



# **JSP 8강 – 에러처리, 배포**

---

**양 명 속**

**[[now4ever7@gmail.com](mailto:now4ever7@gmail.com)]**



# 목차

---

- jsp 페이지의 에러 처리
- 메인화면 - 공지사항
- 배포



# jsp 페이지의 에러 처리

---

- 에러 처리의 개요
- 에러 코드별 처리



# 에러 처리의 개요

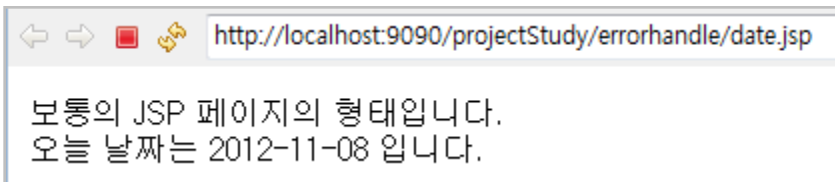
---

- jsp 에서의 에러 메시지
  - 에러가 발생하면 그 에러가 어떠한 경로로 발생하게 되었는지 스택을 뒤집어서 그 경로를 추적하여 에러 메시지 표시
  - 에러가 발생했을 때 에러 메시지를 표시하지 말고 다른 페이지를 보여주자는 것



# 에러 처리의 개요

- 에러 페이지를 사용한 에러 처리 예제
  - 오늘 날짜를 표시하는 예제, 고의로 에러가 발생하도록 해서 에러의 처리를 에러 페이지인 error.jsp 에서 처리하도록
  - 에러가 발생하는 페이지 - date.jsp
  - 에러 페이지 - error.jsp





# date.jsp

---

```
<%@ page contentType="text/html; charset=utf-8" %>
```

```
<%@ page import="java.util.*, java.text.*" %>
```

```
<%@ page errorPage="error.jsp"%>
```

```
<html>
```

```
<body>
```

```
<%
```

```
    Date date = new Date();
```

```
    SimpleDateFormat simpleDate = new SimpleDateFormat("yyyy-MM-dd");
```

```
    String strdate = simpleDate.format(date);
```

```
%>
```

```
    보통의 JSP 페이지의 형태입니다.<br>
```

```
    오늘 날짜는 <%= strdat%> 입니다.
```

```
</body>
```

```
</html>
```

에러가 발생되면 에러를 처리하는  
페이지인 **error.jsp** 에서 처리함

고의로 에러가 발생하도록 선언되지 않은 변수 사용



# error.jsp

```
<%@ page contentType="text/html; charset=utf-8" %>
```

```
<%@ page isErrorPage="true"%>
```

```
<html>
```

```
<head><title>에러 페이지</title></head>
```

```
<body>
```

요청하신 페이지에서 문제가 발생했습니다.

서비스 사용에 불편을 끼쳐드려서 대단히 죄송합니다.

빠른시간내에 문제를 처리하겠습니다.

```
</body>
```

```
</html>
```

에러 페이지를 작성할 때는 **page** 디렉티브의 **isErrorPage** 속성의 값을 **"true"**로 설정해야 함

- 에러 페이지에 의한 에러 처리가 일어나지 못함 – 컨테이너에 따라 에러 페이지에 의한 에러 처리를 지원하지 않을 수 있기 때문
- 보통 에러 처리는 에러 페이지를 사용하지 않고 에러 코드별 처리나 에러 종류별 처리를 사용함



## HTTP Status 500 - Unable to compile class for JSP:

**type** Exception report

**message** Unable to compile class for JSP:

**description** The server encountered an internal error that prevented it from fulfilling this request.

**exception**

org.apache.jasper.JasperException: Unable to compile class for JSP:

An error occurred at line: 12 in the jsp file: /errorhandle/date.jsp  
strdat cannot be resolved to a variable

```
9:      String strdate = simpleDate.format(date);
10:      %>
11:      보통의 JSP 페이지의 형태입니다.<br>
12:      오늘 날짜는 <%= strdat%> 입니다.
13:  </body>
14: </html>
15:
```

Stacktrace:

```
org.apache.jasper.compiler.DefaultErrorHandler.javacError(DefaultErrorHandler.java:92)
org.apache.jasper.compiler.ErrorDispatcher.javacError(ErrorDispatcher.java:330)
org.apache.jasper.compiler.JDTCompiler.generateClass(JDTCompiler.java:439)
```





# 에러 코드별 처리

---

- HTTP 에러 코드표 - HTTP 에러 코드에 따른 에러 상황을 나타낸 표 (상황에 따라 다른 코드를 표시하게 됨)
- 주로 발생하는 에러
  - 404 - 사용자가 잘못된 페이지를 요청할 때 발생
    - not found, 문서를 찾을 수 없음
  - 500 - 프로그램 코딩 오류일 때
    - internal server error, 서버 내부 오류



# 에러 코드별 에러 처리 예제

---

- 404 에러와 500 오류 처리하기
  - 에러를 코드별로 처리할 때 - web.xml에 기술
  - web.xml
    - <error-page></error-page>태그 안에 처리할 에러 코드와 페이지를 기술
  - 원래의 에러 메시지 대신 사용자에게 익숙한 문구가 표시되도록 web.xml에서 에러를 코드별로 처리하도록 작성



# web.xml

```
<?xml version="1.0" encoding="utf-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-
    app_2_4.xsd"
  version="2.4">
```

```
<error-page>
```

```
  <error-code>404</error-code>
```

```
  <location>/error/404code.jsp</location>
```

```
</error-page>
```

에러 코드명 기술

에러 코드를 처리할 페이지

**404** 에러가 발생하면 프로그램 제어가 **/error/404code.jsp**로 이동  
이때 요청 페이지는 사용자가 요청한 페이지를 그대로 유지함

```
<error-page>
```

```
  <error-code>500</error-code>
```

```
  <location>/error/500code.jsp</location>
```

```
</error-page>
```

```
</web-app>
```



# 404code.jsp

---

```
<%@ page contentType="text/html; charset=utf-8" %>
```

```
<%
```

```
    response.setStatus(HttpServletResponse.SC_OK);
```

```
%>
```

현재 페이지가 정상적으로 응답되는 페이지임을 지정하는 코드,  
이 코드가 생략되면 웹 브라우저는 자체적으로 제공하는 화면을 표시함

```
<html>
```

```
    <head><title>404에러 페이지</title></head>
```

```
    <body>
```

```
        요청하신 페이지는 존재하지 않습니다.
```

```
    </body>
```

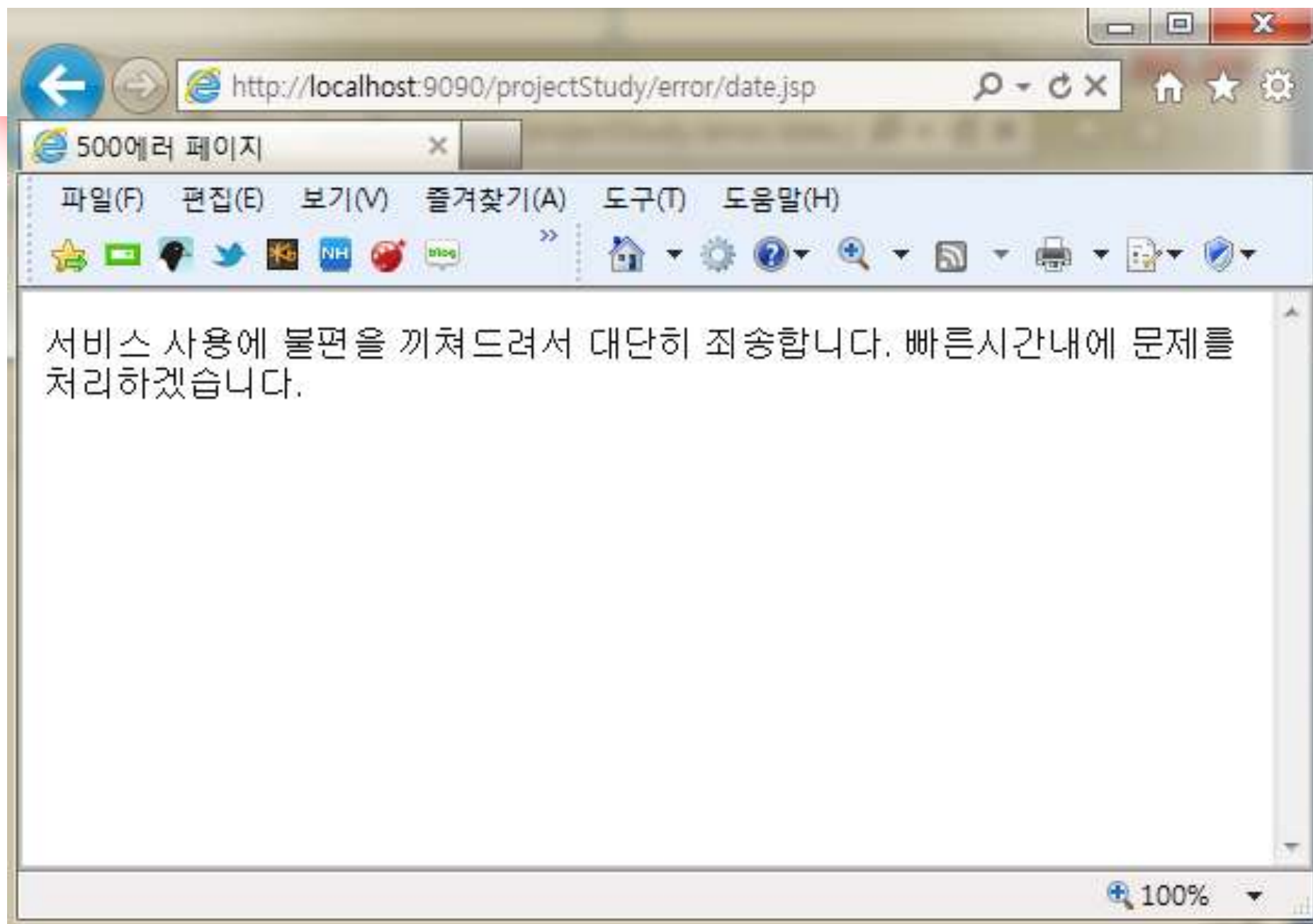
```
</html>
```

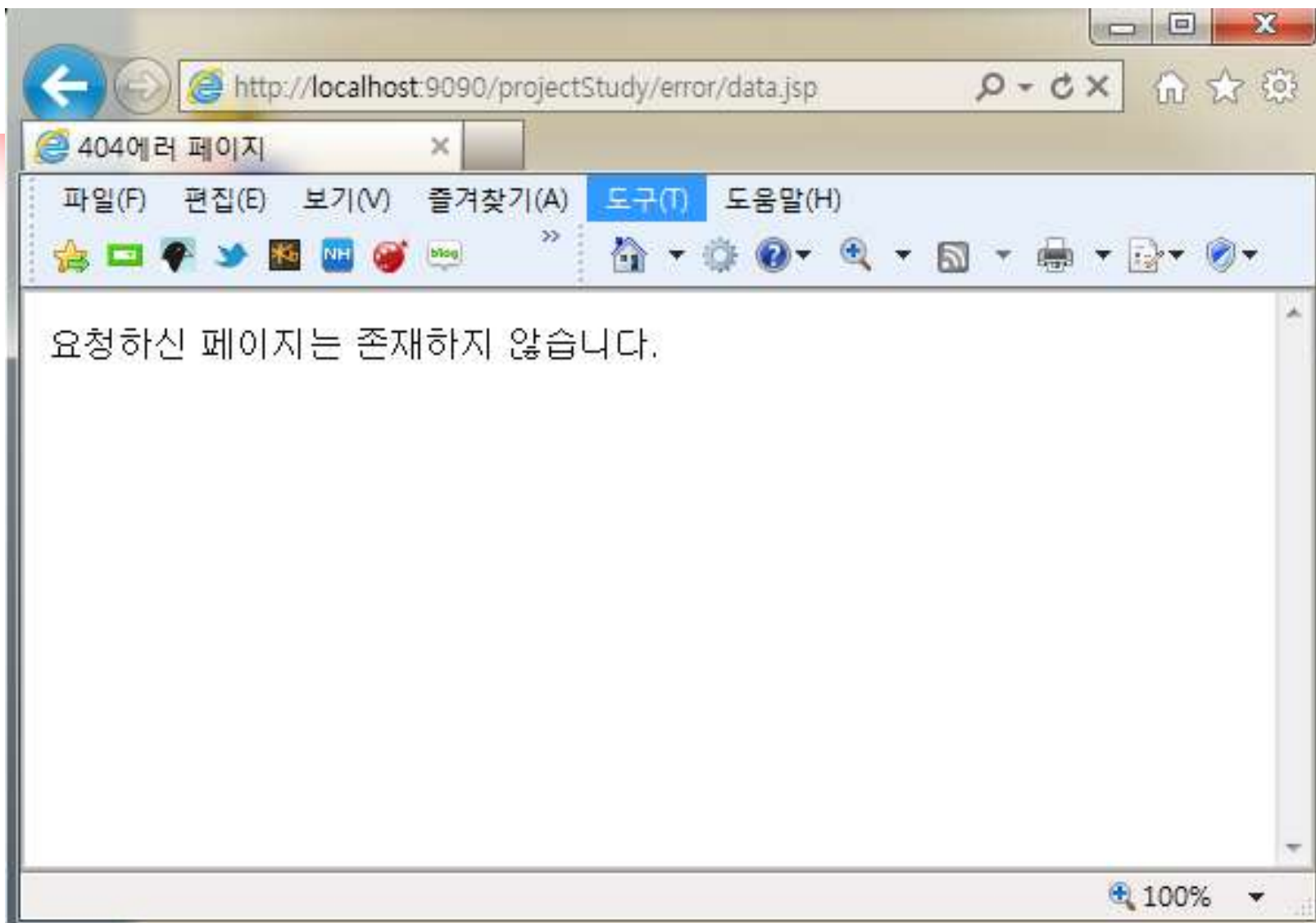


# 500code.jsp

---

```
<%@ page contentType="text/html; charset=utf-8" %>
<%
    response.setStatus(HttpServletResponse.SC_OK);
%>
<html>
    <head><title>500에러 페이지</title></head>
    <body>
        서비스 사용에 불편을 끼쳐드려서 대단히 죄송합니다.
        빠른시간내에 문제를 처리하겠습니다.
    </body>
</html>
```





<http://localhost:9090/projectStudy/error/data.jsp> => 이 페이지는 존재하지 않는 페이지로 404 에러 유발



# 메인 화면 공지사항

---



베스트

자유게시판

이용후기

Q&A

FAQ

이벤트&쿠폰

허브

허브차

아로마오일

허브향초



공지사항

MORE

- ▶ 공지사항
- ▶ 책 추천
- ▶ 문의합니다
- ▶ 공지합니다
- ▶ 테스트
- ▶ 질문있어요

## 상품 목록

### 신 상품



로즈마리  
8,000원



자스민  
9,000원



라벤다허브차  
7,000원



로즈마리허브차  
8,000원

# 메인 공지사항-mainNotice.jsp

```
<%
    //메인 페이지의 오른쪽 상단에 include 되는 페이지 - 공지사항
    BoardService boardService = new BoardService();
    List<BoardVo> alist = boardService.listMainNotice();
%>
<style type="text/css">
    .divNotice{
        width:310px
    }
    .divNotice div table{
        width:300px
    }
    .divNotice div span{
        padding:0 0 0 160px;
    }
    .divNotice div .img1{
        width:310px;height:6px
    }
</style>
<div class="divNotice">
    <div>
        
        <span><a href="<%=request.getContextPath()%>/notice/list.jsp">
```

```
</a></span>
```

```
</div>
```

```
<div>
```

```

```

```
</div>
```

```
<div> <!-- 공지사항 -->
```

```
<table summary="최근 공지사항 6건을 보여주는 표입니다.">
```

```
<caption>공지사항</caption>
```

```
<tbody>
```

```
<!-- 반복시작 -->
```

```
<%for(int i=0;i<alist.size();i++){
```

```
BoardBean vo = alist.get(i);
```

```
%>
```

```
<tr>
```

```
<td>
```

```

```

```
<a href="<%=request.getContextPath()%>/notice/detail.jsp?no=<%=vo.getNo()%>">
```

```
<%=vo.getTitle() %>
```

```
</a>
```

```
</td>
```

```
</tr>
```

```
<%}//for %>
```

```
<!-- 반복 끝 -->
```

```
</tbody>
```

```
</table>
```

```
</div>
```

```
</div>
```



## 6건의 데이터만 조회하기

- 의사 컬럼 : 모조, 유령 컬럼 - 테이블에 실제로 존재하지는 않지만, 일반적인 컬럼처럼 행동하는 컬럼
  - [1] rownum - row들에 대한 순서값
    - 쿼리의 결과로 나오는 각각의 row들에 대한 순서 값을 가리키는 의사컬럼
    - 주로 특정 개수나 그 이하의 row를 선택할 때 사용됨
  - [2] rowid - row들이 저장된 주소값(id)

```
select rownum, rowid as "row id", ename, sal
from emp;
```

```
select rownum, ename, sal
from emp
where rownum<=5;
```

• 5건의 정보만 조회하기

```
select rownum, ename, sal
from emp
order by ename desc;
```



## 6건의 데이터만 조회하기

---

- inline view : from절에 sub query가 오는 것

```
select rownum, ename, sal
from (select ename, sal from emp order by ename desc)
where rownum<=5;
```

- 실습) student 테이블에서 grade순서로 정렬한 결과에서 상위 10개만 조회하기



# BoardDAO

Board 테이블에서 상위 6건의 데이터만 조회

```
public class BoardDAO {  
    private ConnectionPoolMgr pool;  
    public BoardDAO() {  
        pool= new ConnectionPoolMgr();  
    }//생성자
```

```
    public List<BoardVO> listMainNotice() throws SQLException{  
        Connection con=null;  
        PreparedStatement ps=null;  
        ResultSet rs=null;  
        List<BoardVO> alist = new ArrayList<BoardVO>();  
        try{  
            con=pool.getConnection();
```

```
            String sql="select * from (Select * From board order by no desc) where rownum<=6";  
            ps=con.prepareStatement(sql);  
            rs=ps.executeQuery();
```



# BoardDAO

---

```
        while(rs.next()){
            BoardVO vo = new BoardVO();

            vo.setNo(rs.getInt("no"));
            vo.setName(rs.getString("name"));
            vo.setPwd(rs.getString("pwd"));
            vo.setTitle(rs.getString("title"));
            vo.setContent(rs.getString("content"));
            vo.setRegdate(rs.getString("regdate"));
            vo.setEmail(rs.getString("email"));
            vo.setReadCount(rs.getInt("readcount"));

            alist.add(vo);
        } //while
    } finally{
        pool.dbClose(con, ps, rs);
    }
    return alist;
}
```



# index.jsp

---

```
<%@ page language="java" contentType="text/html; charset=utf-8"    pageEncoding="utf-8"%>
<%@include file="inc/shopTop.jsp"%>

<!-- content page -->
<article id="centerCon">
    <br/><br/>
</article>
<article id="rightCon">
    <!-- 공지사항 -->
    <%@include file="inc/mainNotice.jsp" %>
</article>
<article id="listCon">
    <!-- 이벤트별 상품 목록 -->
    <%@include file="shop/product/productCatalog.jsp" %>
</article>

<%@include file="inc/shopBottom.jsp"%>
```





배포

---



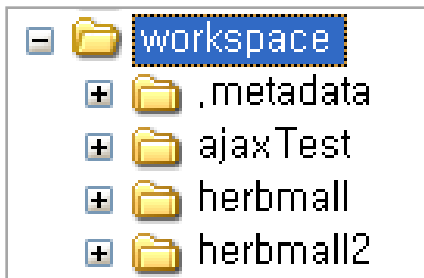
## WAR 파일을 이용한 웹 어플리케이션의 배포

- 웹 사이트 구축시 개발된 모든 코드를 하나로 묶어서 설치하는 것이 효율적
- 서블릿과 jsp 기술에서는 웹 어플리케이션 디렉토리 전체를 WAR 파일이라는 아카이브 형태의 파일로 만드는 방법을 제공함
- [1] WAR 파일 만드는 방법
  - WAR 파일(Web ARchive file)
    - JDK에 포함되어 있는 jar.exe 프로그램을 이용해서 웹 어플리케이션 디렉토리의 구조 전체를 하나로 묶은 일종의 아카이브 파일임
  - 아카이브할 웹 애플리케이션 디렉토리로 가서 jar.exe 프로그램을 실행
    - 명령 프롬프트 창에서 실행해야 함

```
C:\ java\ jdk1.8.0_65\ bin\ jar.exe
```

# WAR 파일을 이용한 웹 어플리케이션의 배포

- 예) herbmall2 디렉토리를 herbmall.war 라는 이름의 WAR 파일로 만들기



```
D:\>cd D:\java\workspace
```

```
D:\java\workspace>jar cvf herbmall.war -C herbmall2/ .
```

c: 새로운 아카이브 파일을 생성하라는 뜻

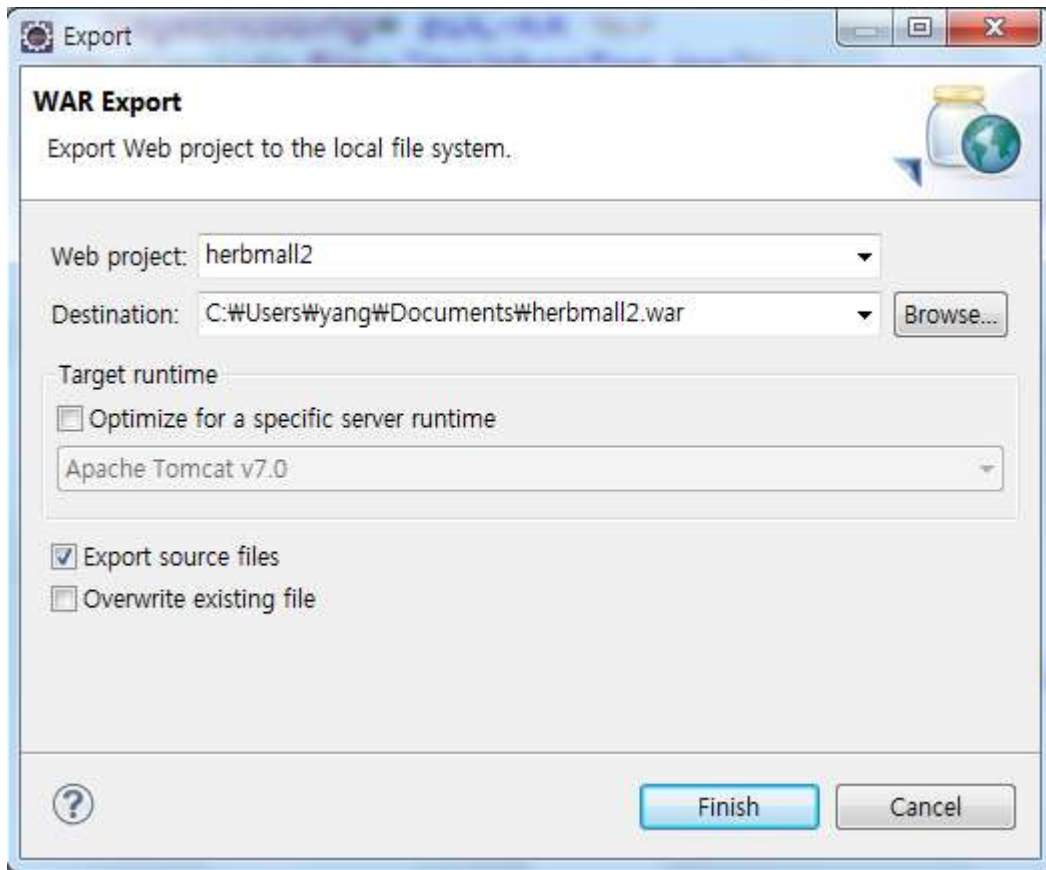
v: 아카이브에 대한 상세한 정보를 모니터로 출력하라는 뜻

f: 아카이브 파일의 이름을 지정하겠다는 뜻

herbmall2 디렉토리로 가서 현재 디렉토리(.)에 있는 모든 파일을 아카이브(압축)하라는 뜻

# WAR 파일을 이용한 웹 어플리케이션의 배포

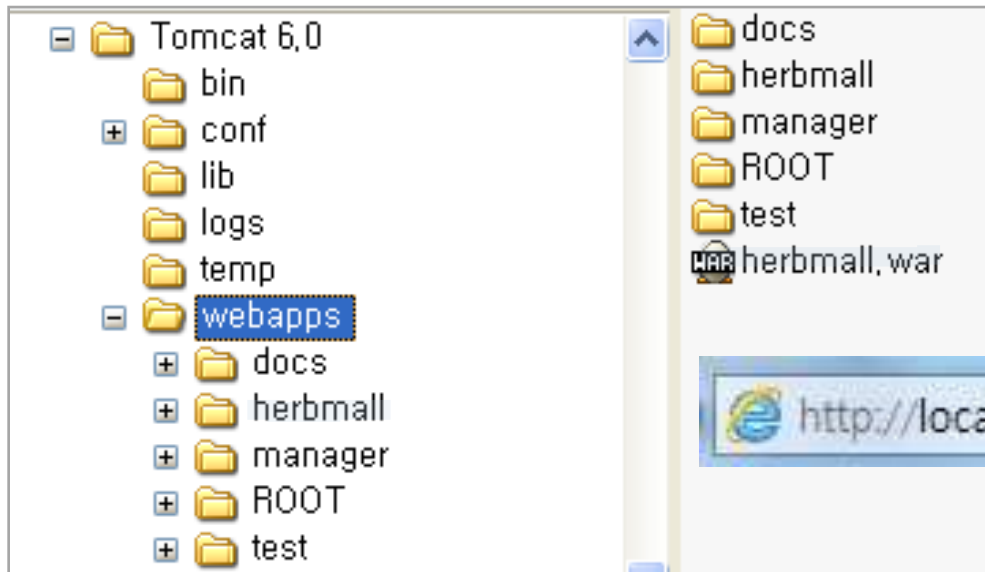
- [2] 이클립스에서 WAR 파일 만들기
  - 웹 프로젝트에서 오른쪽 마우스 - Export -> WAR file



tomcat monitor 찾으됨

# 톰캣에 WAR 파일을 설치하는 방법

- 웹 컨테이너의 종류마다 WAR 파일을 설치하는 방법은 다르다
- 톰캣 - 웹 어플리케이션 디렉토리들이 있는 webapps 디렉토리에 WAR 파일을 복사해 놓기만 하면 됨
  - WAR 파일을 설치하고 나면 톰캣은 자동으로 그 파일의 아카이브를 풀어서 새로운 웹 어플리케이션 디렉토리를 만듦
  - 그 디렉토리는 WAR 파일의 이름에서 확장자를 뺀 나머지 부분과 같은 이름을 가짐



아님 기존 workspace의 톰캣 sever.xml 의 context 를 복사해서 외부 톰캣 server.xml의 context 부분에 붙여야 됨

그다음에 docBase 부분을 실행하려는 war의 압축 해제 절대경로와 똑같이 만든다  
ex) C:\Tomcat 9.0\webapps\herbmall

