KB금융그룹 | 국민의 평생 금융파트너

2024년 상반기 K-디지털 트레이닝

JDBC 프로그래밍 - Todo

[KB] IT's Your Life

KB국민은행

1 JDBC 프로그래밍

데이터베이스 준비

- jdbc_ex 계정 작업

```
CREATE TABLE TODO (
  ID          INTEGER AUTO_INCREMENT NOT NULL PRIMARY KEY,
  TITLE       VARCHAR(128) NOT NULL,
  DESCRIPTION VARCHAR(512) NULL,
  DONE        BOOLEAN,
  USER_ID     VARCHAR(12) NOT NULL,
  FOREIGN KEY (USER_ID) REFERENCES USERS(ID)
     ON UPDATE CASCADE
     ON DELETE CASCADE
);

INSERT INTO todo(title, description, done, user_id)
values ("야구장", "프로야구 경기도 봐야합니다.", false, "guest"),
       ("놀기", "노는 것도 중요합니다.", false, "guest"),
       ("Vue 학습", "Vue 학습을 해야 합니다", false, "member"),
       ("ES6 공부", "ES6공부를 해야 합니다", true, "guest");
```

Todo.java

```
package org.example.domain;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class TodoVO {
    private Long id;
    private String title;
    private String description;
    private Boolean done;
    private String userId;
}
```

TodoDao.java

```
package org.example.dao;

import org.example.common.JDBCUtil;
import org.example.domain.TodoVO;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

public class TodoDAO {
    // TODO 테이블 관련 SQL 명령어
    private String TODO_LIST   = "select * from todo";
    private String TODO_GET    = "select * from todo where id = ?";
    private String TODO_INSERT = "insert into todo(title, description, done, user_id) values(?, ?, ?, ?)";
    private String TODO_UPDATE = "update todo set done = ? where id = ?";
    private String TODO_DELETE = "delete todo where id = ?";
```

TodoDao.java

```
// TODO 등록
public void insertTodo(TodoVO todo) {
    Connection conn = JDBCUtil.getConnection();
    try (PreparedStatement stmt = conn.prepareStatement(TODO_INSERT)) {
        stmt.setString(1, todo.getTitle());
        stmt.setString(2, todo.getDescription());
        stmt.setBoolean(3, todo.getDone());
        stmt.setString(4, todo.getUserId());
        stmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

1 JDBC 프로그래밍

TodoDao.java

```java
// TODO 목록 조회
public List<TodoVO> getTodoList() {
    List<TodoVO> todoList = new ArrayList<TodoVO>();
    Connection conn = JDBCUtil.getConnection();
    try (PreparedStatement stmt = conn.prepareStatement(TODO_LIST);
         ResultSet rs = stmt.executeQuery()) {

        while(rs.next()) {
            TodoVO todo = new TodoVO();
            todo.setId(rs.getLong("id"));
            todo.setTitle(rs.getString("title"));
            todo.setDescription(rs.getString("description"));
            todo.setDone(rs.getBoolean("done"));
            todo.setUserId(rs.getString("user_id"));
            todoList.add(todo);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }

    return todoList;
}
```

TodoDao.java

```java
// TODO 정보 조회
public TodoVO getTodo(String id) {
    Connection conn = JDBCUtil.getConnection();
    try (PreparedStatement stmt = conn.prepareStatement(TODO_GET)) {
        stmt.setLong(1, id);
        try(ResultSet rs = stmt.executeQuery()) {
            if(rs.next()) {
                TodoVO todo = new TodoVO();
                todo.setId(rs.getLong("id"));
                todo.setTitle(rs.getString("title"));
                todo.setDescription(rs.getString("description"));
                todo.setDone(rs.getBoolean("done"));
                todo.setUserId(rs.getString("user_id"));
                return todo;
            }
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }

    return null;
}
```

TodoDao.java

```java
    // TODO 수정
    public void updateTodo(TodoVO todo) {
        Connection conn = JDBCUtil.getConnection();
        try ( PreparedStatement stmt = conn.prepareStatement(TODO_UPDATE)) {
            stmt.setBoolean(1, todo.getDone());
            stmt.setLong(2, todo.getId());
            stmt.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
```

1 JDBC 프로그래밍

TodoDao.java

```java
    // TODO 테이블 관련 CRUD 메소드
    // TODO 삭제
    public void deleteTodo(Long id) {
        Connection conn = JDBCUtil.getConnection();
        try(PreparedStatement stmt = conn.prepareStatement(TODO_DELETE)) {
            stmt.setLong(1, id);
            stmt.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

}
```

TodoDaoTest.java

```
package org.example;

import org.example.dao.TodoDAO;
import org.example.domain.TodoVO;

import java.util.List;

public class TodoDAOTest {
    public static void main(String[] args) {
        TodoDAO dao = new TodoDAO();

        // 새로운 Todo 추가
        TodoVO todo = new TodoVO(null, "영화보기", "오펜하이머 영화를 본다.", false, "admin");
        dao.insertTodo(todo);

        // Todo 목록 얻기
        List<TodoVO> list = dao.getTodoList();
        for(TodoVO vo : list) {
            System.out.println(vo);
        }
```

TodoDaoTest.java

```
        // 1개의 Todo 얻기
        TodoVO vo = dao.getTodo(5L);
        System.out.println(vo);

        // Todo 수정하기
        vo.setDone(true);
        dao.updateTodo(vo);

        // Todo 삭제하기
        dao.deleteTodo(5L);

    }
}
```