

DB연동 구조 2





DBMS연동, 객체반환, 트렌젝션 처리 등 중복 코드를 새로운 클래스에서 구동될 수 있게 연동 구조 재설계(싱글톤 패턴 적용)

singleton : 객체 사용시 새로운 객체를 계속 생성해서 사용하는 것이 아니라 하나의 객체만 생성하여 공유하여 사용하는 것을 말함.

Model

- Service

Connection class관리 및 객체 반환, 트렌젝션관리

Common

class 내부의 중복 코드를 처리하는 클래스

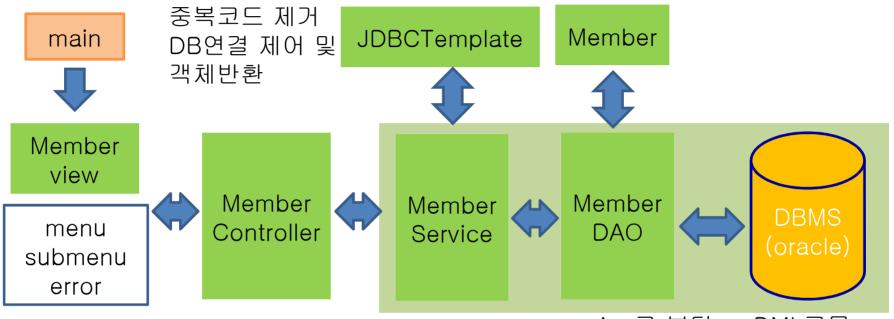
Connection 생성, Connection/Statement/PreparedStatement

반환 메소드, 트렌젝션(commit. rollback)

JDBC



class 구조(2)

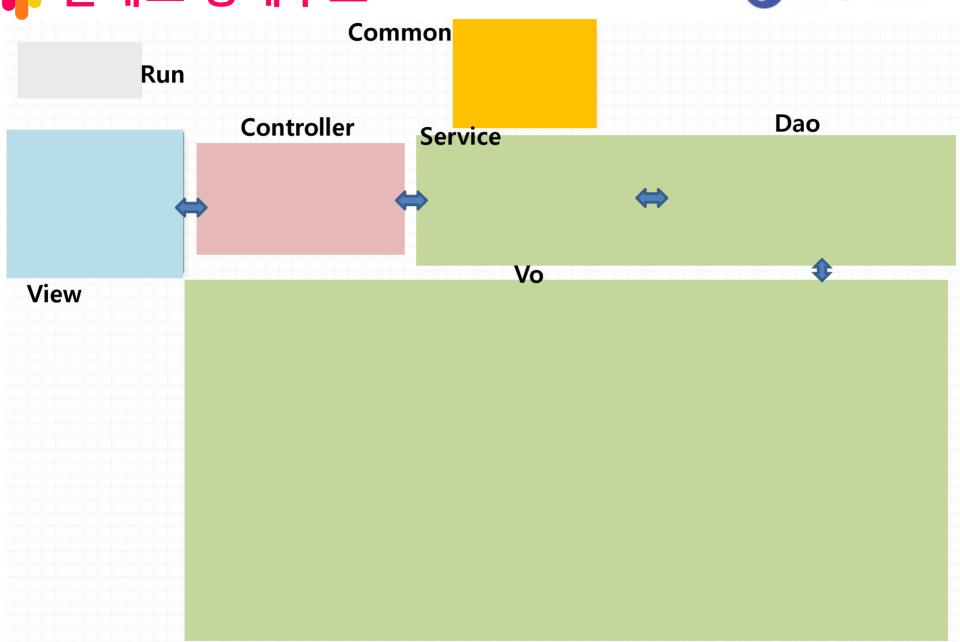


Service에서 받은 처리결과를 Application에 맞춰서 데이터변경 및 Client에게 보여 줄 View 선택 DB연결 정보 DAO전달 및 반환 Controller에 결과 전송 service로 부터 받은 DB 연결 객체로 요청 쿼리문을 DBMS에 전달하고 결과값 service에 전달

DML구문 처리후 결과값 전송

클래스 상세구조





클래스 상세구조

+setEnrollDate(enrollDate: Date): void



Common **JDBCTemplate** (from common) Main (from run) +getConnection(): Connection Run +close(con: Connection): void +main(args: String[*]): voic +close(stmt: Statement): void +close(rset: ResultSet): void +commit(con: Connection): void +rollback(con: Connection): void Dao Controller Service MemberView (from view) MemberDao MemberService MemberController (from dao) -sc: Scanner (from controller) (from service) -mcontroller: MemberController -prop: Properties «constructor»+MemberController() «constructor»+MemberService() «constructor»+MemberView() «constructor»+MemberDao() +selectAll(): void +selectList(): Member[*] +mainMenu(): void +selectList(con: Connection); Member[*] +selectMember(id: String): void +selectOne(id: String): Member -updateMember(): Member +selectOne(con: Connection, id: String): Member +selectNameSearch(inputName: String): void +selectNameSearch(inputName: String): Member[*] -inputMember(): Member +selectNameSearch(con: Connection, inputName: String); Member[*] +insertMember(m: Member): void +insertMember(m; Member); int +insertMember(con: Connection, m: Member): int -inputName(): String +updateMember(m: Member): void +updateMember(m: Member): int +inputMemberld(): String +updateMember(con: Connection, m: Member): int +deleteMember(id: String): void +deleteMember(id: String): int +displayMemberList(list: Member[*]): void +deleteMember(con: Connection, id: String): int +displayMember(m: Member); void +displayError(message: String): void Vo View Member

(from vo) -serialVersionUID: long = 10000000L {readOnly} -memberld: String -memberPwd: String -memberName: String -gender: char -age: int -email: String -phone: String address: String -hobby: String -enrollDate: Date «constructor»+Member() «constructor»+Member(memberld: String, memberPwd: String, memberName: String, gender: char, age: int, email: String, phone: String, address: String, hobby: String, enrollDate: Date) +getMemberId(): String +setMemberld(memberld; String): void +getMemberPwd(): String +setMemberPwd(memberPwd: String): void +getMemberName(): String +setMemberName(memberName: String): void +getGender(): char +setGender(gender: char): void +aetAge(): int +setAge(age: int): void +getEmail(): String +setEmail(email: String): void +getPhone(): String +setPhone(phone: String): void +getAddress(): String +setAddress(address: String): void +getHobby(): String +setHobby(hobby: String): void +getEnrollDate(): Date





실습 1

- 구조2(싱글톤)를 활용한 DB연동

실습 2

- Properties 이용 DBMS접속, SQL문 작성