



1544-9970
전국대표문의전화(연중무휴)



IT·게임·영상
지점별 위치 찾기

[국비지원과정](#)[교육원소개](#)[취업지원센터](#)[프로젝트](#)[커뮤니티](#)[산학연계](#)[마이페이지](#)

김현지

(디지털컨버전스)공공데이터융합자바개발자양성과정A30

[강남 H] 2024. 11. 27 ~ 2025. 05. 15 09:00~18:00

김시연 강사 | 이은규 취업담임

남은 시간 11: 59: 21

수강생 평가

* 첨부파일의 확장자를 소문자로 등록하세요. - 예시 : test.jpg(O), test.JPG(X)

* 평가 답안에 윈도우 이모티콘 입력 불가

[NCS전공교과] 인터페이스 구현 (문제해결시나리오)

1회차

총점 : 100.0

1.2001020212_23v6.1 인터페이스 설계서 확인하기

2001020212_23v6.2 인터페이스 기능 구현하기

2001020212_23v6.3 인터페이스 구현 검증하기

점수 (100.0점)

올해 취업에 성공한 팔용자씨는 공부하며 정리한 내용을 보고 jsp, servlet을 이용하여 웹어플리케이션을 구현하려고 한다. 팔용자씨는 배웠던 내용을 상기하면서 정리한 내용을 보았다. 하지만, 부족한 부분이 있어 혼자 힘으로 하기가 어려웠다.

아래는 팔용자씨의 개발 순서이다.

<팔용자의 개발 순서>

0. 테이블을 만들기 위한 Query문은 다음과 같다.

```
CREATE TABLE MEMBER(
```

```
USER_NO NUMBER PRIMARY KEY,
```

```
USER_ID VARCHAR2(20) NOT NULL,
```

```
USER_PW VARCHAR2(20) NOT NULL,
```

```
USER_NAME VARCHAR2(20) NOT NULL,
```

```
USER_ADDR VARCHAR(50) NOT NULL,
```

```
REG_DATE DATE DEFAULT SYSDATE
```

```
);
```

```
CREATE SEQUENCE SEQ_MEMBER;
```

1. Oracle과 java를 연동하기 위해 라이브러리를 프로젝트에서 import 하기 위하여 라이브러리 파일을 프로젝트의 루트 폴더에 두었다.

2. JdbcTemplate 클래스를 만들어 공통모듈을 구성하는 과정에서 getConnection, close 메소드 총 2개를 작성하였다.

3. 생성한 테이블에 회원을 추가할 수 있는 Query문을 mapper에 작성하였다.

이때 사용한 쿼리문=> (insert member values(1,'user01','pass01','홍길동');

4. Service 클래스에서 공통으로 만들어 놓은 JdbcTemplate 이용하여 연결을 생성하였다.

5. DAO 클래스를 작성하고 Service에서 넘겨준 Connection을 이용하여 Database에서 쿼리문을 실행하였지만,

데이터가 insert 되지 않았고,

두번째 데이터는 unique 제약조건 위반 되어, 데이터 insert 시도조차 되지 않았다.

친구의 도움을 받아 쿼리 변경에 성공하였지만, 실제 DB에는 반영되지 않았다.

팔용자씨의 프로그램의 오류 원인을 분석하고 이에 대한 해결책을 각각 [원인]과 [조치내용] 란에 작성하시오.

(모든 원인에 대한 조치내용이 작성되어야 하며, 원인과 조치내용은 개수가 여러 개일 수 있습니다)

1. Oracle java를 연동하기 위해 라이브러리를 프로젝트에서 import 할 때 WebContent 폴더 안의 lib 폴더에 두어야 한다.

2. JdbcTemplate 클래스에서 공통 모듈 구성하는 과정에서 getConnection, commit, rollback, close, ResultSet 메소드 작성이 필요하다.

3. 생성한 테이블에 회원을 추가할 수 있는 쿼리문이 틀렸다. SQL구문과 같게 작성되어야 한다.

```
<?xml version="1.0" encoding="UTF-8"?>
<properties>
<entry key="insertMember">
```

```
INSERT
INTO MEMBER
(
    USER_NO
    ,USER_ID
    ,USER_PW
    ,USER_NAME
    ,USER_ADDR
)
VALUES
(
    SEQ_MEMBER.NEXTVAL
    ,?
    ,?
    ,?
    ,?
)
</entry>
</properties>
```

4. Service 클래스에서 공통으로 만든 JdbcTemplate 이용하여 연결을 생성한 후 커밋과 롤백을 처리한다.

5. MemberDao 클래스 작성 후 Service에서 넘겨준 Connection을 이용하여 Database에서 쿼리문을 실행한다.

답안철삭

모범답안

1. 컨트롤러를 정상적으로 등록하지 않았다
2. 컨트롤러의 공용 주소를 부여하지 않았다
3. 교차 출처 자원 공유(CORS)를 허용하지 않았다
4. 데이터베이스 모듈을 주입하지 않았다

5. 목록 매핑에 주소를 부여하지 않았다

수강생 답
(조치내용)

조치내용

1. Oracle과 java를 연동하기 위해 라이브러리를 프로젝트에서 import한다.

ojdbc6.jar 파일을 **WebContent/WEB-INF/lib/ojdbc6.jar** 위치할 수 있도록 import 한다.

2. **JdbcTemplate** 클래스를 만들어 공통모듈을 구성한다. **getConnection, commit, rollback, close, ResultSet** 메소드 작성

```
public class JdbcTemplate {
```

```
    public static void commit(Connection conn) {  
        try {  
            if(conn != null && !conn.isClosed()) {  
                conn.commit();  
            }  
        } catch (SQLException e) {  
            e.printStackTrace();  
        }  
    }
```

```
    public static void rollback(Connection conn){  
        try {  
            if(conn !=null&&!conn.isClosed()){  
                conn.rollback();  
            }  
        } catch (SQLException e){  
            e.printStackTrace();  
        }  
    }
```

```
    public static void close(Connection conn){  
        try {  
            if(conn !=null&&!conn.isClosed()){  
                conn.close();  
            }  
        } catch (SQLException e){  
            e.printStackTrace();  
        }  
    }
```

```
    public static void close(Statement stmt){  
        try {  
            if(stmt!=null&&!stmt.isClosed()){  
                stmt.close();  
            }  
        } catch (SQLException e){  
            e.printStackTrace();  
        }  
    }
```

```
    public static void close(ResultSetrset){  
        try {  
            if(rset!=null&&!rset.isClosed()){  
                rset.close();  
            }  
        } catch (SQLException e){  
            e.printStackTrace();  
        }  
    }
```

3. **member-mapper.xml** 파일을 생성하여 회원을 추가할 수 있는 쿼리문 작성한다.

Member 클래스

```
package kh.member.model.vo;  
import java.sql.Date;
```

```
public class Member {
```

```
    private int userNo;  
    private String userId;  
    private String userPw;  
    private String userAddr;  
    private Date regDate;
```

```

public Member() {}

public Member(int userNo, String userId, String userPw, String userAddr, Date regDate) {
    super();
    this.userNo = userNo;
    this.userId = userId;
    this.userPw = userPw;
    this.userAddr = userAddr;
    this.regDate = regDate;
}

public int getUserNo() {
    return userNo;
}

public void setUserNo(int userNo) {
    this.userNo = userNo;
}

public String getUserId() {
    return userId;
}

public void setUserId(String userId) {
    this.userId = userId;
}

public String getUserPw() {
    return userPw;
}

public void setUserPw(String userPw) {
    this.userPw = userPw;
}

public String getUserAddr() {
    return userAddr;
}

public void setUserAddr(String userAddr) {
    this.userAddr = userAddr;
}

public Date getRegDate() {
    return regDate;
}

public void setRegDate(Date regDate) {
    this.regDate = regDate;
}

@Override
public String toString() {
    return "Member [userNo=" + userNo + ", userId=" + userId + ", userPw=" + userPw + ", "
    userAddr=" + userAddr
        + ", regDate=" + regDate + "]";
}
}

```

member-mapper.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<properties>
<entry key="insertMember">

```

```

INSERT
INTO MEMBER

```

```

(
    USER_NO
    ,USER_ID
    ,USER_PW
    ,USER_NAME
    ,USER_ADDR
)
VALUES
(

```

```

        SEQ_MEMBER.NEXTVAL
        ,?
        ,?
        ,?
        ,?
    )
</entry>
</properties>

```

4. Service 클래스에서 공통으로 만든 `JdbcTemplate` 이용하여 연결을 생성한 후 커밋과 롤백을 처리한다.

```

public int insertMember(Member m){
    Connection conn = JdbcTemplate.getConnection();
    int result = new MemberDao().insertMember(conn,m);

    if(result>0){
        JdbcTemplate.commit(conn);
    }else {
        JdbcTemplate.rollback(conn);
    }
    JdbcTemplate.close(conn);
    return result;
}
}

```

5. DAO 클래스를 작성하고 Service에서 넘겨준 `Connection`을 이용하여 Database에서 쿼리문을 실행한다.

해당 Dao에는 쿼리문의 키값과 메소드명이 동일하게 작성되어야 하며 예외처리 후 `close(pstmt);` 로 닫아주고 `return result;` 로 결과값을 반환하여 service에서 `commit`과 `rollback` 처리를 한다.

```

public class MemberDao {
    public int insertMember(Connection conn,Member m){
        int result = 0;
        PreparedStatement pstmt = null;
        String sql = prop.getProperty("insertMember");
        try {
            pstmt = conn.prepareStatement(sql);
            pstmt.setString(1,m.getUserId());
            pstmt.setString(2,m.getUserPw());
            pstmt.setString(3,m.getUserName());
            pstmt.setString(4,m.getUserAddr());
            result = pstmt.executeUpdate();
        } catch (SQLException e){
            e.printStackTrace();
        }finally {
            JdbcTemplate.close(pstmt);
        }
        return result;
    }
}

```

답안첨삭

모범답안

1. 클래스 위에 `@RestController` 작성
2. 클래스 위에 `@RequestMapping("/item")` 작성
3. 클래스 위에 `@CrossOrigin` 작성
4. DAO 위에 `@Autowired` 작성
5. 메소드 위에 `@RequestMapping("/list")` 작성
(`@GetMapping` 또는 `@PostMapping` 가능)

성취기준		채점기준
평가항목	100 ~ 75	원인에 대하여 모두 파악하였으며 상황에 알맞은 해결 방법 및 소스코드를 명확하게 제시함
	75 ~ 50	원인에 대하여 대부분 규명하였으나, 문제 조치를 위한 해결 방법 및 소스코드를 제시함
	50 ~ 25	원인에 대해 규명하였으나, 약간 미흡하고 문제 조치를 위한 해결 방법 및 소스코드를 제시함
	0 ~ 25	원인에 대해 규명하였으나, 미흡하고 문제 조치를 위한 해결 방법 및 소스코드 제시도 미흡함

평가 메인으로