

# 주소록 프로젝트(오라클DB)



*Address Book*



# 주소록(Address Book) 프로젝트

## 주소록(Address Book) 프로젝트

### 프로그램 소스 목록

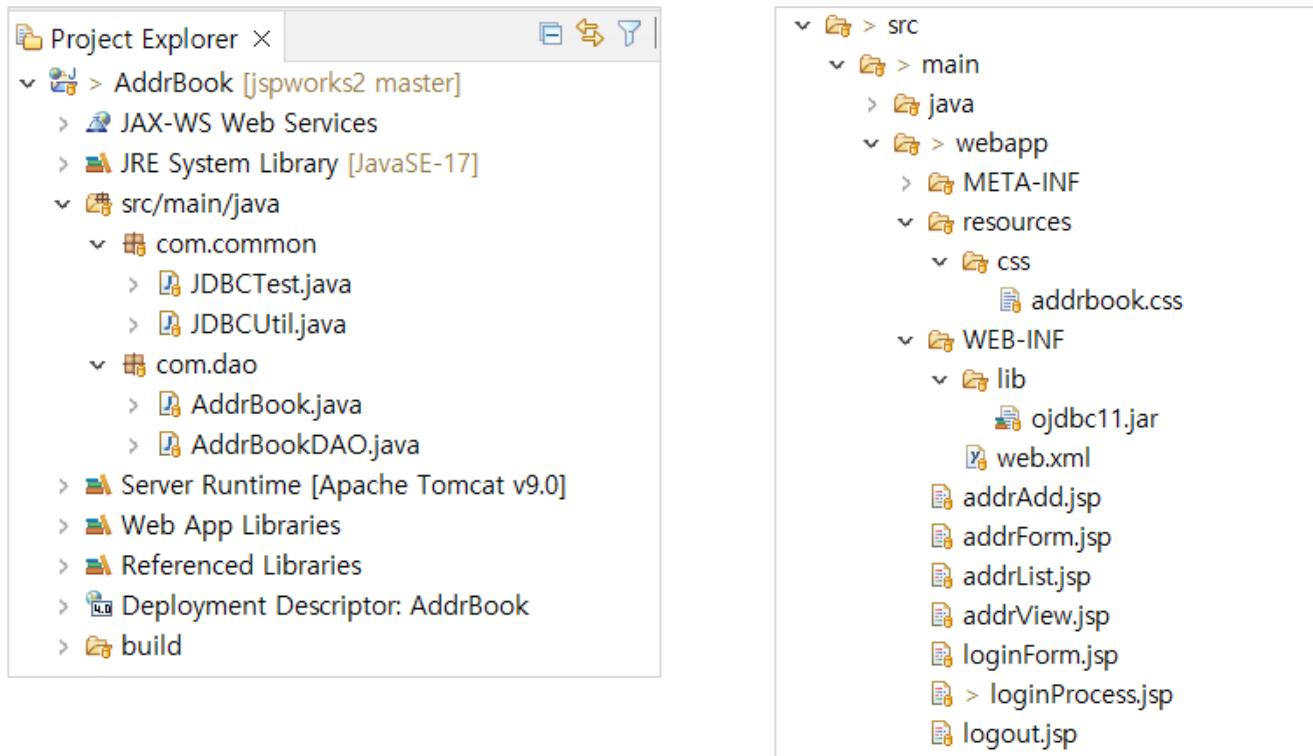
파일 이름	역 할
addrForm.jsp	주소록 등록 양식 화면
addrbook.css	전체 레이아웃, 테이블 등의 스타일을 구현
t_address.sql	주소록을 저장할 db 테이블 파일
addrAdd.jsp	addrForm에 입력된 내용을 AddrBook으로 매핑하고 AddrBookDAO를 이용해 데이터 추가 가능 수행
addrList.jsp	application scope에 저장된 AddrBookDAO 객체를 통해 저장된 데이터를 가져와서 화면에 출력
AddrBook.java	주소록 데이터 매핑을 위한 빈즈 클래스
AddrBookDAO.java	AddrBook을 관리 사용하는 클래스(application 객체 사용)



# 주소록(Address Book) 프로젝트

## 주소록(Address Book) 프로젝트

### 프로젝트 계층도



# 주소록(Address Book) 프로젝트

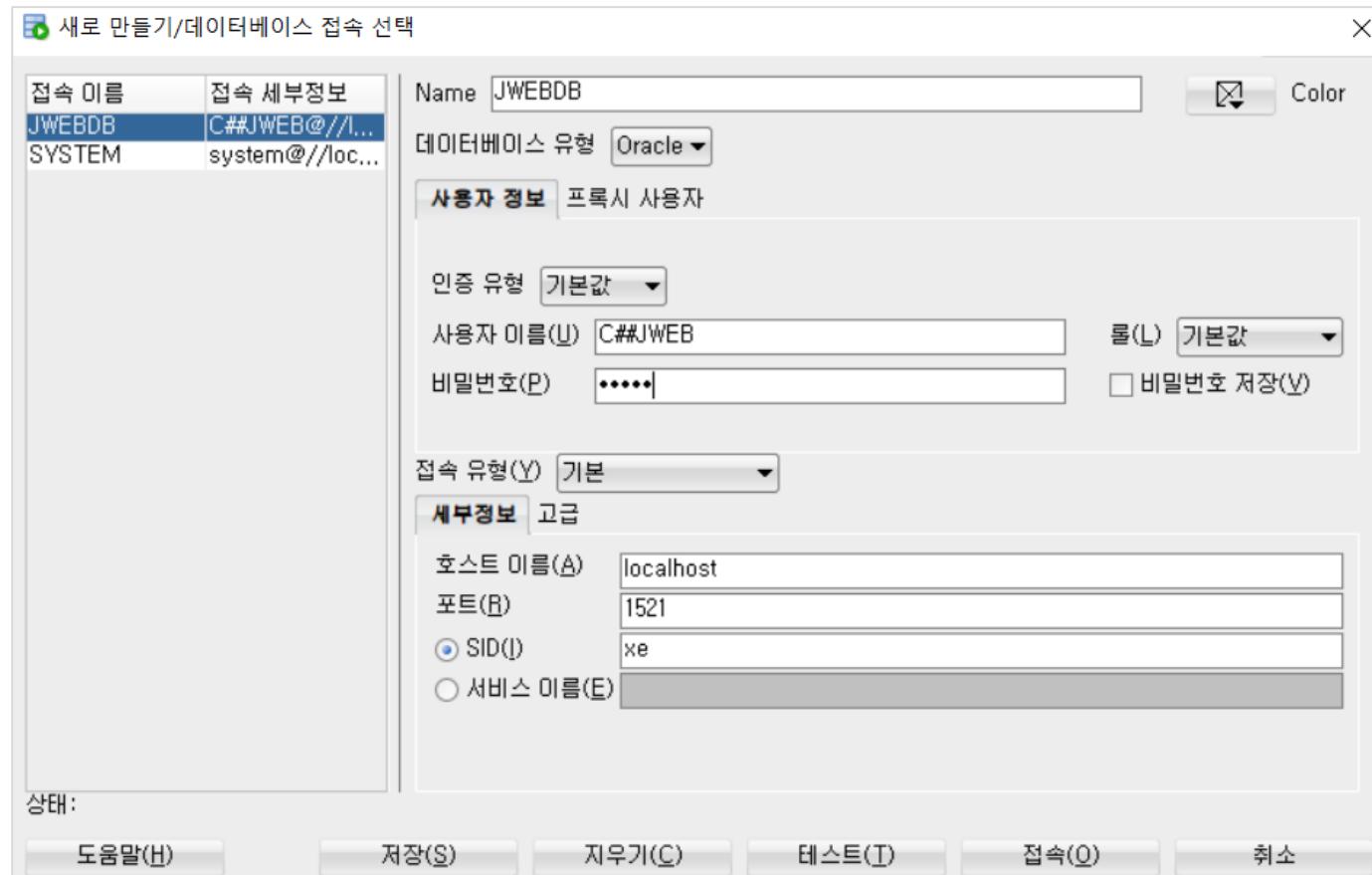
## 오라클 데이터베이스 만들기

```
-- USER(jweb) 생성  
CREATE USER C##jweb IDENTIFIED BY 54321;  
  
-- 세션 권한 부여  
GRANT CREATE SESSION TO C##jweb;  
  
-- 테이블과 테이블 공간 생성  
GRANT CREATE TABLE, RESOURCE TO C##jweb;  
  
-- 테이블 스페이스 권한 주기  
ALTER USER C##jweb DEFAULT TABLESPACE USERS QUOTA UNLIMITED ON USERS;  
  
-- 권한 불충분 에러가 뜰 경우  
-- sqlplus 접속후  
-- GRANT CREATE ANY TABLE TO C##jweb;  
-- GRANT CONNECT, RESOURCE, DBA TO C##jweb];
```



# 주소록(Address Book) 프로젝트

## 오라클 데이터베이스 만들기



# 주소록(Address Book) 프로젝트

## 테이블 만들기

```
-- t_address 테이블 생성
CREATE TABLE t_address(
    num      NUMBER(4),
    username VARCHAR2(20),
    tel      VARCHAR2(20) NOT NULL,
    email    VARCHAR2(30) NOT NULL,
    gender   VARCHAR2(6),
    joinDate DATE DEFAULT SYSDATE,
    PRIMARY KEY(num)
);
-- 일련번호(시퀀스) 생성
CREATE SEQUENCE ab_seq
INCREMENT BY 1
START WITH 1
MINVALUE 1
MAXVALUE 1000
NOCYCLE
NOCACHE;
```

```
INSERT INTO t_address
VALUES (ab_seq.NEXTVAL, '김기용', '010-1234-5678', 'test@test.com', '남', SYSDATE);

SELECT * FROM t_address;

COMMIT;
```



# 주소록(Address Book) 프로젝트

주소록 등록

이름	박마늘
전화번호	010-3333-4444
이메일	garlic@test.com
성별	여 ▼
<input type="button" value="확인"/> <input type="button" value="취소"/>	



박마늘님이 등록되었습니다.

[목록 보기](#)



주소록

[[주소 추가](#)]

번호	이름	성별	가입일	회원 보기
1	이상추	남	2021-03-03	<input type="button" value="보기"/>
2	박마늘	여	2021-03-03	<input type="button" value="보기"/>

# 주소록(Address Book) 프로젝트

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>주소록 프로그램</title>
<link rel="stylesheet" href=".//resources/css/addrbook.css">
</head>
<body>
    <div id="container">
        <h2>주소록 등록</h2>
        <hr>
        <p><a href=".//LoginForm.jsp">[로그인]</a></p>
        <form action=".//addrAdd.jsp" method="post">
            <table id="tbl">
                <tr>
                    <td>이름</td>
                    <td><input type="text" name="username" maxlength=20 size=20></td>
                </tr>
                <tr>
                    <td>전화번호</td>
                    <td><input type="text" name="tel" maxlength=20 size=20></td>
                </tr>
            </table>
        </form>
    </div>
</body>
</html>
```

addrForm.jsp



# 주소록(Address Book) 프로젝트

```
<tr>
    <td>이메일</td>
    <td><input type="text" name="email" maxlength=20 size=20></td>
</tr>
<tr>
    <td>성별</td>
    <td>
        <select name="gender">
            <option selected>남</option>
            <option>여</option>
        </select>
    </td>
</tr>
<tr>
    <td colspan="2">
        <input type="submit" value="확인">
        <input type="reset" value="취소">
    </td>
</tr>
</table>
</form>
```



# 주소록(Address Book) 프로젝트

```
1 @charset "UTF-8";
2 /* 전체 스타일 */
3 #container{width: 80%; margin: 0 auto; text-align: center; margin-top: 50px;}
4
5 /* 주소 등록 스타일*/
6 #tbl{width: 400px; margin: 0 auto;}
7 #tbl, td{border: 1px solid #ccc; border-collapse: collapse;}
8 #tbl td{width: 80px; height: 40px;}
9 #tbl input{height: 25px;}
10
11 /* 목록 보기 스타일*/
12 #tbl_list{width: 600px; margin: 0 auto;}
13 #tbl_list, td{border: 1px solid #ccc; border-collapse: collapse;}
14 #tbl_list td{height: 40px;}
15
16 /* 로그인 스타일*/
17 #tbl_login{width: 400px; margin: 0 auto;}
18 #tbl_login, td{border: 1px solid #ccc; border-collapse: collapse;}
19 #tbl_login td{height: 50px;}
20 #tbl_login input[type="text"]{height:25px;}
21 #tbl_login td:nth-child(1){width: 100px;}
22
23 /* 상세 보기 스타일*/
24 #tbl_view{width: 400px; margin: 0 auto;}
25 #tbl_view, td{border: 1px solid #ccc; border-collapse: collapse;}
26 #tbl_view td{height: 40px;}
```

addrbook.css



# 주소록(Address Book) 프로젝트

```
package com.dao;
import java.io.Serializable;
import java.util.Date;
public class AddrBook implements Serializable{
    private static final long serialVersionUID = 20120808L;
    private int num;
    private String username;
    private String tel;
    private String email;
    private String gender;
    private Date joinDate;

    public int getNum() {
        return num;
    }
    public void setNum(int num) {
        this.num = num;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
}
```

AddrBook.java

```
public String getTel() {
    return tel;
}
public void setTel(String tel) {
    this.tel = tel;
}
public String getEmail() {
    return email;
}
public void setEmail(String email) {
    this.email = email;
}
public String getGender() {
    return gender;
}
public void setGender(String gender) {
    this.gender = gender;
}
public Date getJoinDate() {
    return joinDate;
}
public void setJoinDate(Date joinDate) {
    this.joinDate = joinDate;
}
```



# 주소록(Address Book) 프로젝트

```
package com.common;  
  
import java.sql.Connection;■  
  
public class JDBCUtil {  
    private static String driverClass = "oracle.jdbc.OracleDriver";  
    private static String url = "jdbc:oracle:thin:@localhost:1521:xe";  
    private static String username = "c##jweb";  
    private static String password = "54321";  
  
    //DB 연결 메서드  
    public static Connection getConnention() {  
        try {  
            Class.forName(driverClass);  
            return DriverManager.getConnection(url, username, password);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
        return null;  
    }  
}
```

JDBCUtil.java



# 주소록(Address Book) 프로젝트

```
//DB 연결 종료 메서드
public static void close(Connection conn, PreparedStatement pstmt) {
    if(pstmt != null) {
        try {
            pstmt.close();
        } catch (Exception e) {
            e.printStackTrace();
        }finally {
            pstmt = null;
        }
    }

    if(conn != null) {
        try {
            conn.close();
        } catch (Exception e) {
            e.printStackTrace();
        }finally {
            conn = null;
        }
    }
}
```

JDBCUtil.java



# 주소록(Address Book) 프로젝트

```
//연결 종료(ResultSet이 있는 경우)
public static void close(Connection conn, PreparedStatement pstmt, ResultSet rs) {
    if(rs != null) {
        try {
            rs.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    if(pstmt != null) {
        try {
            pstmt.close();
        } catch (Exception e) {
            e.printStackTrace();
        }finally {
            pstmt = null;
        }
    }
    if(conn != null) {
        try {
            conn.close();
        } catch (Exception e) {
            e.printStackTrace();
        }finally {
            conn = null;
        }
    }
}
```

JDBCUtil.java



# 주소록(Address Book) 프로젝트

```
public class JDBCTest {
    private static String driverClass = "oracle.jdbc.OracleDriver";           //오라클 드라이버
    private static String url = "jdbc:oracle:thin:@localhost:1521:xe";          //db 경로 포트-1521
    private static String username = "c##jweb";                                  //사용자 이름
    private static String password = "54321";

    public static void main(String[] args) {
        //연결 객체 선언
        Connection conn = null;

        try {
            Class.forName(driverClass);
            System.out.println("Oracle 드라이버 로딩");
            conn = DriverManager.getConnection(url, username, password);
            System.out.println("Connection 객체 생성 : " + conn);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            try {
                conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}
```

JDBCTest.java



# 주소록(Address Book) 프로젝트

```
public class AddrBookDAO {  
    // JDBC 관련 변수  
    private Connection conn = null;  
    private PreparedStatement pstmt = null;  
    private ResultSet rs = null;  
  
    // 주소 추가  
    public void addAddress(AddrBook addrBook) {  
  
        try {  
            conn = JDBCUtil.getConnnection();  
            String sql = "INSERT INTO t_address(num, username, tel, email, gender)"  
                + " VALUES (ab_seq.NEXTVAL, ?, ?, ?, ?)";  
            pstmt = conn.prepareStatement(sql);  
            pstmt.setString(1, addrBook.getUsername());  
            pstmt.setString(2, addrBook.getTel());  
            pstmt.setString(3, addrBook.getEmail());  
            pstmt.setString(4, addrBook.getGender());  
            pstmt.executeUpdate();  
        } catch (SQLException e) {  
            e.printStackTrace();  
        } finally {  
            JDBCUtil.close(conn, pstmt);  
        }  
    }  
}
```

AddrBookDAO.java



# 주소록(Address Book) 프로젝트

```
//주소 목록
public ArrayList<AddrBook> getListAll(){
    ArrayList<AddrBook> addrList = new ArrayList<>();
    try {
        conn = JDBCUtil.getConnnection();
        String sql = "SELECT * FROM t_address ORDER BY joinDate";
        pstmt = conn.prepareStatement(sql);
        rs = pstmt.executeQuery();
        while(rs.next()) {
            AddrBook addrBook = new AddrBook();
            addrBook.setNum(rs.getInt("num"));
            addrBook.setUsername(rs.getString("username"));
            addrBook.setTel(rs.getString("tel"));
            addrBook.setGender(rs.getString("gender"));
            addrBook.setJoinDate(rs.getDate("joinDate"));
            addrList.add(addrBook);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        JDBCUtil.close(conn, pstmt, rs);
    }
    return addrList;
}
```

AddrBookDAO.java



# 주소록(Address Book) 프로젝트

```
<% request.setCharacterEncoding("utf-8"); %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>주소 추가</title>
<link rel="stylesheet" href=".//resources/css/addrbook.css">
</head>
<jsp:useBean id="addrBook" class="com.dao.AddrBook" />
<jsp:setProperty property="num" name="addrBook"/>
<jsp:setProperty property="username" name="addrBook"/>
<jsp:setProperty property="tel" name="addrBook"/>
<jsp:setProperty property="email" name="addrBook"/>
<jsp:setProperty property="gender" name="addrBook"/>
<jsp:setProperty property="joinDate" name="addrBook"/>
<jsp:useBean id="abDAO" class="com.dao.AddrBookDAO" scope="application" />
<%
    if(session.getAttribute("sessionId") != null){
        abDAO.addAddress(addrBook);
    }else{
        response.sendRedirect("./loginForm.jsp");
    }
%>
<body>
    <div id="container">
        <h3><%= addrBook.getUsername() %> 님이 등록되었습니다.</h3>
        <a href=".//addrList.jsp">목록 보기</a>
    </div>
```

addrAdd.jsp



# 주소록(Address Book) 프로젝트

```
<%@page import="com.dao.AddrBook"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>주소록 목록</title>
<link rel="stylesheet" href=".//resources/css/addrbook.css">
</head>
<jsp:useBean id="abDAO" class="com.dao.AddrBookDAO" scope="application"/>
<body>
    <div id="container">
        <h2>주소록</h2>
        <hr>
        <p>
            <a href=".//addrForm.jsp">[주소 추가]</a>&ampnbsp;
            <a href=".//LoginForm.jsp">[로그아웃]</a>
        </p>
    </div>
</body>
```

addrList.jsp



# 주소록(Address Book) 프로젝트

```
<table id="tbl_list">
    <tr>
        <td>번호</td><td>이름</td><td>성별</td><td>가입일</td><td>보기</td>
    </tr>
    <%
        for(int i=0; i<abDAO.getListAll().size(); i++){
            AddrBook addrBook = abDAO.getListAll().get(i);
    %>
    <tr>
        <td><%=addrBook.getNum() %></td>
        <td><%=addrBook.getUsername() %></td>
        <td><%=addrBook.getGender() %></td>
        <td><%=addrBook.getJoinDate() %></td>
        <td><a href=".addrView.jsp?num=<%=addrBook.getNum() %>">
            <button type="button">보기</button></a></td>
        </tr>
    <% } %>
</table>
</div>
```

addrList.jsp



# 주소록(Address Book) 프로젝트

## 주소록(Address Book) 구현 – 로그인 기능

### 프로그램 소스 목록

파일 이름	역 할
loginForm.jsp	이메일이 일치하면 로그인을 할 수 있는 화면
loginProcess.jsp	로그인 버튼을 누르면 AddrBookDAO를 이용해 데이터 로그인기능 수행, 세션을 부여함
logout.jsp	세션이 해제됨



# 주소록(Address Book) 프로젝트

로그인

이메일	park@test.com
<input type="button" value="로그인"/> <input type="button" value="취소"/>	



park@test.com으로 로그인되었습니다.  
[\[목록 보기\]](#)



주소록

[주소 추가] [로그아웃]

번호	이름	성별	가입일	보기
1	김기용	남	2022-07-25	<input type="button" value="보기"/>
2	박명숙	여	2022-07-25	<input type="button" value="보기"/>
3	신유빈	여	2022-07-26	<input type="button" value="보기"/>

# 주소록(Address Book) 프로젝트

## AddrBookDAO – 로그인 체크 기능

```
//로그인 체크
public boolean checkLogin(String email) {
    connDB();
    String sql = "SELECT email FROM t_address WHERE email = ?";
    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, email);
        rs = pstmt.executeQuery();
        if(rs.next()) {
            return true; //이메일 일치
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        disconnect();
    }
    return false; //이메일 불일치
}
```



# 주소록(Address Book) 프로젝트

```
<title>로그인</title>
<link rel="stylesheet" href=".//resources/css/addrbook.css">
</head>
<body>
    <div id="container">
        <h2>로그인</h2>
        <hr>
        <form action="LoginProcess.jsp" method="post">
            <table id="tbl_login">
                <tr>
                    <td>이메일</td>
                    <td><input type="text" name="email" maxlength=30 size=30></td>
                </tr>
                <tr>
                    <td colspan="2">
                        <input type="submit" value="로그인"/>
                        <input type="reset" value="취소"/>
                    </td>
                </tr>
            </table>
        </form>
    </div>
```

loginForm.jsp



# 주소록(Address Book) 프로젝트

```
<jsp:useBean id="abDAO" class="com.dao.AddrBookDAO" scope="application" />
<%
    String email = request.getParameter("email");
    boolean result = abDAO.checkLogin(email);

    if(result){
        session.setAttribute("sessionId", email);      //세션 할급(이름 - sessionId)
    }else{
        out.println("<script>");
        out.println("alert('이메일이 일치하지 않습니다.')");
        out.println("history.go(-1)");
        out.println("</script>");
    }
%>
<body>
    <div id="container">
        <h2><%=session.getAttribute("sessionId") %>으로 로그인되었습니다.</h2>
        <p><a href=".addrList.jsp">[목록 보기]</a></p>
    </div>
</body>
```

loginProcess.jsp



# 주소록(Address Book) 프로젝트

## 로그아웃 – 세션 삭제

logout.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"  
    pageEncoding="UTF-8"%>  
<%  
    session.invalidate(); //모든 세션 삭제  
  
    response.sendRedirect("loginForm.html");  
%>
```

