
프로그래밍 언어 개발

MEMBER 테이블에 대한 CRUD 프로그램 개발

김이화

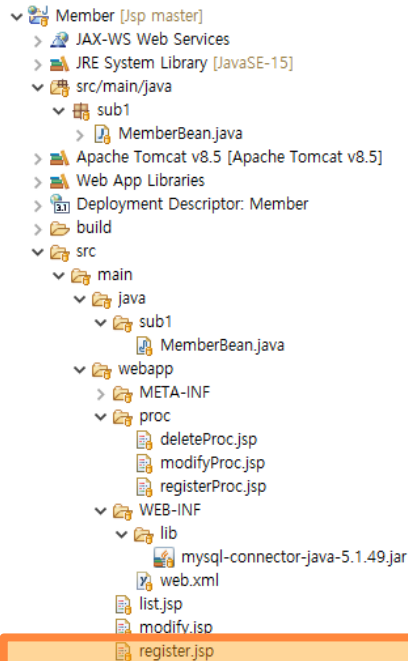
-
- 1 INSERT 기능 및 화면 구현
 - 2 SELECT 기능 및 화면 구현
 - 3 UPDATE 기능 및 화면 구현
 - 4 DELETE 기능 구현

INSERT 기능 및 화면 구현

01

INSERT 기능 및 화면 구현

화면 구현 register.jsp



직원등록

[목록보기](#)

아이디	아이디 입력
이름	이름 입력
휴대폰	휴대폰 입력
직급	사원 ▼
부서	영업1부 ▼

• 목록보기

SELECT 작업을 수행하는 문서(list.jsp)로 이동

```
<a href="./list.jsp">목록보기</a>
```

• 데이터 입력란

MEMBER 테이블에 INSERT할 데이터 정보 입력: 아이디(uid), 이름(name), 휴대폰(hp), 직급(pos), 부서(dep)

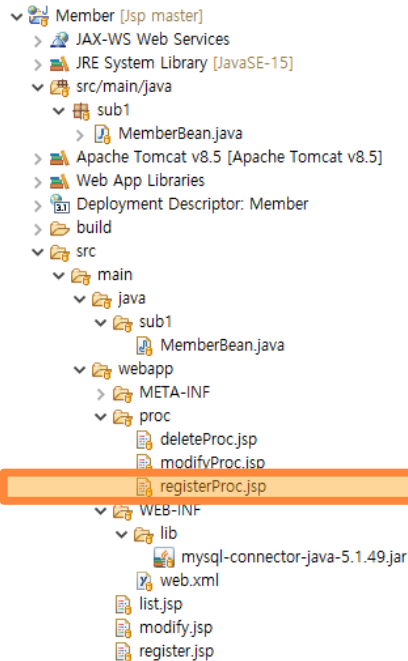
• 직원등록

form에 입력한 정보를 INSERT 기능이 구현된 문서(registerProc.jsp)로 submit한다.
(보안을 위해 post 방식으로 전송)

```
<form action="./proc/registerProc.jsp" method="post">
```

INSERT 기능 및 화면 구현

기능 구현(1/2) registerProc.jsp



- 전송 데이터 인코딩(한글 깨짐 방지) 및 수신

```
// 전송 데이터 인코딩
request.setCharacterEncoding("UTF-8");

// 전송 데이터 수신
String uid  = request.getParameter("uid");
String name = request.getParameter("name");
String hp   = request.getParameter("hp");
String pos  = request.getParameter("pos");
String dep  = request.getParameter("dep");
```

- 데이터베이스 처리 1~6단계 수행

```
// 데이터베이스 처리 1 ~ 6 단계
String host = "jdbc:mysql://192.168.10.114:3306/keh";
String user = "keh";
String pass = "1234";
```

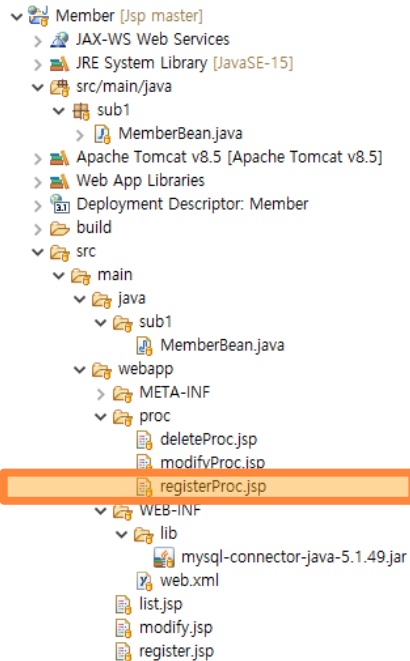
실제로 사용할 데이터베이스의 정보로 수정한다.

```
// 1단계 - JDBC 드라이버 로드
Class.forName("com.mysql.jdbc.Driver");

// 2단계 - 데이터베이스 접속
Connection conn = DriverManager.getConnection(host, user, pass);
```

INSERT 기능 및 화면 구현

기능 구현(2/2) registerProc.jsp



- 데이터베이스 처리 1~6단계 수행(계속)

```
// 3단계 - SQL 실행 객체 생성
Statement stmt = conn.createStatement();

// 4단계 - SQL 실행
String sql = "INSERT INTO `MEMBER` VALUES"
    + "(" + uid + ", " + name + ", " + hp + ", "
    + " " + pos + ", " + dep + ", NOW());";
stmt.executeUpdate(sql);

// 5단계 - 실행 결과 처리(SELECT일 경우)

// 6단계 - 데이터베이스 종료
stmt.close();
conn.close();
```

MEMBER 테이블에 입력해야
하는 데이터 중 입사일(데이
터를 INSERT한 시간)을 입력
하기 위한 코드

INSERT 작업이므로 생략

- INSERT 작업이 수행된 MEMBER 테이블을 확인하기 위해 SELECT 작업을 수행하
는 문서(list.jsp)로 리다이렉트

```
// 리다이렉트
response.sendRedirect("../list.jsp");
```

SELECT

기능 및 화면 구현

SELECT 기능 및 화면 구현

화면 구현 list.jsp

- ▼ Member [Jsp master]
 - > JAX-WS Web Services
 - > JRE System Library [JavaSE-15]
 - ▼ src/main/java
 - ▼ sub1
 - MemberBean.java
 - > Apache Tomcat v8.5 [Apache Tomcat v8.5]
 - > Web App Libraries
 - > Deployment Descriptor: Member
 - > build
 - ▼ src
 - ▼ main
 - ▼ java
 - ▼ sub1
 - MemberBean.java
 - ▼ webapp
 - > META-INF
 - ▼ proc
 - deleteProc.jsp
 - modifyProc.jsp
 - registerProc.jsp
 - ▼ WEB-INF
 - ▼ lib
 - mysql-connector-java-5.1.49.jar
 - web.xml
 - list.jsp**
 - modify.jsp
 - register.jsp

직원목록

등록하기 INSERT 작업을 수행하는 문서(register.jsp)로 이동

아이디	이름	휴대폰	직급	부서	입사일	관리
a101	박혁거세	010-1234-1001	부장	101	21-03-25	수정 삭제
a102	김유신	010-1234-1002	차장	101	21-03-24	수정 삭제
a103	김춘추	010-1234-1003	사원	101	21-03-25	수정 삭제
a104	장보고	010-1234-1004	대리	102	21-03-24	수정 삭제
a105	강감찬	010-1234-1005	과장	102	21-03-24	수정 삭제
a106	이성계	010-1234-1006	차장	103	21-03-24	수정 삭제
a107	정철	010-1234-1007	차장	103	21-03-24	수정 삭제
a108	이순신	010-1234-1008	부장	104	21-03-24	수정 삭제
a109	허균	010-1234-1009	부장	104	21-03-24	수정 삭제
a110	정약용	010-1234-1010	사원	105	21-03-24	수정 삭제
a111	박지원	010-1234-1011	사원	105	21-03-24	수정 삭제
b101	을지문덕	010-5555-1234	사장	107	21-03-24	수정 삭제

MEMBER 테이블의
모든 데이터를 추출
하여 테이블 형태로
화면에 출력

수정 : UPDATE 작업을 수행하는 문서(modify.jsp)로 이동

삭제 : DELETE 작업 수행(deleteProc.jsp)

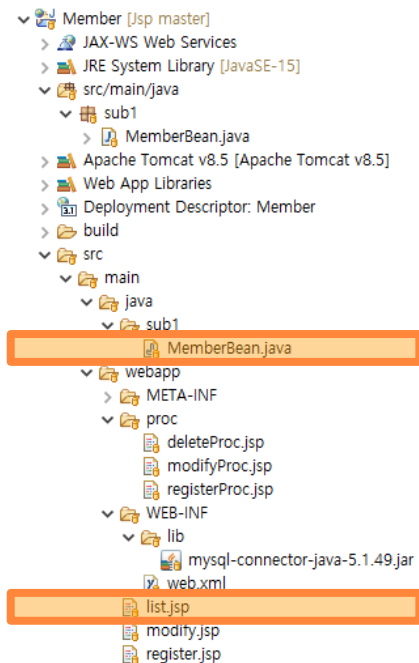
SELECT

기능 및 화면 구현

기능 구현(1/3)

list.jsp

MemberBean.java



- MemberBean 클래스 생성 후 호출(import)

```
<%@page import="sub1.MemberBean"%>
```

- MemberBean.java

- Member 테이블의 칼럼명들을 멤버 변수로 선언

```
package sub1;
```

```
public class MemberBean {
```

```
    private String uid;
    private String name;
    private String hp;
    private String pos;
    private String dep;
    private String rdate;
```

- 각 변수마다 값을 변경 및 호출하기 위한 Getter와 Setter 메서드 생성

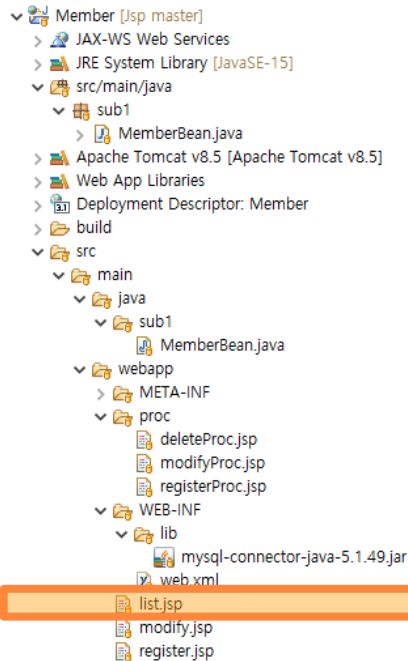
```
    public String getUid() {
        return uid;
    }
    public void setUid(String uid) {
        this.uid = uid;
    }
```

```
    ...
```

SELECT 기능 및 화면 구현

기능 구현(2/3)

list.jsp



• 데이터베이스 처리 1~6단계 수행

1~3단계는 `registerProc.jsp`에서의 코드와 동일

```
// 4단계 - SQL 실행
String sql = "SELECT * FROM `MEMBER`";
ResultSet rs = stmt.executeQuery(sql);

// 5단계 - 실행 결과 처리(SELECT일 경우)
List<MemberBean> mbList = new ArrayList<>();

while(rs.next()){
    MemberBean mb = new MemberBean();

    mb.setUid(rs.getString(1));
    mb.setName(rs.getString(2));
    mb.setHp(rs.getString(3));
    mb.setPos(rs.getString(4));
    mb.setDep(rs.getString(5));
    mb.setRdate(rs.getString(6));

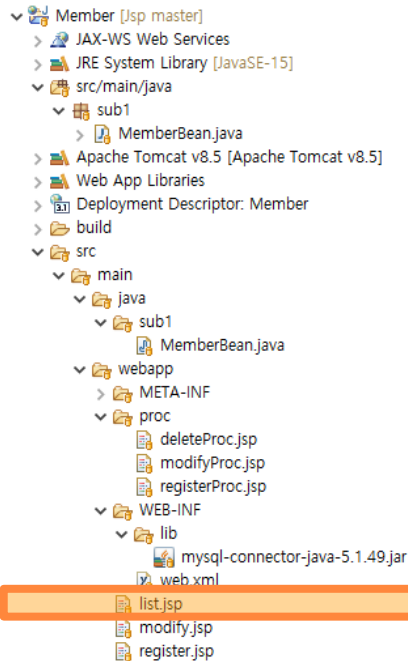
    mbList.add(mb);
}
```

MEMBER 테이블의 모든 데이터를 가져와 리스트 안에 넣는다.

```
// 6단계 - 데이터베이스 종료
rs.close();
stmt.close();
conn.close();
```

SELECT 기능 및 화면 구현

기능 구현(3/3) list.jsp



- 추출한 모든 데이터를 테이블 형태로 출력

```

</tr>
<% for(MemberBean mb : mbList){
    String uid    = mb.getUid();
    String name    = mb.getName();
    String hp      = mb.getHp();
    String pos     = mb.getPos();
    String dep     = mb.getDep();
    String rdate   = mb.getRdate().substring(2, 10);

%>
<tr>
    <td><%= uid %></td>
    <td><%= name %></td>
    <td><%= hp %></td>
    <td><%= pos %></td>
    <td><%= dep %></td>
    <td><%= rdate %></td>
    <td>

```

리스트 안에 넣은 데이터
들을 가져와 테이블 형태
로 출력

UPDATE 작업을 수행하는
문서(modify.jsp)로 아이디
(uid), 이름(name), 휴대폰
(hp) 데이터 전송 후 이동

```

<a href="./modify.jsp?uid=<%= uid %>&name=<%= name %>&hp=<%= hp %>">수정</a>

```

```

<a href="./proc/deleteProc.jsp?uid=<%= uid %>">삭제</a>

```

```

</td>
</tr>
<% } %>

```

DELETE 작업을 수행하는 문서(deleteProc.jsp)로 아이디(uid) 데이터 전송 후 이동

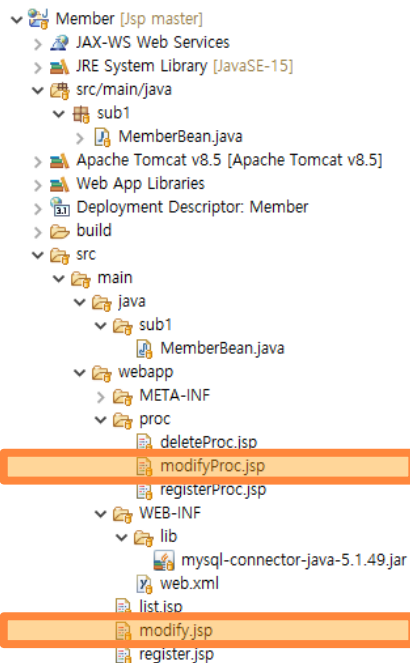
UPDATE

기능 및 화면 구현

UPDATE 기능 및 화면 구현

화면 및 기능 구현

modify.jsp modifyProc.jsp



직원수정

목록보기 →

아이디	a101
이름	박혁거세
휴대폰	010-1234-1001
직급	사원 ▼
부서	영업1부 ▼

직원수정

• 목록보기

SELECT 작업을 수행하는 문서(list.jsp)로 이동

```
<a href="./list.jsp">목록보기</a>
```

• 데이터 입력란

목록 문서(list.jsp)에서 전송한 데이터(아이디, 이름, 휴대폰)를 가져와 출력하고 아이디를 제외한 데이터들은 UPDATE할 내용으로 수정이 가능하게 구현

• 직원수정

form에 수정한 정보를 UPDATE 기능이 구현된 문서(modifyProc.jsp)로 submit한다. (보안을 위해 post 방식으로 전송)

```
<form action="./proc/modifyProc.jsp" method="post">
```

• modifyProc.jsp

데이터베이스 처리 4단계를 제외한 모든 코드는 registerProc.jsp와 동일

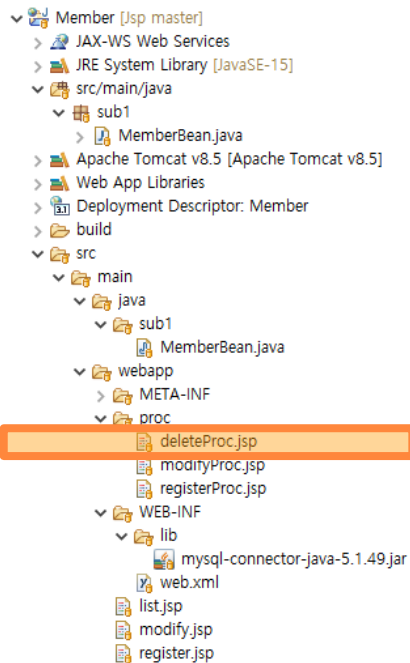
// 4단계 - SQL 실행

```
String sql = "UPDATE `MEMBER` SET ";
sql += "`name`='" + name + "'", ";
sql += "`hp`='" + hp + "'", ";
sql += "`pos`='" + pos + "'", ";
sql += "`dep`='" + dep + "'", ";
sql += "WHERE `uid`='" + uid + "';";
stmt.executeUpdate(sql);
```

DELETE 기능 구현

DELETE 기능 구현

기능 구현 deleteProc.jsp



- 목록 문서(list.jsp)에서 전송받은 아이디(uid) 데이터를 가지고 데이터를 식별하여 삭제

```
// 전송 데이터 수신
```

```
String uid = request.getParameter("uid");
```

```
⋮
```

```
// 4단계
```

```
String sql = "DELETE FROM `MEMBER` WHERE `uid`='"+uid+"'";  
stmt.executeUpdate(sql);
```

```
⋮
```

- 그 외의 코드는 registerProc.jsp와 동일

감사합니다