



MVC웹개발 샘플

```
package controller;
```

```
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
@WebServlet("/main.do")
```

```
public class HomeServlet extends HttpServlet {
```

```
    @Override
```

```
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
```

```
        request.getRequestDispatcher("WEB-INF/views/main.jsp").forward(request, response);
```

```
    }
```

```
}
```

```
package controller;
```

```
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
import model.dao.MemberDao;
import model.service.LoginService;
```

```
@WebServlet("/login.do")
```

```
public class LoginServlet extends HttpServlet {
```

```
    @Override
```

```
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
```

```
        //response.getWriter().println("hello get!!!!!!");
        request.getRequestDispatcher("WEB-INF/views/login.jsp").forward(request, response);
```

```
    }
```

```
    @Override
```

```
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
```

```
        // response.getWriter().println(" hello post!!!!!!");
```

```
        String id ="test01";    // String id = request.getParameter("id");
        String pw ="0000";    // String pw = request.getParameter("pw");
        MemberDao dao = new MemberDao();
        LoginService service = new LoginService(dao);
        int result =service.checkUser(id, pw);
        if( result ==1 ) {
            request.getRequestDispatcher("WEB-INF/views/main.jsp").forward(request, response);
        }else if( result ==2 ) {
            String message="아이디없음";
            request.setAttribute("message", message);
            request.getRequestDispatcher("WEB-INF/views/login.jsp").forward(request, response);
        }else if( result ==3 ) {
            String message="비밀번호 오류";
            request.setAttribute("message", message);
            request.getRequestDispatcher("WEB-INF/views/login.jsp").forward(request, response);
        }
    }
```

```

    }
}

package model.dao;
public class MemberDao {
    //연결
    private void dbCon() {
    }

    //
    public int confrimUser(String id, String pw) {
        // 1 : 정상, 2 :아이디없음, 3: 비번 잘못
        return 1;
    }
}

```

```

package model.service;

import model.dao.MemberDao;

public class LoginService {
    MemberDao dao;

    public LoginService() {
    }

    public LoginService( MemberDao memberDao) {
        this.dao = memberDao;
    }

    public int checkUser( String id, String pw) {
        int result =dao.confrimUser(id, pw);
        return result;
    }
}

```

<<view>>: WEB-INF\views\ 위치함 , url요청으로 접근할 수 없는 영역임 , 처리결과로 보여줄 view로서만 사용됨

### login.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

<h2>로그인</h2>
<%
    String message = (String)request.getAttribute("message");
    if( message != null){
        out.println( message);
    }
%>

<form action="/sample/login.do" method="post">
    ID: <input type="text" > <br>
    PW: <input type="text" > <br>
    <button>Login</button>
</form>

</body>
</html>
```

### main.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

main page !!!!!!!
<a href="/sample/login.do">로그인</a>
</body>
</html>
```

MVC 개발패턴 (변경에 유리한 코드작성하겠다 !!! )

<<코드분리: 역할분리, 코드유지보수 용이 >>

데이터베이스관련작업 :DAO 작성자

서비스관련 작업 : Service 작성자 , (DAO아직 미개발일 경우 : 가짜 DAO를 사용함, 매서드는 맞춤,  
말로만 맞추는게 아니라 정확성을 위해서 인터페이스를 하나 둠)

View작성자 : 화면 뷰 작성자

```
interface DAOInterface{
    void insert( Member member);
    ArrayList<Member> members();
    Member member( String id);
    void delete(String id);
    void update( Member member);
}
```

// 가짜 DAO

```
DAOInterfaceImpDumy implements DAOInterface{
    void insert( Member member){
        System.out.println("등록");
    }
    ArrayList<Member> members(){
        System.out.println("전체조회");
    }

    Member member( String id){
        System.out.println("고객조회");
    }
    void delete(String id){
        System.out.println("고객삭제");
    }
    void update( Member member){
        System.out.println("고객변경");
    }
}
```

|

// 진짜 DAO, 데이터베이스연결후 실제 crud 작업을 함

```
DAOInterfaceImpReal implements DAOinterface{
    void insert( Member member){
        System.out.println("진짜등록");
    }
    ArrayList<Member> members(){
        System.out.println("진짜전체조회");
    }

    Member member( String id){
        System.out.println("진짜고객조회");
    }
    void delete(String id){
        System.out.println("진짜고객삭제");
    }
    void update( Member member){
        System.out.println("진짜고객변경");
    }
}
```

|