

# 0406

## JDBC SETTING

- 진행 순서
  - view > service > dao > db
- 

## ProjectRun

- 프로그램이 시작되면 가장 먼저 실행되는 Main 클래스
  - 시작 메뉴화면을 보여주는 MainView를 호출한다
- 

## MainView

- 메인 화면이다 사용자에게 최초의 화면과 기능 구성들을 보여줘 입력받는다
  - 사용자에게 원하는 기능을 입력받아 해당하는 메소드를 호출한다.
- 

## MemberView

- 메인 화면에서 호출한 메서드를 실행
  - 이 때 필요한 데이터를 입력받아 우선 Service 클래스에 전달한다
- 

## MemberService

- 멤버 뷰 클래스에서 호출 한 메서드를 실행한다
  - 이때 필요에 의해 데이터를 가공한다
  - 가공이 필요한 정보는 가공을 해 DB와 연결하기 위한 클래스인 DAO에 전달하고
    - 이때 Connection 객체와 필요한 자료형의 객체 클래스를 같이 전달해야 한다.
  - Connection을 이 클래스에서 열고 닫는다.
- 

## MemberDAO

- DB와 연결하기 위한 클래스
  - Connection 은 Service에서 전달 받고 Statement 와 Rs만 직접 생성한다.
  - 이 때 SQL 명령문도 XML 파일에서 읽어 와 매우 편하고 간단하게 사용 할 수 있다.
- 

## Member(vo)

- 접근하는 DB이 자료형에 맞게 구성해야한다

- 필수적으로 선언이 필요 : 매개변수들(DB와 같은 자료형) , 기본 생성자 ,  
getter /setter

## 오늘 배운 것들 중 기억해야 하는 것들

- 반환 할 때 오류로 대입되지 않을 가능성이 있으므로 전역변수로 NULL으로 먼저 선언해놔 에러가 나도 NULL값으로 인지할수 있게 해준다

- ```
private Member loginMember = null;
```

- Throws 로 호출한 메소드로 예외 처리를 던진다

```
/**회원가입 DAO
public int signUp(Connection conn, Member signUpMember)throws Exception {

/** 로그인 메서드
public Member login(Connection conn, Member mem)throws Exception{
```

- Service 클래스에서 Connection 을 먼저 생성하고 DB와 연결하는 DAO 클래스에 보낼때 같이 보낸다.

```
//1) Connection 객체 생성
Connection conn = getConnection();

//2) 회원 가입 DAO 메서드를 호출하고 결과 반환
int result = dao.signUp(conn,signUpMember);
```

- RS로 행을 가져온 후 get /set 하기

```
loginMember = new Member();
loginMember.setMemberNo(rs.getInt("MEMBER_NO"));
loginMember.setMemberId(rs.getString("MEMBER_ID"));
loginMember.setMemberName(rs.getString("MEMBER_NM"));
loginMember.setMemberGender(rs.getString("MEMBER_GENDER").charAt(0));
loginMember.setEnrollDate(rs.getDate("ENROLL_DATE"));
```

- 헷갈리니 잘 알아보자
- 프로그램이 종료된 후 사용된 JDBC 객체를 반환한다 이때 ! conn은 매개로 전달 받기 때문에 생성한 클래스에서 닫을 것이다.

- Static으로 설정된 메서드를 클래스의 최상단에서 Import 시킨다.

```

6 import static edu.kh.jdbc.common.JDBCTemplate.getConnection;
7 import static edu.kh.jdbc.common.JDBCTemplate.close;
8 import static edu.kh.jdbc.common.JDBCTemplate.commit;
9 import static edu.kh.jdbc.common.JDBCTemplate.rollback;

```

- 상위에 static 으로 메서드 생성 시
- 호출한 메서드를 바로 사용 가능하다.

```
close(conn);
```

- DAO(DATA ACCESS OBJECT) 에 원래는 SQL을 각각 적어야 하지만 XML파일을 읽어와 XML파일에 있는 SQL문을 읽어온다

```

private Properties prop = null;
//MAP K/V 모두 String, 외부 xml파일 입출력에 특화

```

Properties 변수를 생성 해 sql을 불러올 준비를 한다.

```

public MemberDAO() {
    //MemberDAO 객체 생성 시
    //member-sql.xml파일의 내용을 읽어와
    //Properties 객체 생성
    try {
        prop = new Properties();
        prop.loadFromXML(new FileInputStream("member-sql.xml"));
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

1. MemberDAO 객체 생성시 바로 XML 파일을 load해온다
2. 이때 try - catch로 file not found 예외를 처리해야한다

```

//1) SQL 작성 (Properties에 저장된 sql 얻어오기)
String sql = prop.getProperty("signUp");

```

- 불러온 xml파일에서 key값만 찾아 호출해 온다.

회원 가입을 위한 메서드 순서(SignUP)

- 1) Connection 객체 생성
- 2) 회원 가입 DAO 메서드를 호출하고 결과 반환

3)DAO 수행 결과에 따라 트랜잭션 처리

4)사용한 Connection 반환

5)수행 결과 View 반환

회원 가입을 위한 메서드 순서(DAO)

1) SQL 작성 (Properties에 저장된 sql 얻어오기)

2) Prepared Statement 객체 생성(Connection , sql 필요)

3) 위치 홀더 '?' 에 알맞은 값 세팅

4) sql 수행후 결과 반환 받기

5) 사용한 JDBC 자원 반환

---

로그인을 위한 메서드 순서 (Service)

1) Connection 생성

2) DAO 메서드 호출

select 라 트랜잭션 제어 처리 X

3) 수행후 Connection 반환

4) DAO 조회 결과 return

로그인을 위한 메서드 순서(DAO)

1) sql 작성

2) pstmt 객체 생성

3) 위치 홀더 매칭

4)sql 수행 후 결과 반환 (Result Set)받기 (rs변수에)

5) if or while을 이용해 rs에 한행씩 접근하여 원하는 값 얻어오기

6)원하는 컬럼 값을 이용해 member객체를 생성하여 loginMember 변수에 저장

장

7) Connection 을 제외한 객체 자원 반환