

2

개발 환경 설정



이 장에서 다룰 내용

1

J2SE(JDK7.0) 설치와 환경변수 설정

2

웹 컨테이너(Apache Tomcat 7.0) 설치

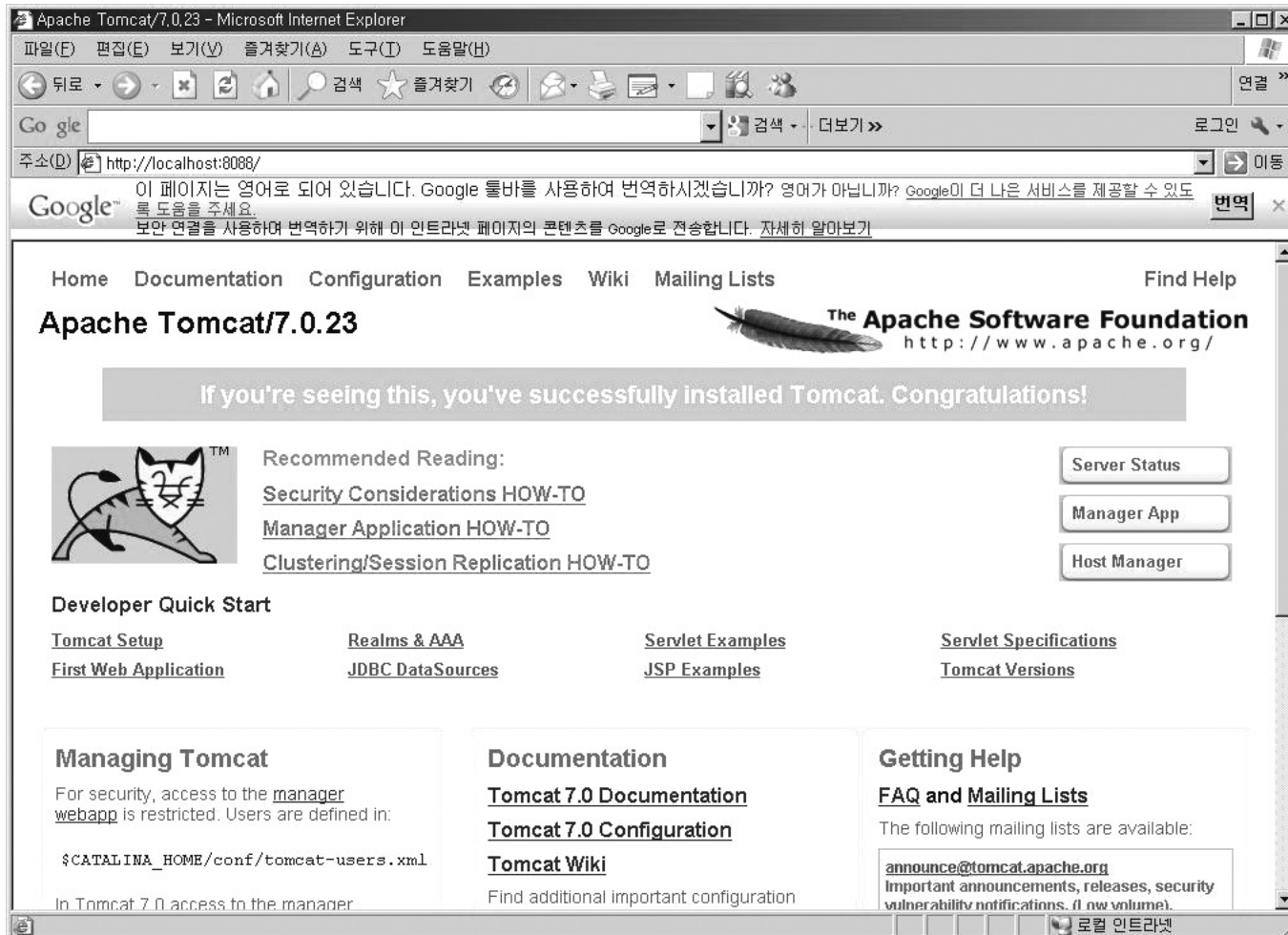
3

개발툴(이클립스) 설치



J2SE(JDK7.0) 설치와 환경변수 설정

- 웹 브라우저를 실행하고 <http://localhost:8088> 주소로 접속



웹 컨테이너(Apache Tomcat 7.0) 설치

□ servlet-api.jar 파일 복사하기

- 탐색기를 열어 아파치가 설치된 경로 아래의 lib 폴더(C:\wjsp2.2\tomcat\lib)로 들어간다. lib 폴더에 있는 파일들 중 servlet-api.jar 파일을 C:\wjsp2.2\Java\jdk1.7.0_02\jre\lib\ext 경로에 복사한다.



J2SE(JDK7.0) 설치와 환경변수 설정

□ 톰캣에서 ServletTest 서블릿과 JSP 파일 동작 확인

- 웹 프로젝트를 서비스하기 위한 기본 디렉토리
 - 톰캣설치경로\webapps
- 브라우저의 접근 경로
 - “http://localhost:8088/폴더명”



웹 컨테이너(Apache Tomcat 7.0) 설치

□ 웹 애플리케이션으로 동작하게 하기 위한 파일 만들기

- **webapps** 폴더 밑에 다음 그림과 같이 **ex1**이라는 폴더를 생성한다.
- **ex1** 폴더 안에 **WEB-INF** 폴더를 생성한다.
- **WEB-INF** 폴더 안에 **classes**, **src**, **lib** 폴더를 생성한다.
 - ▶ **classes** : 컴파일된 클래스 파일들이 존재하는 디렉토리로, 웹 애플리케이션에서는 이 경로를 클래스들의 루트 경로로 인식함
 - ▶ **src** : 자바 소스 파일들이 존재하는 경로. 실제 애플리케이션을 배포할 때는 필수적인 디렉토리는 아니지만 관리의 편의상 만들어 둠
 - ▶ **lib** : 웹 애플리케이션에서 사용하는 라이브러리 파일들이 존재하는 디렉토리. 만약, 웹에서 DB 연동 작업을 한다면 **Ojdbc14.jar** 파일이 이 디렉토리에 존재해야 함



웹 컨테이너(Apache Tomcat 7.0) 설치

■ ServletTest.java 파일 생성(코드 51p참조)

ServletTest.java



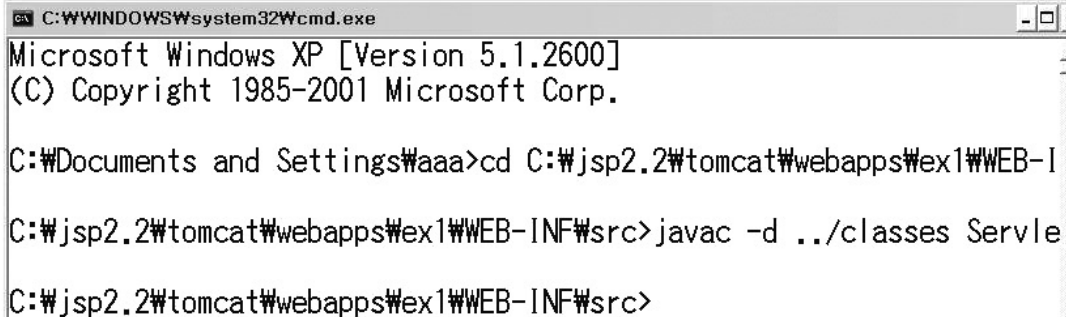
jsp2.2\Chapter2\ex1\ServletTest.java

```
1  import javax.servlet.*;
2  import javax.servlet.http.*;
3  import java.io.*;
4  import java.util.Calendar;
5  public class ServletTest extends HttpServlet{
6      public void doGet(HttpServletRequest request, HttpServletResponse response)
7      throws IOException{
8          response.setContentType("text/html");
9          response.setCharacterEncoding("euc-kr");
10         Calendar c=Calendar.getInstance();
11         int hour=c.get(Calendar.HOUR_OF_DAY);
12         int minute=c.get(Calendar.MINUTE);
13         int second=c.get(Calendar.SECOND);
14         PrintWriter out=response.getWriter();
15         out.write("<HTML><HEAD><TITLE>ServletTest</TITLE></HEAD>");
16         out.write("<BODY><H1>");
17         out.write("현재시각은 ");
18         out.write(Integer.toString(hour));
19         out.write("시 ");
20         out.write(Integer.toString(minute));
21         out.write("분 ");
22         out.write(Integer.toString(second));
23         out.write("초입니다.");
24         out.write("</H1></BODY></HTML>");
25         out.close();
26     }
27 }
```



웹 컨테이너(Apache Tomcat 7.0) 설치

■ 컴파일



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\aaa>cd C:\jsp2.2\tomcat\webapps\ex1\WEB-INF\src
C:\jsp2.2\tomcat\webapps\ex1\WEB-INF\src>javac -d ../classes Servle
C:\jsp2.2\tomcat\webapps\ex1\WEB-INF\src>
```



웹 컨테이너(Apache Tomcat 7.0) 설치

- web.xml 파일을 톰캣 설치 디렉토리(C:\wjsp2.2\tomcat\conf)에서 C:\wjsp2.2\tomcat\wenapps\ex1\WEB-INF 디렉토리에 복사
- web.xml 파일을 수정

참고

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
version="3.0">
```

```
<servlet>
<servlet-name>ServletTest</servlet-name>
<servlet-class>ServletTest</servlet-class>
</servlet>
```

```
<servlet-mapping>
<servlet-name>ServletTest</servlet-name>
<url-pattern>/test</url-pattern>
</servlet-mapping>
```

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

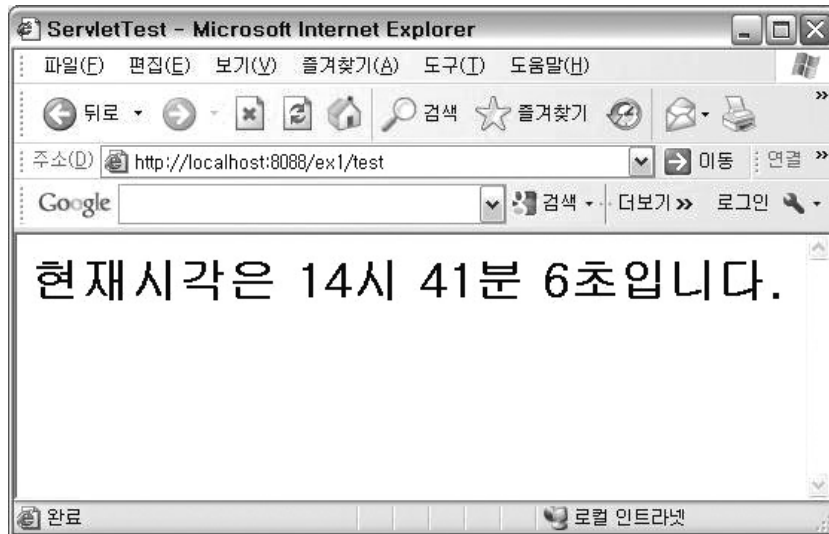
추가하는 부분



웹 컨테이너(Apache Tomcat 7.0) 설치

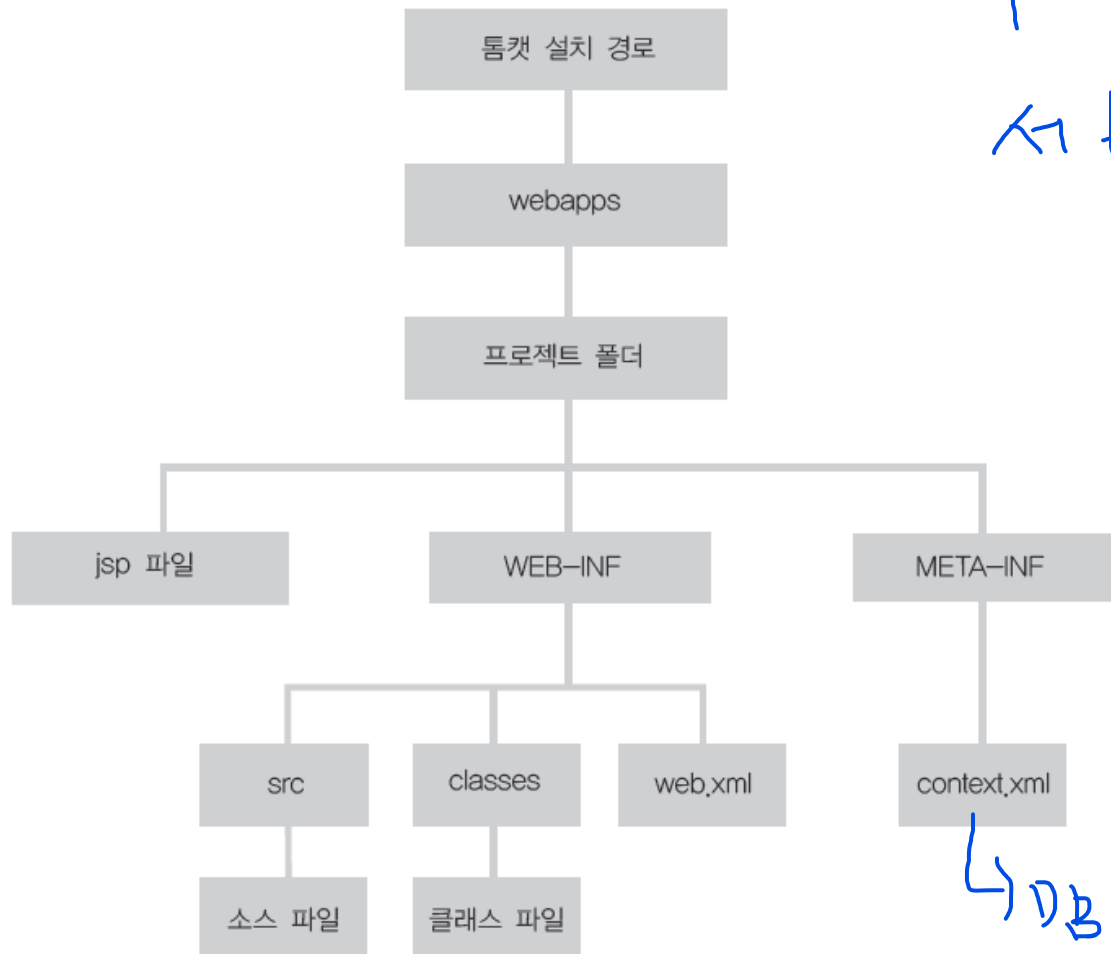
□서블릿 구동하기

- web.xml 파일을 수정하였으므로 톰캣을 재가동
- 브라우저를 실행하고 주소창에 <http://localhost:8088/ex1/test>를 입력하고 실행



웹 컨테이너(Apache Tomcat 7.0) 설치

□ 웹 애플리케이션 폴더 구조



JSP 코딩

서비스한다면

↳ 디플로이

[그림2-1] 웹 애플리케이션 폴더 구조



웹 컨테이너(Apache Tomcat 7.0) 설치

□서블릿 클래스를 JSP로 만들기

ServletTest.jsp



jsp2.2\Chapter2\ServletTest.jsp

```
1  <%@ page language="java" contentType="text/html; charset=EUC-KR"
2      pageEncoding="EUC-KR"%>
3  <%@page import="java.util.Calendar"%>
4  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/
5  html4/loose.dtd">
6  <html>
7  <head>
8  <%
9  Calendar c=Calendar.getInstance();
10 int hour=c.get(Calendar.HOUR_OF_DAY);
11 int minute=c.get(Calendar.MINUTE);
12 int second=c.get(Calendar.SECOND);
13 %>
14 <meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
15 <title>Servlet Test</title>
16 </head>
17 <body>
18 <h1>현재시간은 <%=hour %>시 <%=minute %>분 <%=second %>초 입니다.</h1>
19 </body>
20 </html>
```

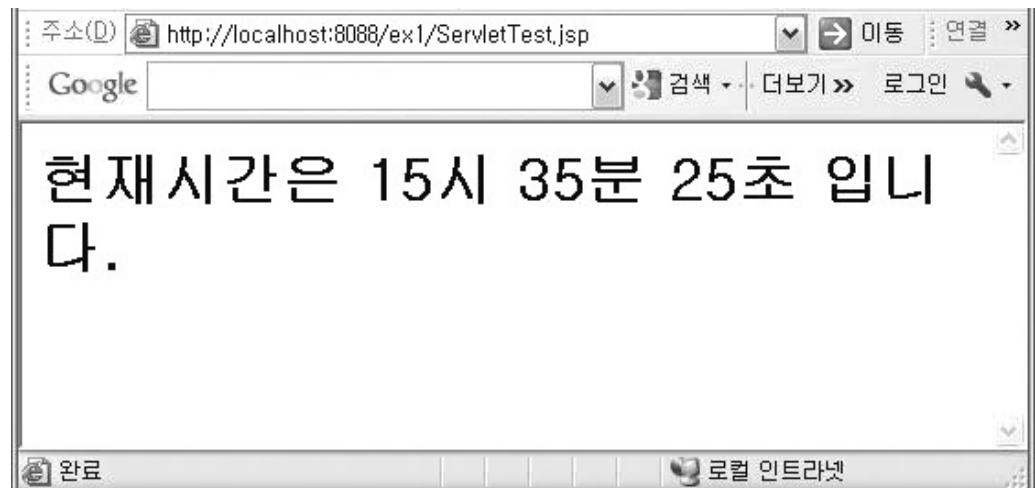


웹 컨테이너(Apache Tomcat 7.0) 설치

■ ServletTest.jsp 저장



■ 브라우저의 주소 창에 “http://localhost/ex1/ServletTest.jsp” 를 입력하고 실행



개발툴(이클립스) 설치

□ 다운로드 및 설치

- 이클립스 홈페이지(www.eclipse.org)에 접속후 파일 다운로드
- 다운로드된 이클립스 압축 파일을 풀고 eclipse.exe 파일을 실행
- 작업이 진행될 디렉토리를 C:\jsp2.2\workspace로 지정

□ 이클립스에 톰캣을 서버로 등록하기

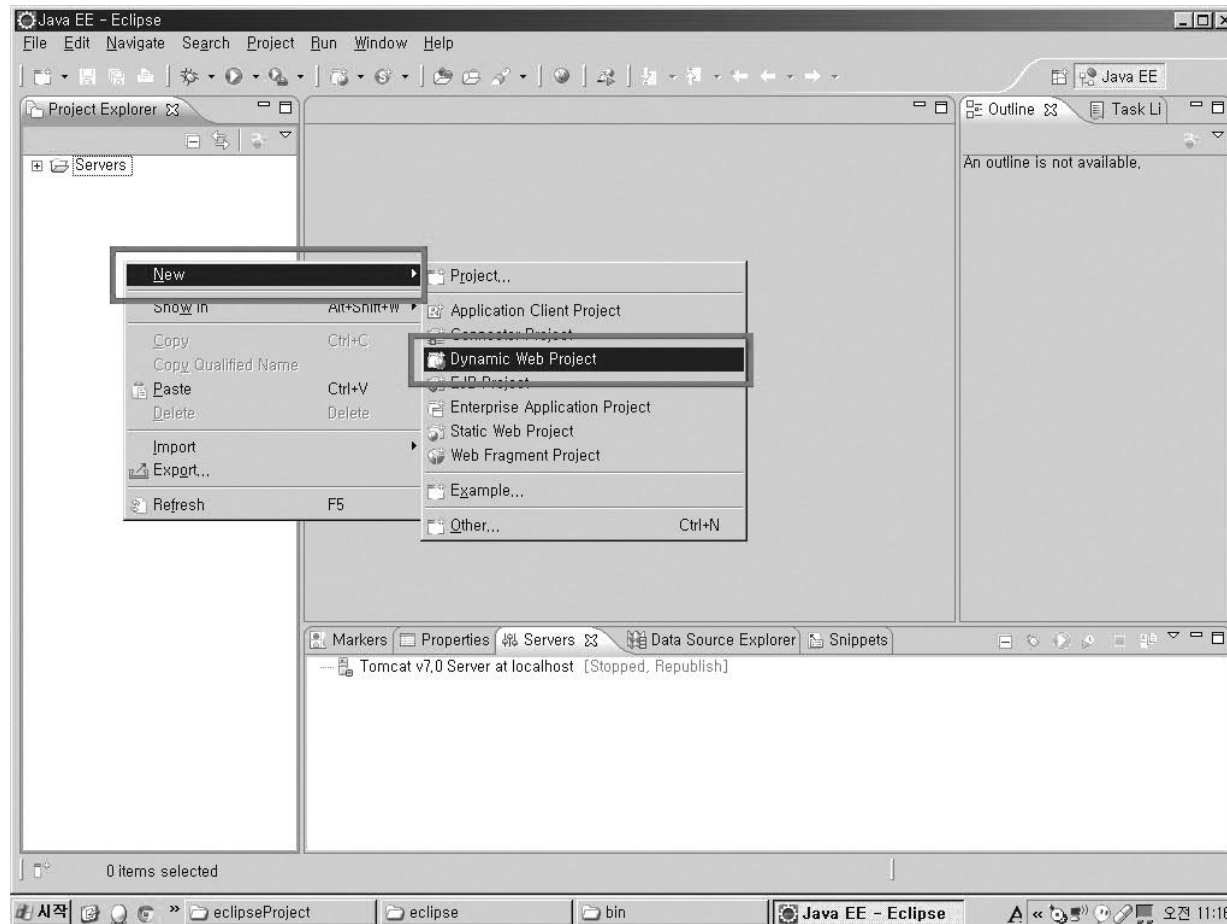
- 서블릿이나 JSP는 웹 컨테이너에서 실행되기 때문에 이클립스에 톰캣을 등록하는 과정이 필요하다.



개발툴(이클립스) 설치

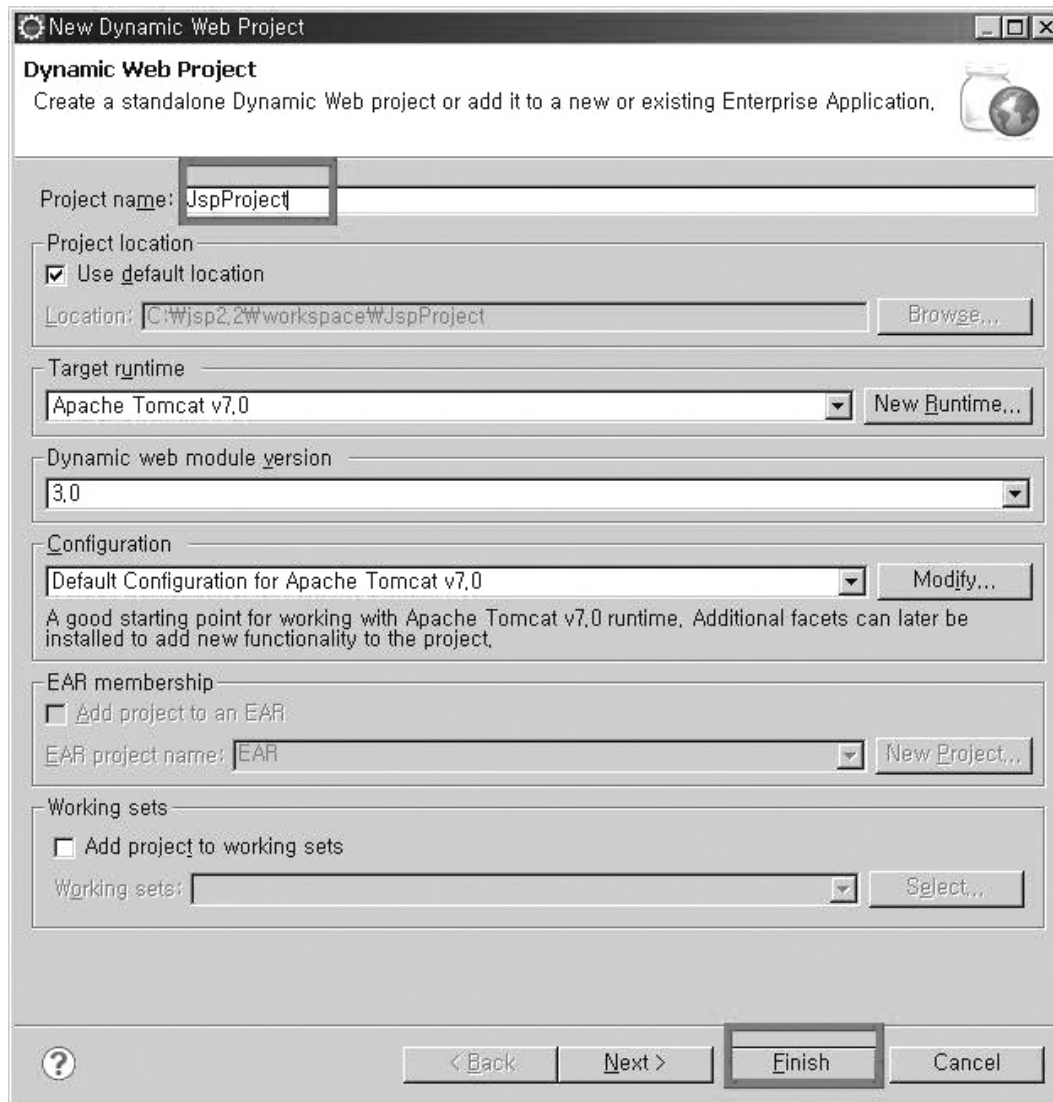
□ 웹 프로젝트 생성하고 애플리케이션 실행하기

- Project Explorer 뷰에서 마우스 우측 버튼을 클릭한 후 [New]-[Dynamic Web Project]를 실행한다.



개발툴(이클립스) 설치

- Project name을 JspProject로 지정하고 [Finish] 버튼을 클릭한다.

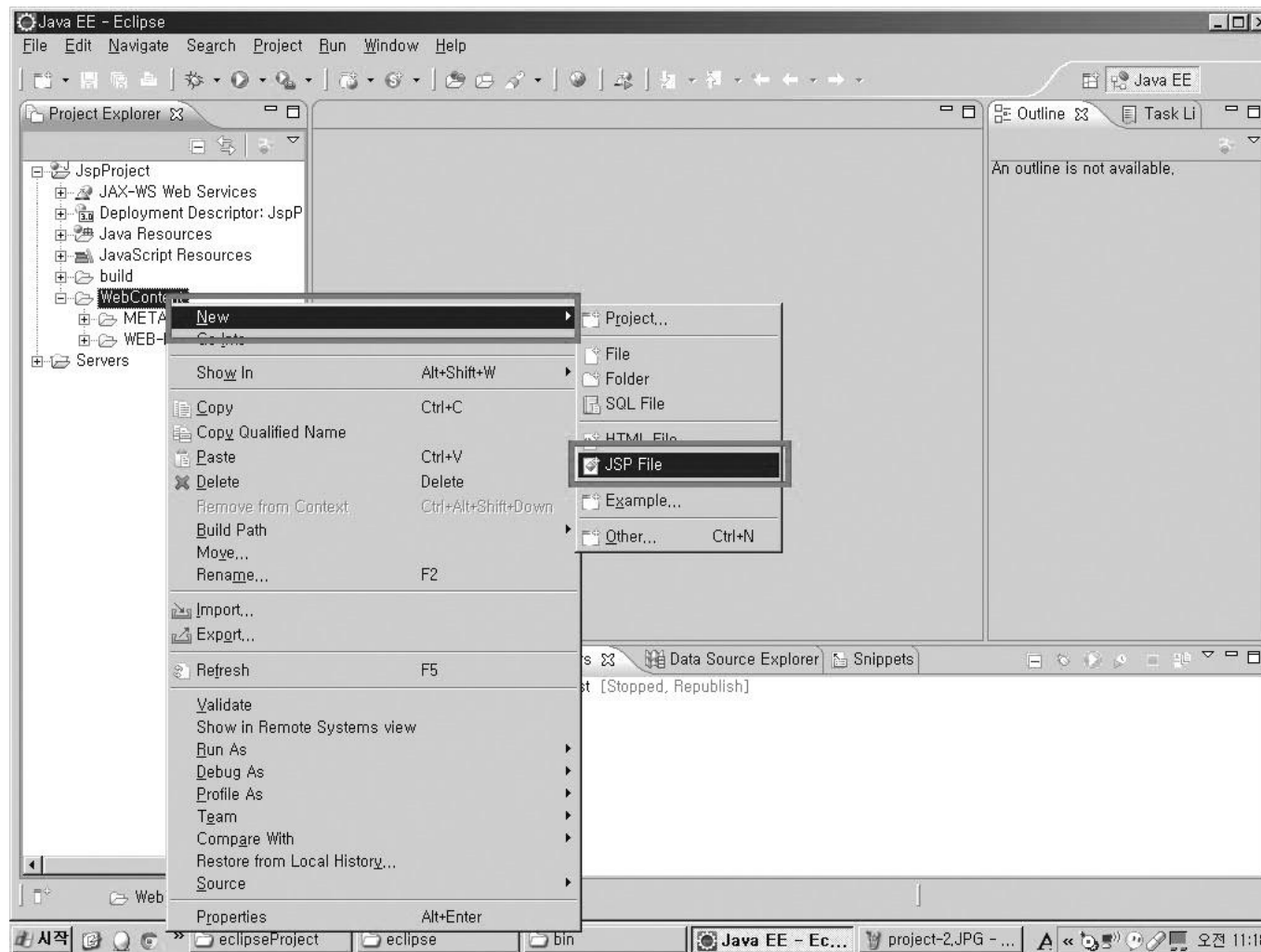


The screenshot shows the 'New Dynamic Web Project' dialog box in Eclipse. The 'Project name' field is set to 'JspProject'. The 'Project location' section has 'Use default location' checked, with the location path 'C:\wsp2\workspace\JspProject' displayed. The 'Target runtime' is set to 'Apache Tomcat v7.0'. The 'Dynamic web module version' is set to '3.0'. The 'Configuration' is set to 'Default Configuration for Apache Tomcat v7.0'. The 'EAR membership' section has 'Add project to an EAR' unchecked, and the 'EAR project name' is set to 'EAR'. The 'Working sets' section has 'Add project to working sets' unchecked. The 'Finish' button is highlighted with a red box.



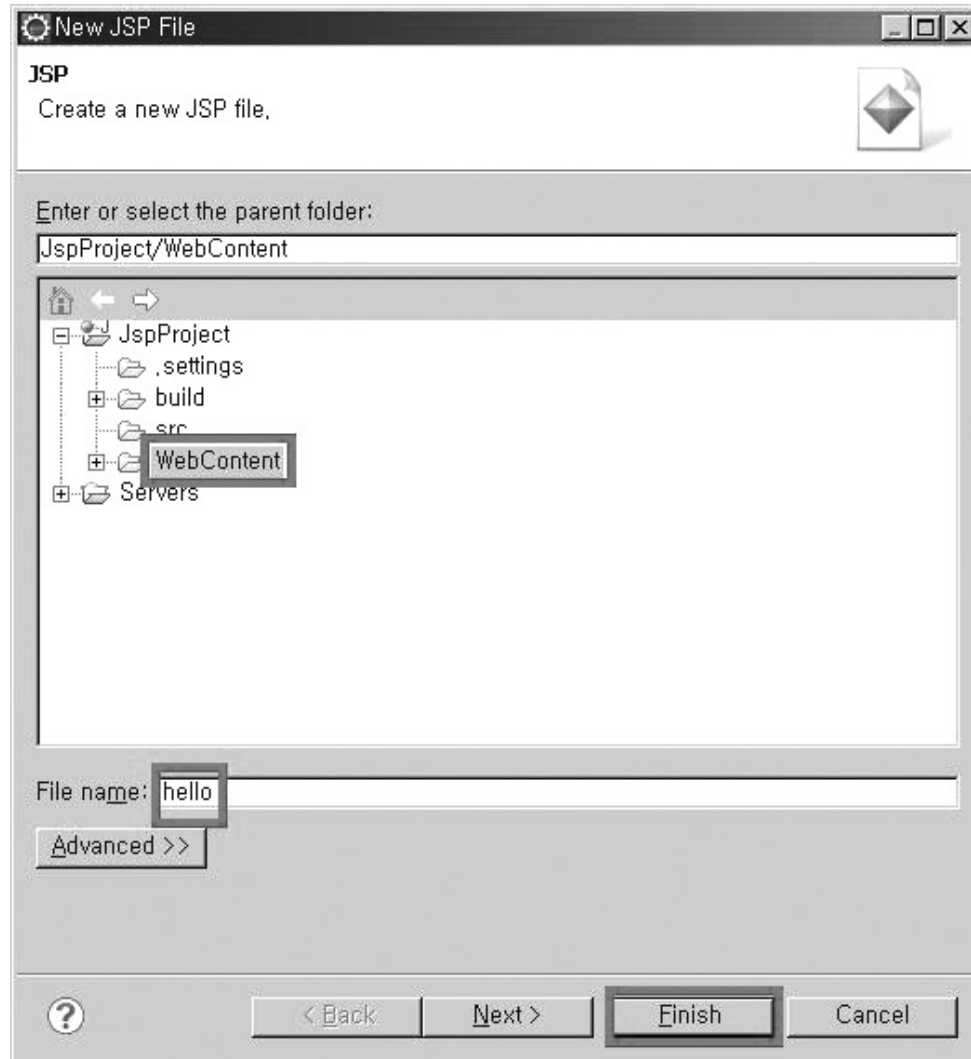
개발툴(이클립스) 설치

- 해당 프로젝트의 [WebContent]–[New]–[JSP File] 메뉴를 실행한다.



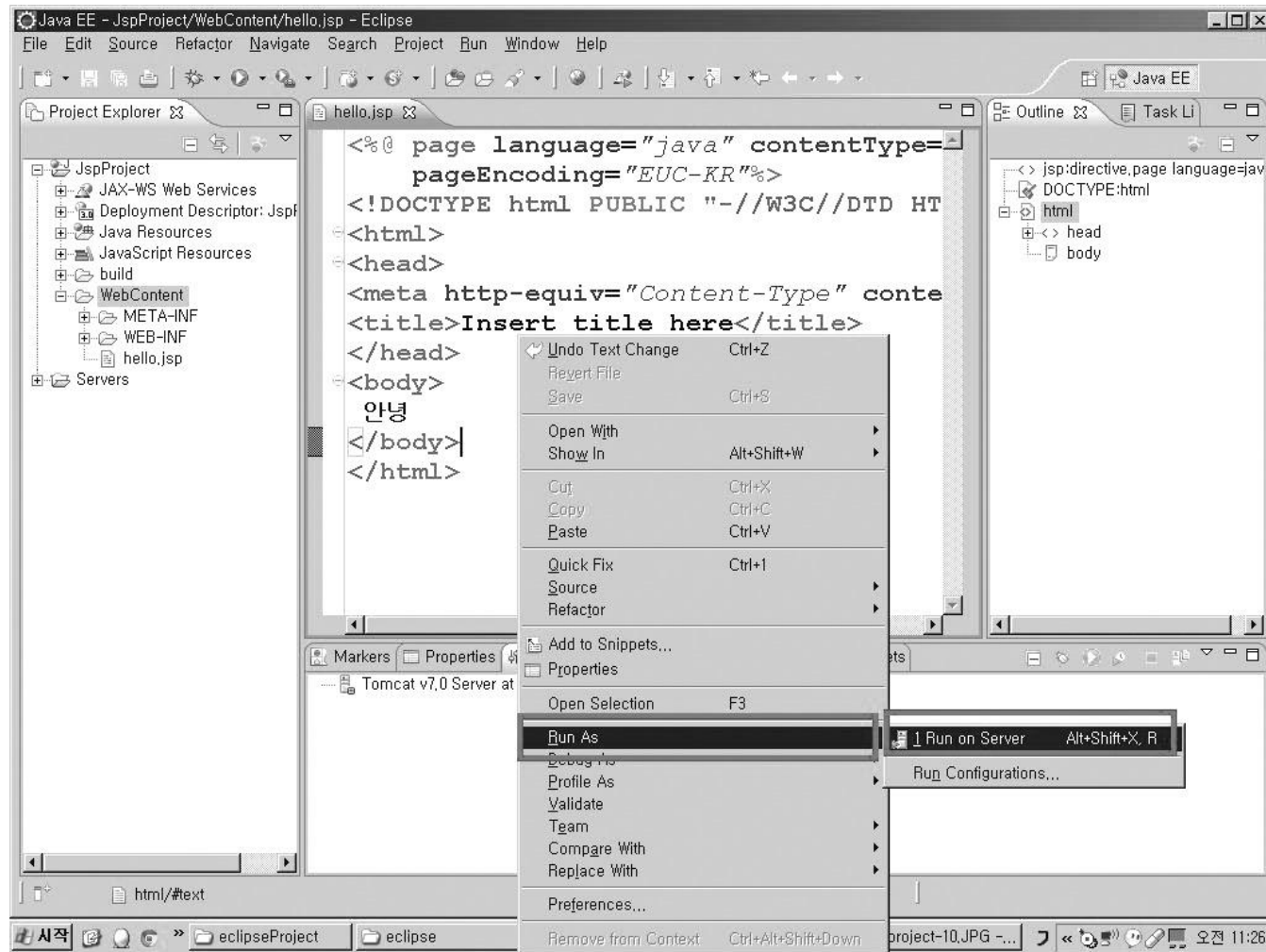
개발툴(이클립스) 설치

- **File Name**을 **hello**로 지정한다. 확장자는 생략이 가능하다.



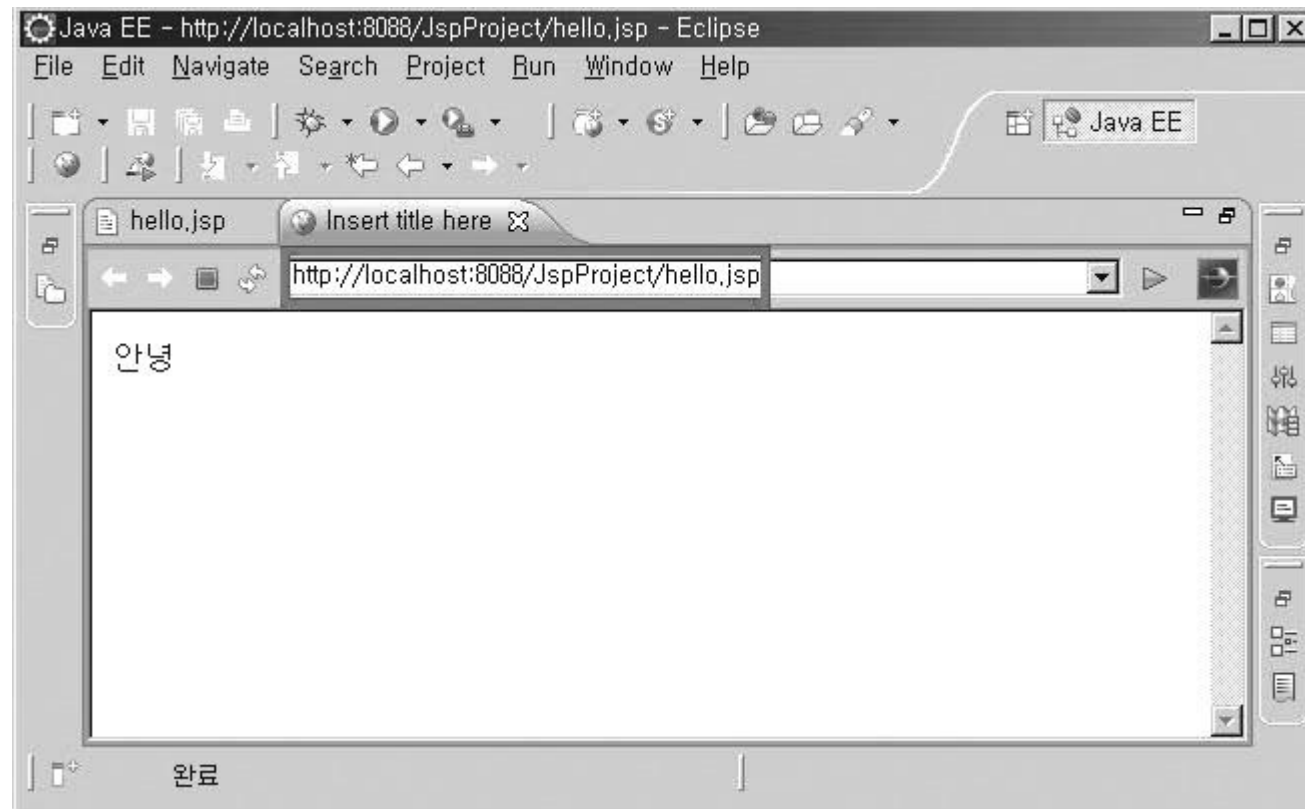
개발툴(이클립스) 설치

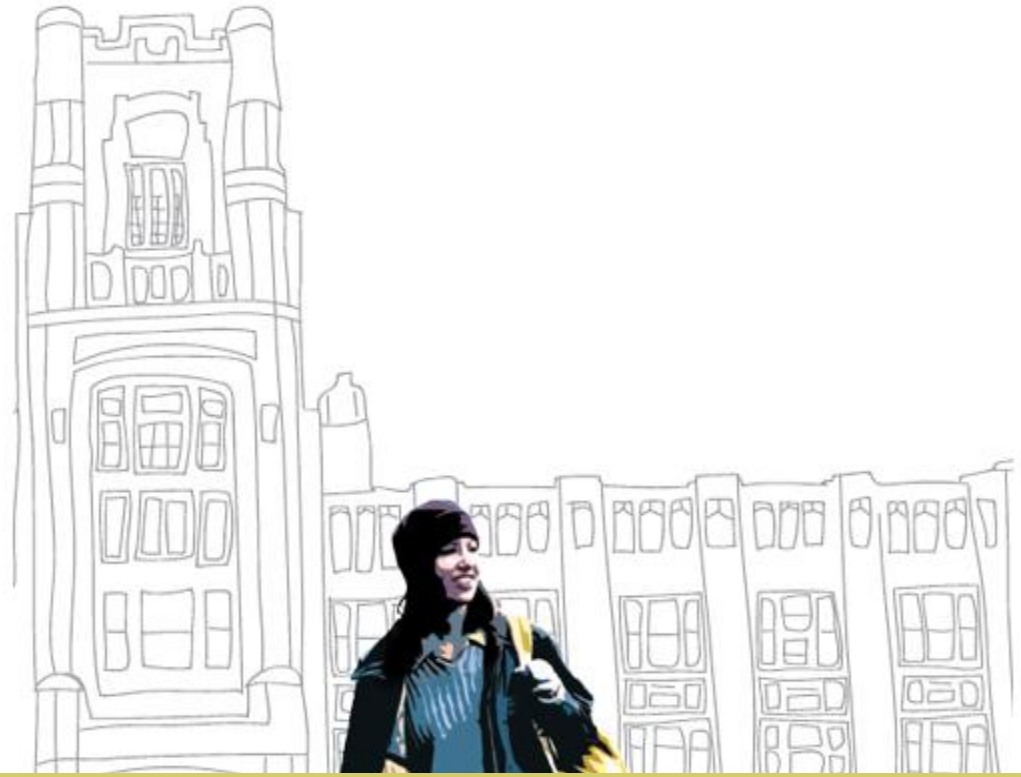
- 해당 jsp를 선택한 후 [Run As]-[Run on Server]를 선택하여 실행한다



개발툴(이클립스) 설치

- 실행 결과 화면은 다음과 같다.





Thank You

