 K-디지털 트레이닝

MongoDB Java Todo App

[KB] IT's Your Life

KB국민은행

1 MongoDB Java Todo App

- **프로젝트 만들기**
  - Name: todoapp

MongoDB Java Todo App

setting.gradle

```
rootProject.name = 'todoapp'
include ':mylib'
project(':mylib').projectDir=new File('C:\\KB_Fullstack\\04_Java\\myapp')
```

build.gradle

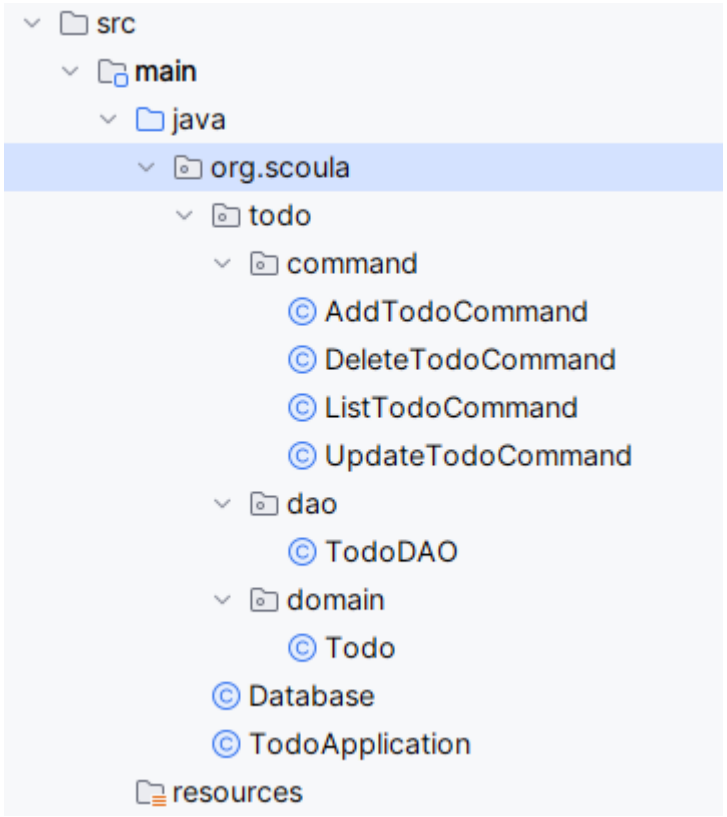
```
dependencies {
    implementation project(":mylib")
    implementation 'ch.qos.logback:logback-classic:1.2.11'
    implementation 'org.mongodb:mongodb-driver-sync:5.0.0'
    compileOnly("org.projectlombok:lombok:1.18.32")
    annotationProcessor("org.projectlombok:lombok:1.18.32")
    testCompileOnly("org.projectlombok:lombok:1.18.32")
    testAnnotationProcessor("org.projectlombok:lombok:1.18.32")

    testImplementation platform('org.junit:junit-bom:5.9.1')
    testImplementation 'org.junit.jupiter:junit-jupiter'
}
```

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 Java Todo App



todo.command.ListTodoCommand.java

```
public class ListTodoCommand implements Command {

    @Override
    public void execute() {
        System.out.println("Todo 목록 보기");
    }
}
```

todo.command.AddTodoCommand.java

```
public class AddTodoCommand implements Command {

    @Override
    public void execute() {
        System.out.println("Todo 추가");
    }
}
```

todo.command.UpdateTodoCommand.java

```
public class UpdateTodoCommand implements Command {

    @Override
    public void execute() {
        System.out.println("Todo 수정");
    }
}
```

todo.command.DeleteTodoCommand.java

```
public class DeleteTodoCommand implements Command {

    @Override
    public void execute() {
        System.out.println("Todo 삭제");
    }
}
```

Application.java

```
public class TodoApplication extends Application {

    @Override
    public void createMenu(Menu menu) {
        super.createMenu(menu);

        menu.add(new MenuItem("목록", new ListTodoCommand()));
        menu.add(new MenuItem("추가", new AddTodoCommand()));
        menu.add(new MenuItem("수정", new UpdateTodoCommand()));
        menu.add(new MenuItem("삭제", new DeleteTodoCommand()));
    }

    public static void main(String[] args) {
        TodoApplication app = new TodoApplication();
        app.run();
    }
}
```

MongoDB Java Todo App

- TodoApplication 실행 확인

Database.java

```
package org.scoula;

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 MongoCollection<Document> getTodoCollection() {
        MongoCollection<Document> collection = database.getCollection("todo");
        return collection;
    }
}
```

Application.java

```
public class TodoApplication extends Application {
  …
    @Override
    public void cleanup() {
        super.cleanup();
        Database.close();
    }

}
```

MongoDB Java Todo App

Todo 도메인 객체 정의

- Document는 자바 프로그램에서 바로 이용하기에는 불편

todo.domain.Todo.java

```
package org.scoula.todo.domain;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;
import org.bson.Document;
import org.bson.types.ObjectId;

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

todo.domain.Todo.java

```
    public Todo(Document doc) {
        id = doc.getObjectId("_id").toString();
        title = doc.getString("title");
        desc = doc.getString("desc");
        done = doc.getBoolean("done");
    }

    public Document toDocument() {
        Document doc = new Document();
        if(id != null) {
            doc.append("_id", new ObjectId(id));
        }
        doc.append("title", title);
        doc.append("desc", desc);
        doc.append("done", done);
        return doc;
    }
}
```

MongoDB Java Todo App

- **TodoDAO**
  - 데이터베이스에 Todo의 CRUD 연산을 수행하는 클래스
  - 싱글톤 객체로 정의

todo.dao.TodoDAO.java

```
package org.scoula.todo.dao;
…

public class TodoDAO {
    private static TodoDAO instance;

    MongoCollection<Document> collection;

    private TodoDAO() {
        collection = Database.getTodoCollection();
    }

    public static TodoDAO getInstance() {
         if(instance == null) {
             instance = new TodoDAO();
         }
         return instance;
    }
```

todo.dao.TodoDAO.java

```
public void add(Todo todo) {
    InsertOneResult result = collection.insertOne(todo.toDocument());
}
```

todo.dao.TodoDAO.java

```
public List<Todo> getList() {
    new ArrayList<>();
    FindIterable<Document> docs = collection.find();
    List<Todo> list = StreamSupport.stream(docs.spliterator(), false)
            .map(Todo::new)
            .toList();

    return list;
}

public Todo get(String id) {
    Bson query = eq("_id", new ObjectId(id));
    Document doc = collection.find(query).first();
    return doc != null ? new Todo(doc) : null;
}
```

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

todo.dao.TodoDAO.java

```
public void update(Todo todo) {
    Bson query = eq("_id", new ObjectId(todo.getId()));

    Bson updates = Updates.combine(
            Updates.set("title", todo.getTitle()),
            Updates.set("desc", todo.getDesc()),
            Updates.set("done", todo.isDone())
    );
    collection.updateOne(query, updates);
}
```

todo.dao.TodoDAO.java

```
    public void delete(String id) {
        Bson query = eq("_id", new ObjectId(id));
        collection.deleteOne(query);
    }

}
```

MongoDB Java Todo App

- **메뉴 기능 구현하기**
  - TodoDAO 객체를 멤버 변수로 가져야 함

todo.command.ListTodoCommand.java

```
public class ListTodoCommand implements Command {
    TodoDAO dao = TodoDAO.getInstance();

    @Override
    public void execute() {
        System.out.println("Todo 목록 보기");
        List<Todo> list = dao.getList();
        for(Todo todo: list) {
            System.out.printf("[%s] %s, %s, %s",
                              todo.getId(), todo.getTitle(), todo.getDesc(), todo.isDone());
        }
    }
}
```

todo.command.AddTodoCommand.java

```
public class AddTodoCommand implements Command {
    TodoDAO dao = TodoDAO.getInstance();

    @Override
    public void execute() {
        System.out.println("Todo 추가");

        String title = Input.read("제목: ");
        String desc = Input.read("설명: ");
        Todo todo = Todo.builder()
                .title(title)
                .desc(desc)
                .build();
        dao.add(todo);
    }
}
```

todo.command.AddTodoCommand.java

```
public class UpdateTodoCommand implements Command {
    TodoDAO dao = TodoDAO.getInstance();

    @Override
    public void execute() {
        System.out.println("Todo 수정");
        String id = Input.read("수정할 Todo ID: ");
        Todo todo = dao.get(id);

        String title = Input.read("제목", todo.getTitle());
        todo.setTitle(title);
        String desc = Input.read("설명", todo.getDesc());
        todo.setDesc(desc);
        boolean done = Input.read("설명", String.valueOf(todo.isDone())).equalsIgnoreCase("true");
        todo.setDone(done);

        dao.update(todo);
    }
}
```

todo.command.DeleteTodoCommand.java

```
public class DeleteTodoCommand implements Command {
    TodoDAO dao = TodoDAO.getInstance();

    @Override
    public void execute() {
        System.out.println("Todo 삭제");
        String id = Input.read("삭제할 Todo ID: ");
        dao.delete(id);

    }
}
```