

아래의 예시는 Eclipse3.4.x(가니메데)를 기준으로 작성된 것이다. Eclipse3.5.x(갈릴레오)의 설치도 같은 방법으로 한다.

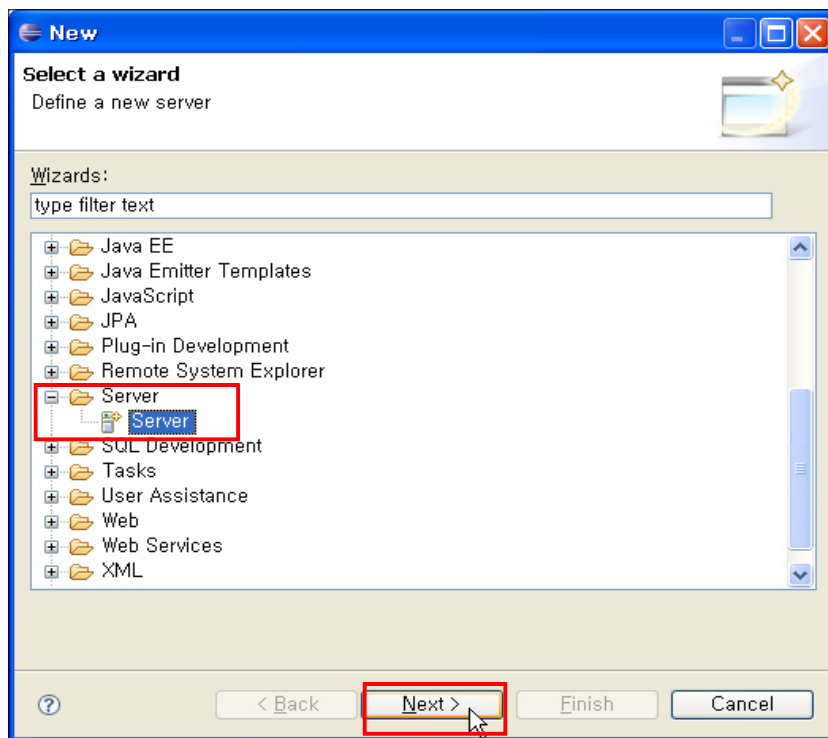
1. 서버(Server)설정

이클립스에서는 프로젝트를 작성해서 프로그래밍을 한다. JSP도 예외가 아니기 때문에 프로젝트를 작성해야하는데, 웹에서 동작하는 웹 프로젝트의 경우에는 먼저 이 프로젝트가 동작될 서버를 먼저 설정해야한다. 서버를 설정한 후 서버에 동적 웹프로젝트를 등록해야 비로서 제대로 웹페이지를 실행할 수 있는 환경이 갖춰지게 된다.

자, 그럼 서버를 설정해 보자.

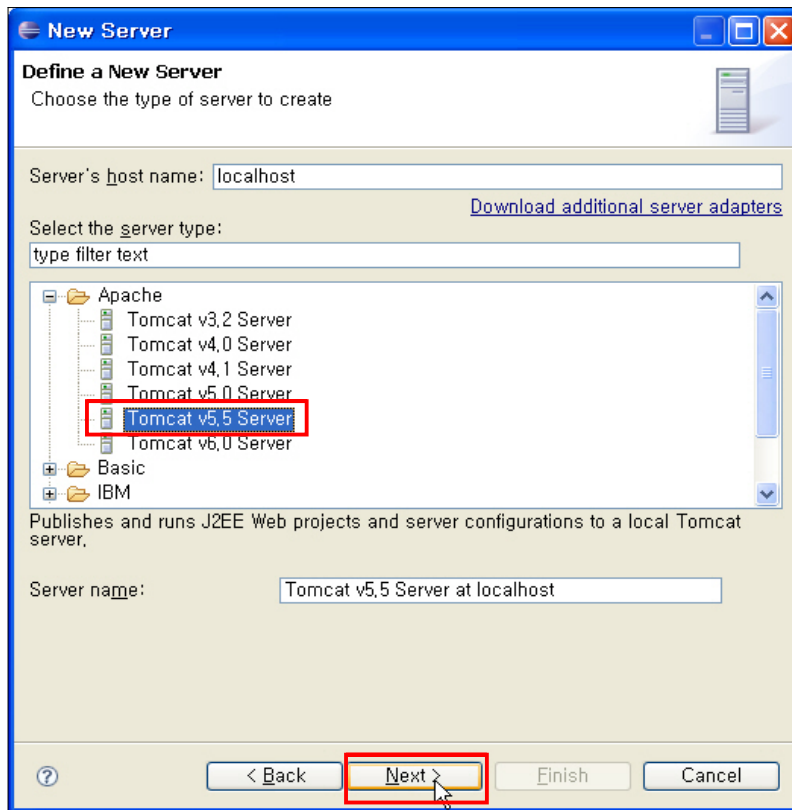
① 서버 설정을 하기 위해서 [File]-[New]-[Other]메뉴를 선택한다.

② [New]창에서 [Select] 표시되면 [Server]항목의 하위항목인 [Server]항목을 선택한 후 [Next]단추를 클릭한다.



<서버(Server)설정 1>

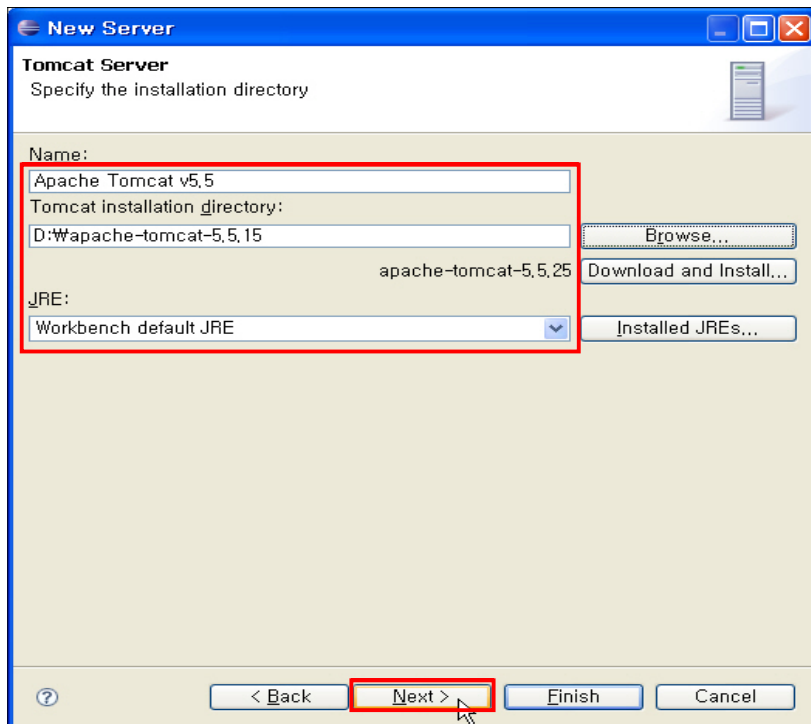
③ [New Server]창이 [Define a New Server]로 진행하면, [Apache]-[Tomcat v5.5 Server]항목을 선택한 후 [Next]단추를 클릭한다. 톰캣6.0.x버전의 사용시는 [Tomcat v6.0 Server]항목을 선택한다.



<서버(Server)설정 2>

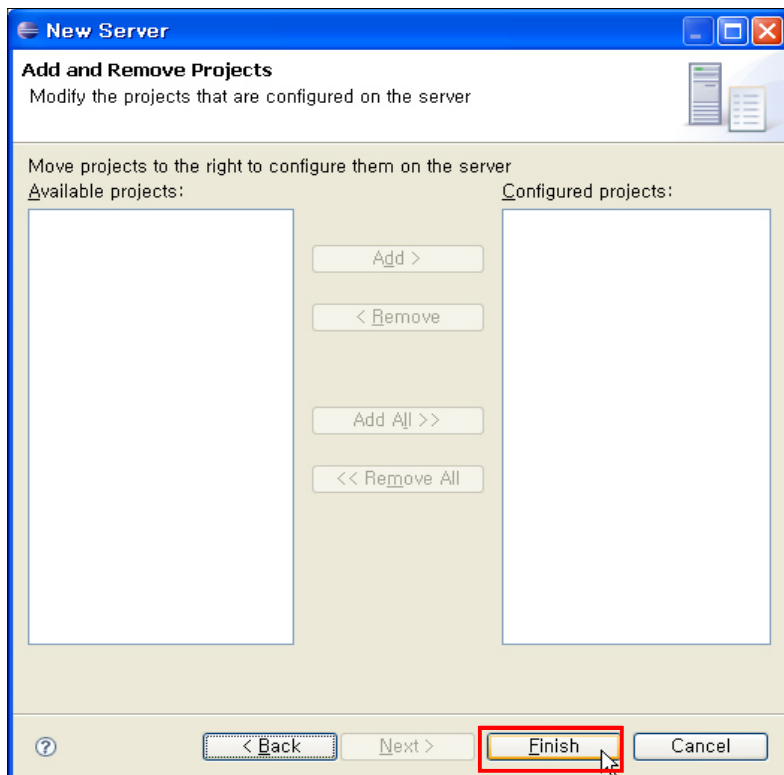
④ [Tomcat Server]화면으로 진행하면 다음과 같이 설정 후 [Next]단추 클릭한다.

Name	설정 이름을 입력하는데 기본적으로 서버명과 버전이 그대로 지정된다. 기본값을 그대로 사용한다.
Tomcat installation directory	사용하는 Tomcat Server가 설치되어 있는 폴더의 경로를 지정한다. 오른쪽의 [Browse]단추를 클릭해서 선택한다. 필자의 경우 Tomcat이 설치된 경로가 C:\Wapache-tomcat-5.5.15 이다. 각자 자신의 경로를 기술한다.
JRE	설치된 JRE설정에서 어떤 JRE를 사용해서 서버를 실행할지를 지정한다. 기본값을 그대로 사용한다.



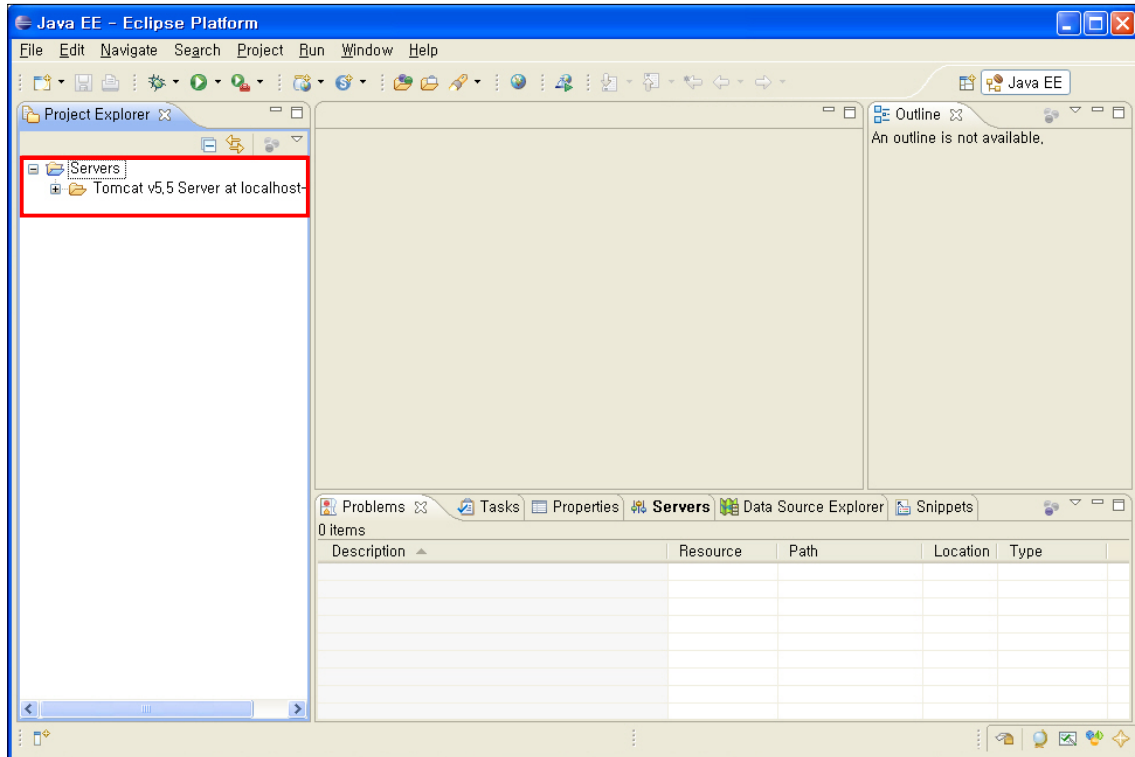
<서버(Server)설정 3>

⑤ [Add and Remove Projects]화면에서는 아직 프로젝트를 작성한 것이 없으므로 그냥 [Finish]단추를 클릭해서 완성한다.



<서버(Server)설정 4>

⑥ 서버의 설정이 완료되면 [Project Explorer]뷰에서 [Servers]항목이 생성되고, 이 항목의 하위항목에는 [Tomcat v5.5 Server~]항목이 표시된다.



<서버(Server)설정 5>

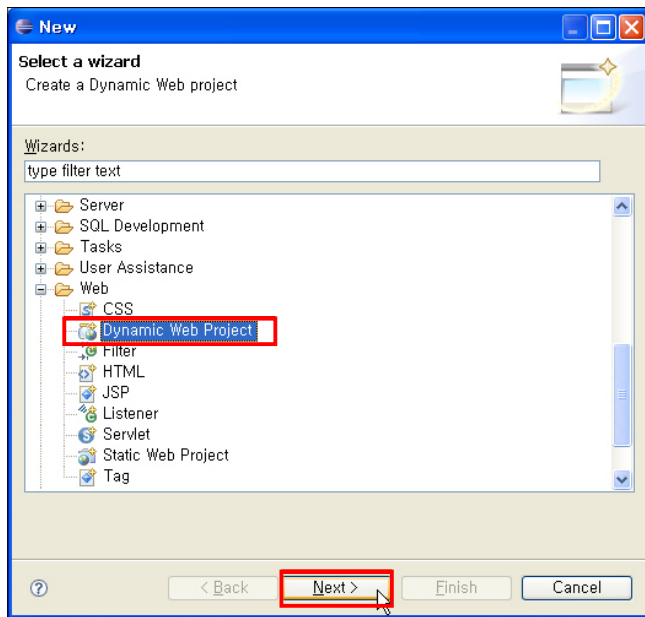
이로써 서버의 설정이 완료되었다. 이번에는 동적 웹 프로젝트를 작성해보자.

2. 동적 웹 프로젝트(Dynamic Web Project) 설정

이클립스에서 모든 프로그래밍은 프로젝트 단위로 이루어진다. 웹 프로그래밍의 경우에는 동적 웹 프로젝트를 작성해서 이 프로젝트안에 JSP페이지, 서블릿 및 각종 모듈을 위치시켜서 실행한다.

① 이클립스 창에서 [File]-[New]-[Other]메뉴를 선택한다.

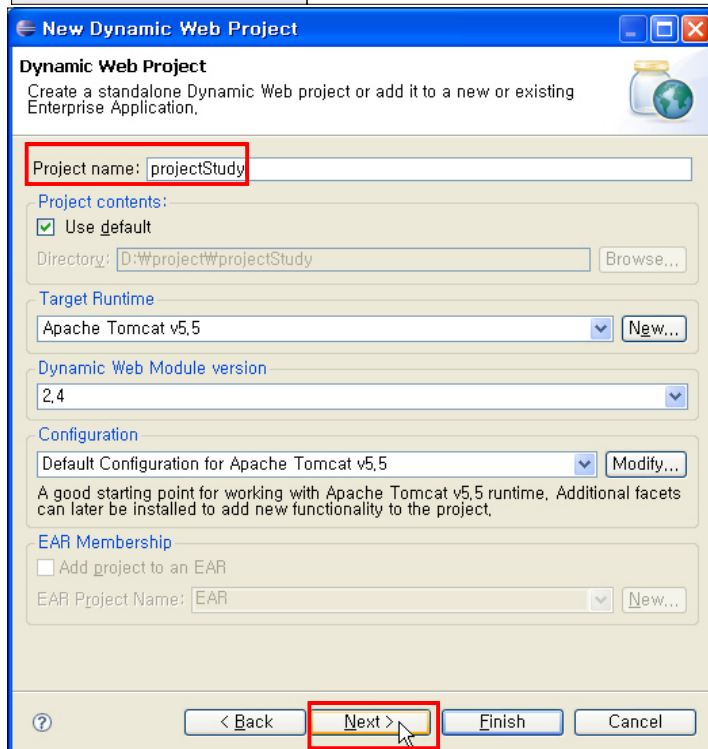
② [New]창에서 [Web]항목의 하위항목인 [Dynamic Web Project] 메뉴를 선택한다.



<동적 웹 프로젝트(Dynamic Web Project) 설정 1>

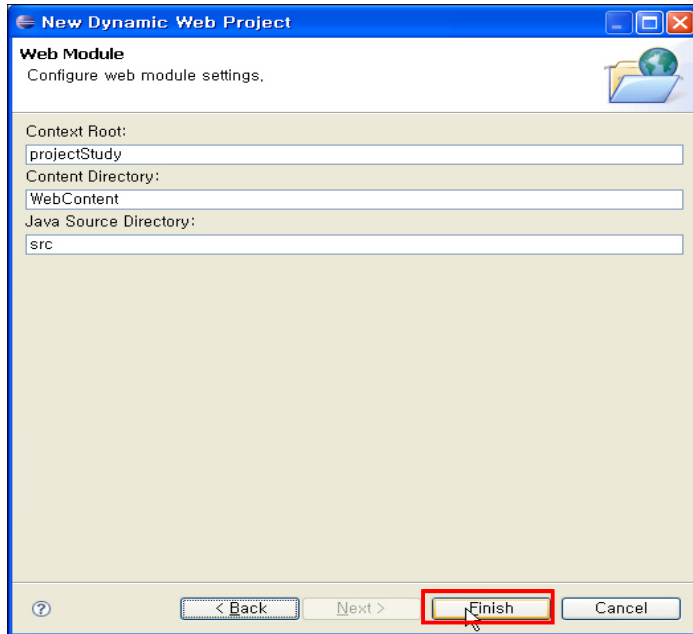
② [Dynamic Web Project]

Project name	프로젝트명을 'projectStudy'로 입력한다.
Target Runtime	기본값을 그대로 사용한다. 다른 서버를 사용하고 있는 경우는 그 서버를 선택한다.
Configurations	기본값을 그대로 사용한다.



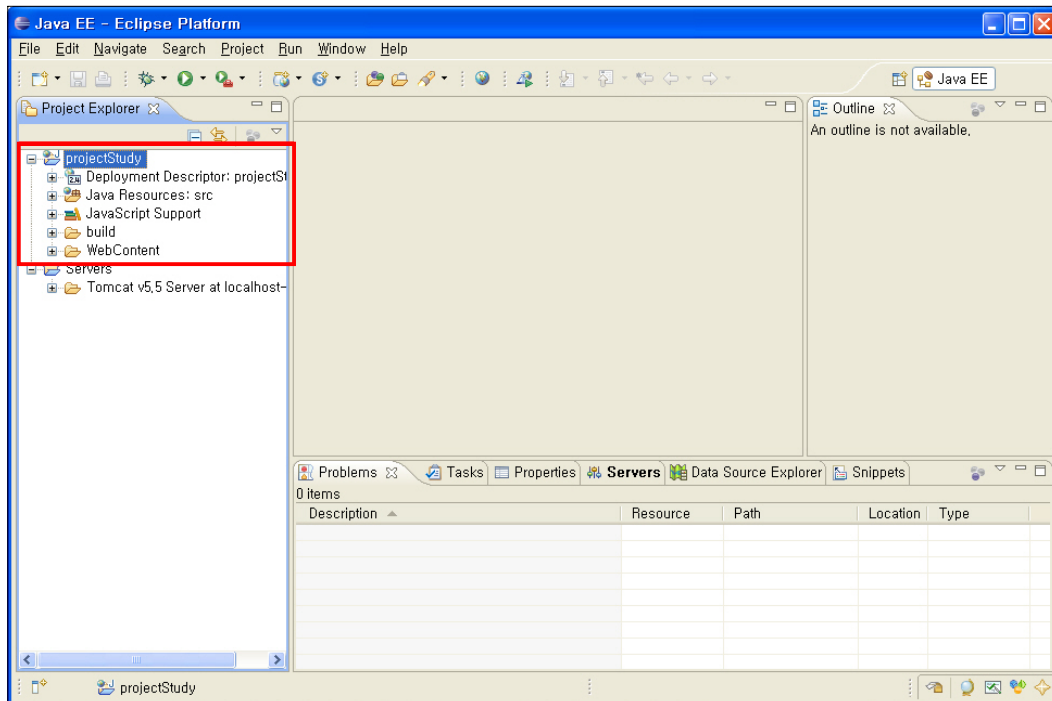
<동적 웹 프로젝트(Dynamic Web Project) 설정 2>

③ [Web Module]화면이 표시되면 기본값 그대로 사용하고 [Finish]단추를 클릭한다.



<동적 웹 프로젝트(Dynamic Web Project) 설정 3>

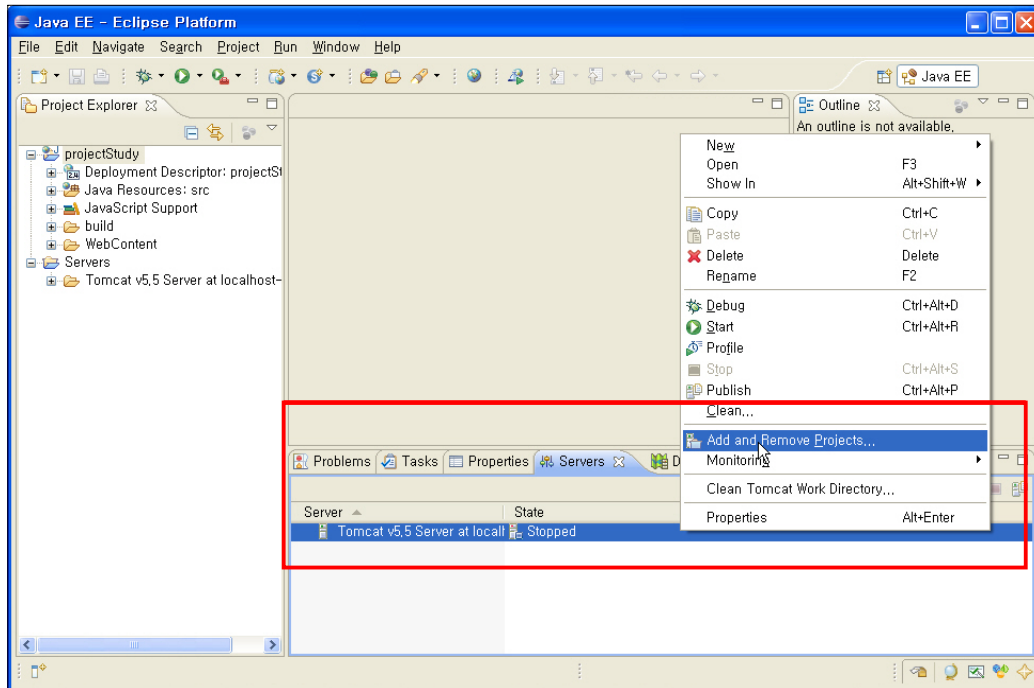
④ 다음과 같이 [projectStudy]가 추가된 것을 확인할 수 있다.



<동적 웹 프로젝트(Dynamic Web Project) 설정 4>

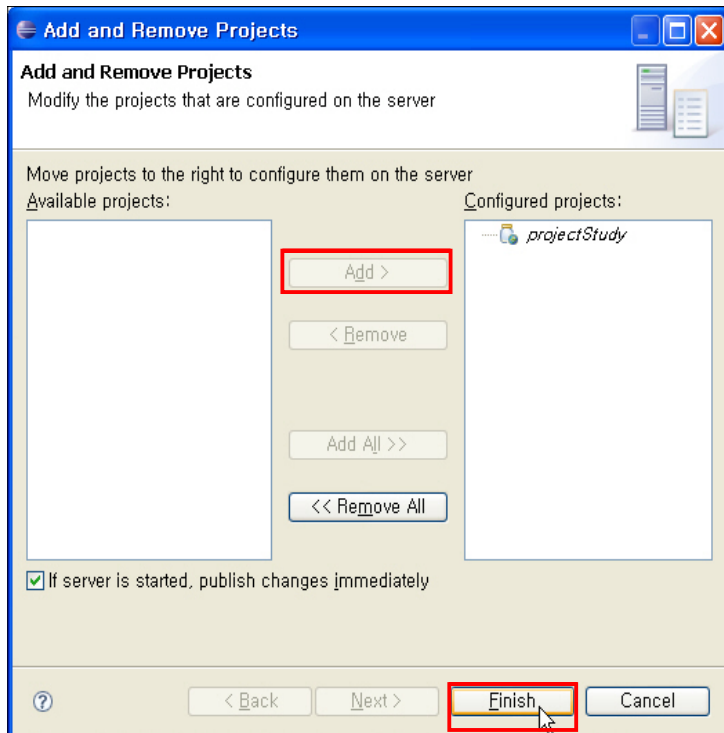
⑤ [projectStudy]프로젝트를 서버에 추가하기 위해서 [Server]뷰를 선택 후 해당서버를 선택

택해서 [Add and Remove Projects..]를 선택한다.



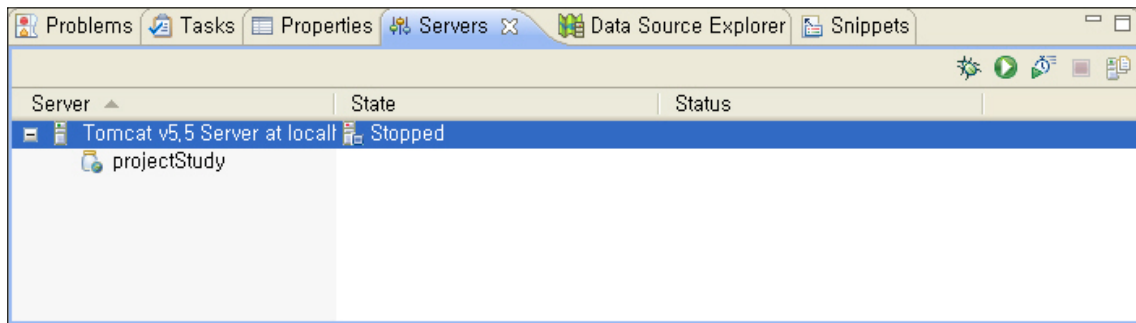
<동적 웹 프로젝트(Dynamic Web Project) 설정 5>

⑥ [Add and Remove Projects]창이 표시되면 [Available projects:]항목에서 'projectStudy'를 선택한 후 [Add]단추를 클릭한다. 그러면 해당 프로젝트가 [Configured projects]항목에 추가 된다. 프로젝트가 추가되면 [Finish]단추를 클릭한다.



<동적 웹 프로젝트(Dynamic Web Project) 설정 6>

⑦ [Server]뷰의 해당서버 안에 [projectStudy]프로젝트가 추가된 것을 확인할 수 있다.



<동적 웹 프로젝트(Dynamic Web Project) 설정 7>

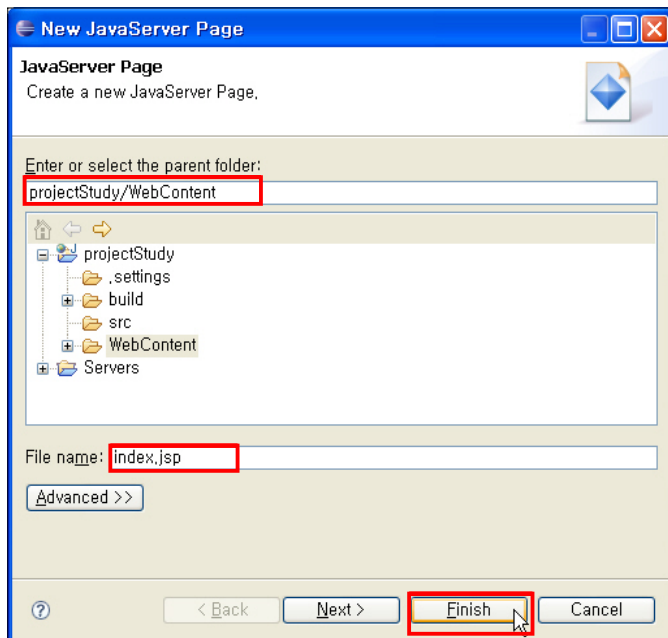
동적 웹 프로젝트가 작성되었으니 이제, JSP웹페이지와 서블릿을 작성해보자.

3. JSP 페이지 작성

이클립스에서 JSP페이지를 작성하기 위해서는 다음과 같은 순서로 한다.

① [projectStudy]프로젝트의 [WebContent]폴더를 선택한 후 [File]-[New]-[Other] 메뉴를 선택후, [Web]-[JSP]항목을 선택한다.

② [Enter or select the parent folder:]항목의 내용이 'projectStudy/WebContent'이면 [File name:]항목에 'index.jsp'를 입력한 후 [Finish]단추를 클릭한다.



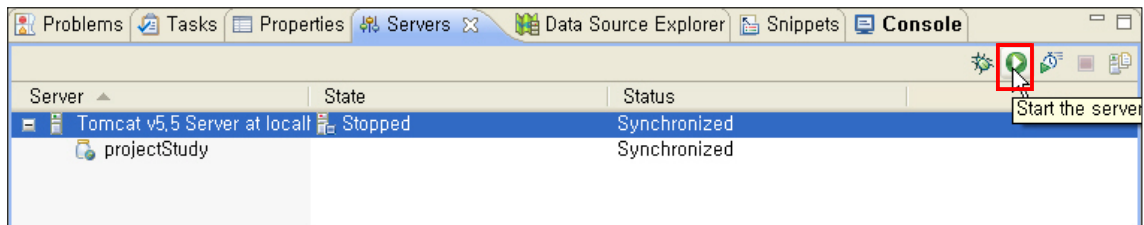
<이클립스에서 JSP 페이지 작성1>

③ index.jsp가 [편집기]뷰에 표시되면 다음과 같이 수정 후 저장한다.

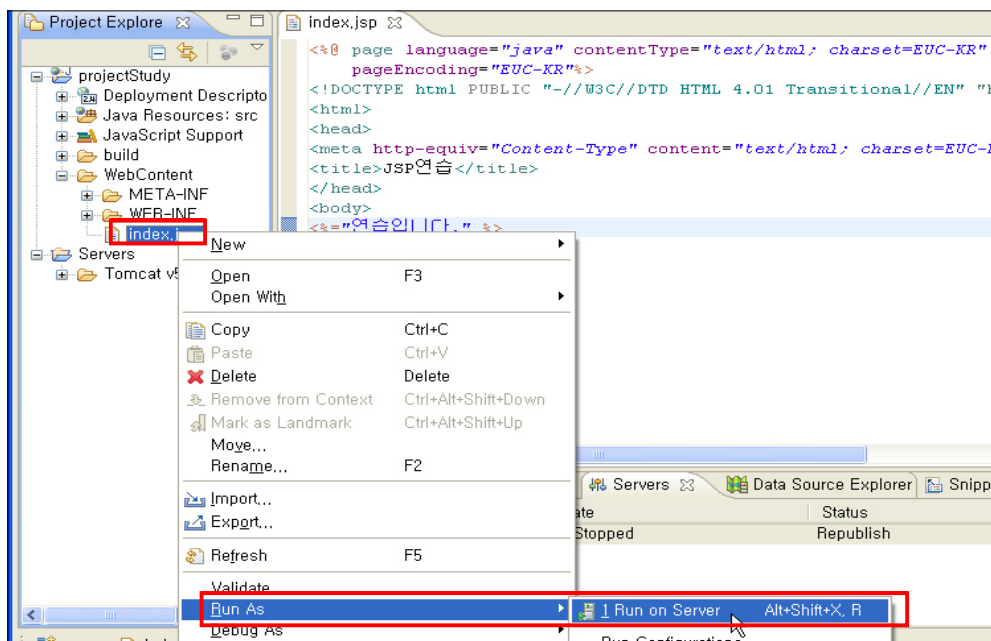


<그림3판용29.jpg><이클립스에서 JSP 페이지 작성2>

④ [Servers]뷰에서 [Start the server]단추를 클릭해서 서버를 올린다. (참고: 서버를 올리지 않아도 실행시 자동으로 서버가 올라간다)

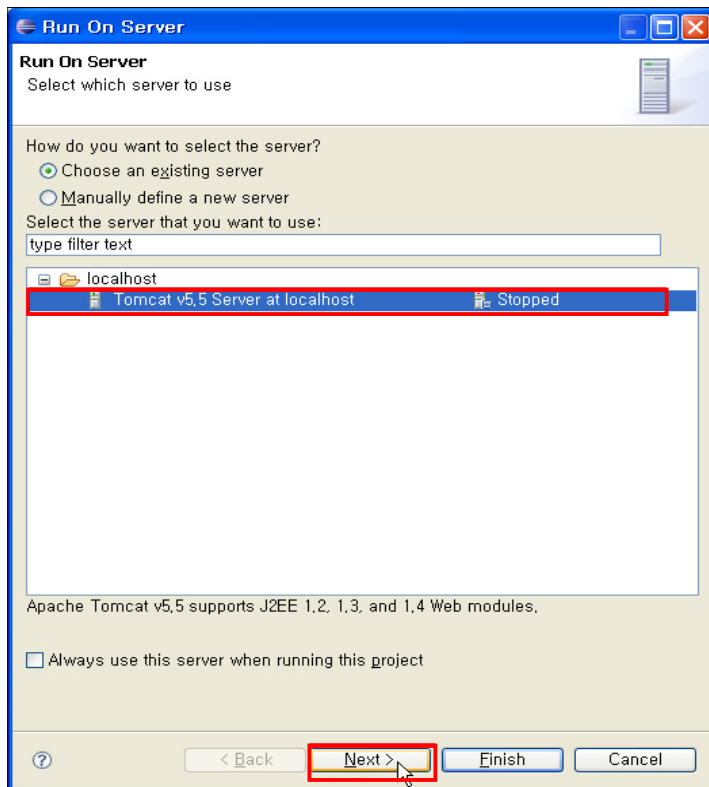


index.jsp를 선택후 오른쪽 마우스 단추를 클릭해서 [Run As]–[Run on Server]메뉴를 선택한다.



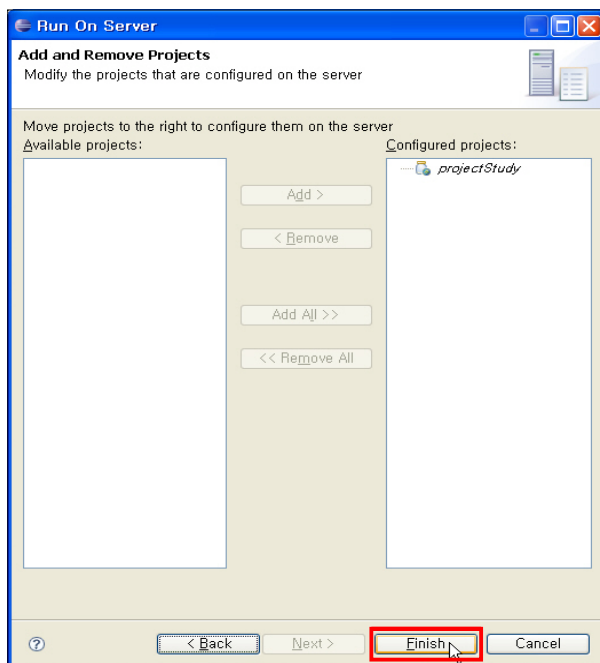
<이클립스에서 JSP 페이지 작성3>

⑤ [Run On Server]창이 표시되면 [localhost]–[Tomcat v5.5 Server at localhost]를 선택 후 [Next]단추를 클릭한다.



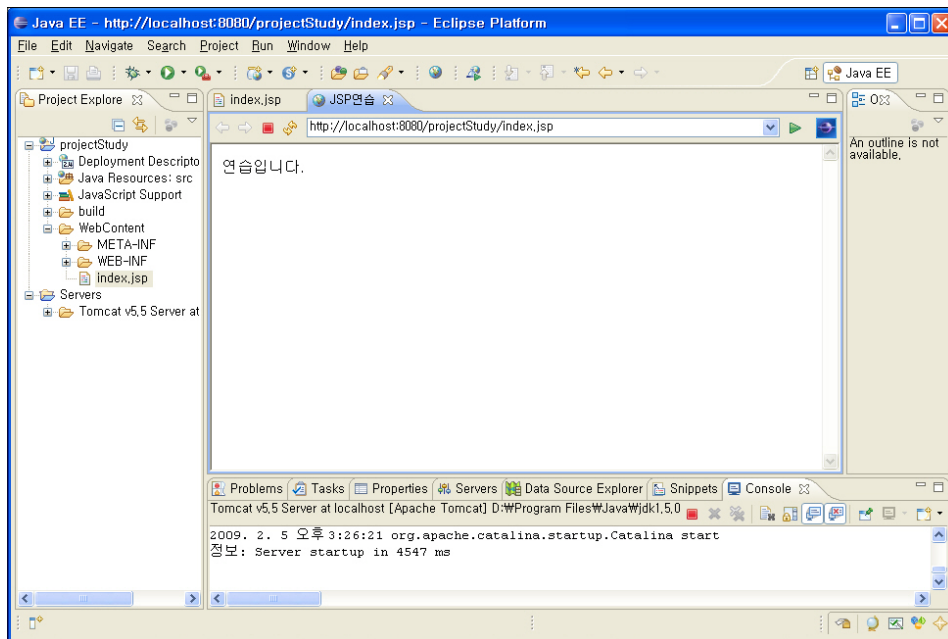
<이클립스에서 JSP 페이지 작성4>

⑥ [Finish] 단추를 클릭한다.



<이클립스에서 JSP 페이지 작성5>

⑦ 그러면 다음과 같이 index.jsp 페이지가 웹에서 서비스되는 것을 확인할 수 있다.

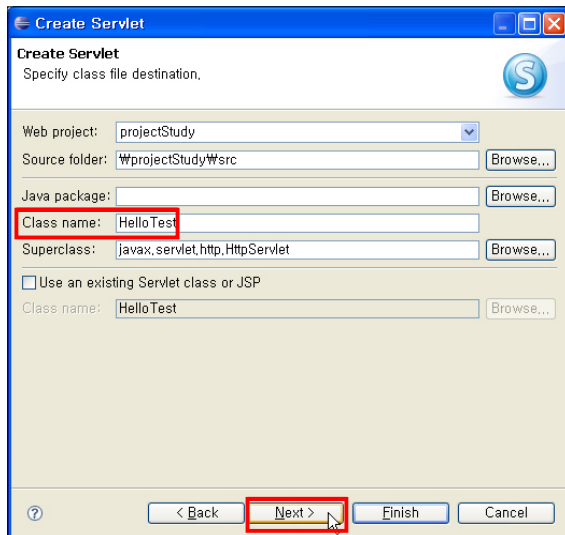


<이클립스에서 JSP 페이지 작성6>

4. 서블릿 페이지 작성

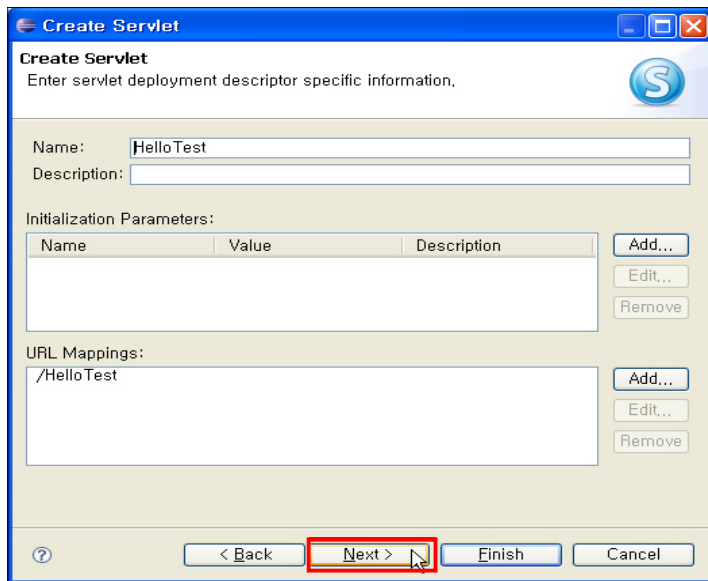
① [projectStudy] 프로젝트의 [WebContent] 폴더를 선택한 후, [File]–[New]–[Other] 메뉴를 선택후 [Web]–[Servlet] 항목을 선택해서 작성한다.

② [Create Servlet] 창이 표시되면 [Class name] 항목에 'HelloTest'를 입력후 [Next] 단추를 클릭한다.



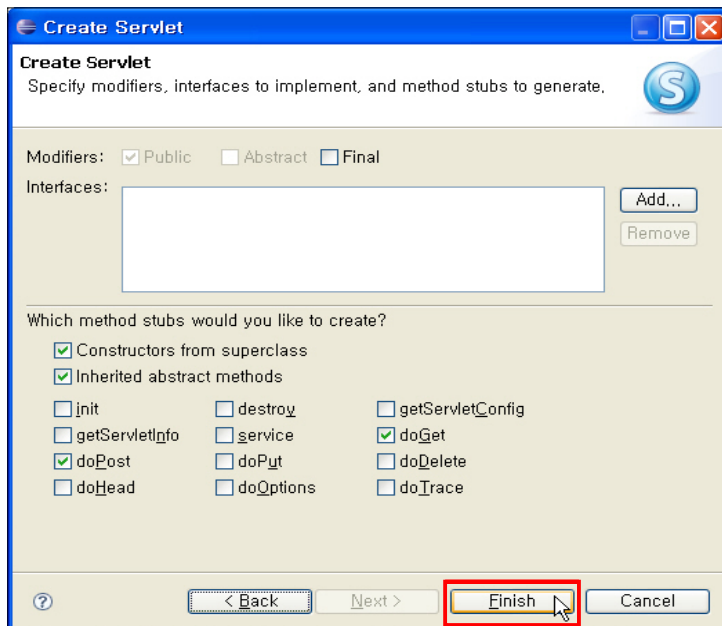
<이클립스에서 서블릿 페이지 작성 1>

③ 기본값을 그대로 사용하고 [Next] 단추를 클릭한다.



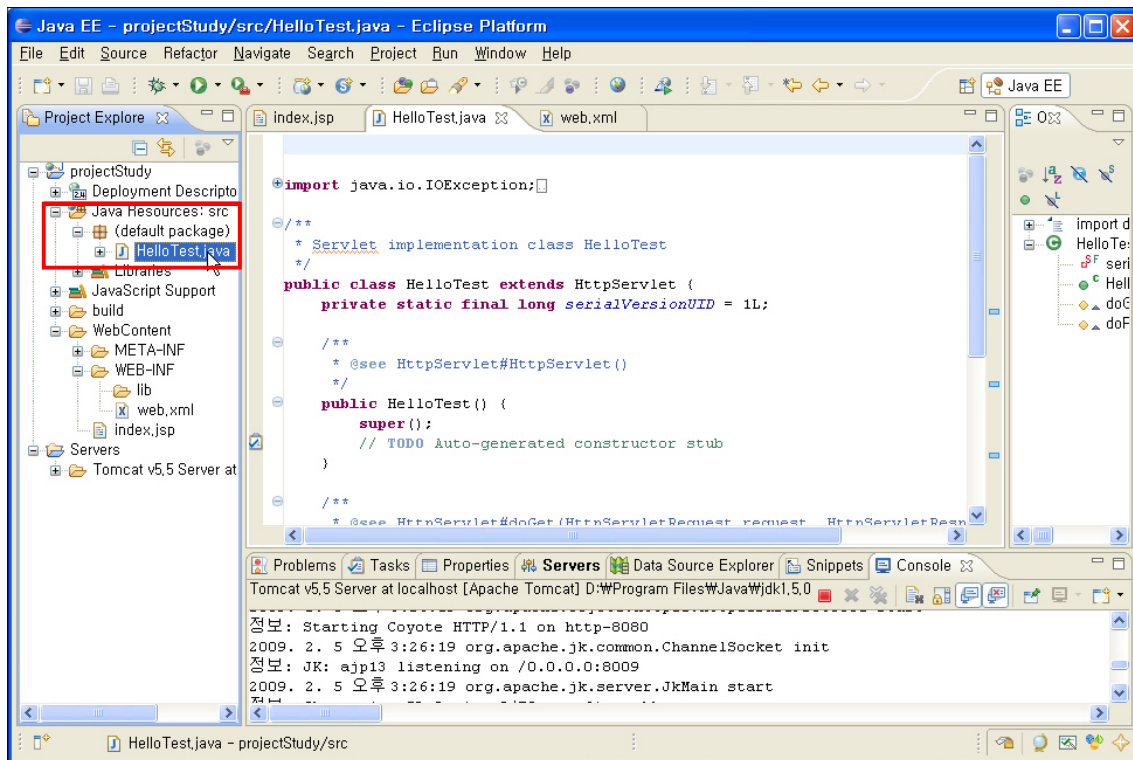
<이클립스에서 서블릿 페이지 작성 2>

④ 기본값을 그대로 사용하고 [Finish] 단추를 클릭한다.



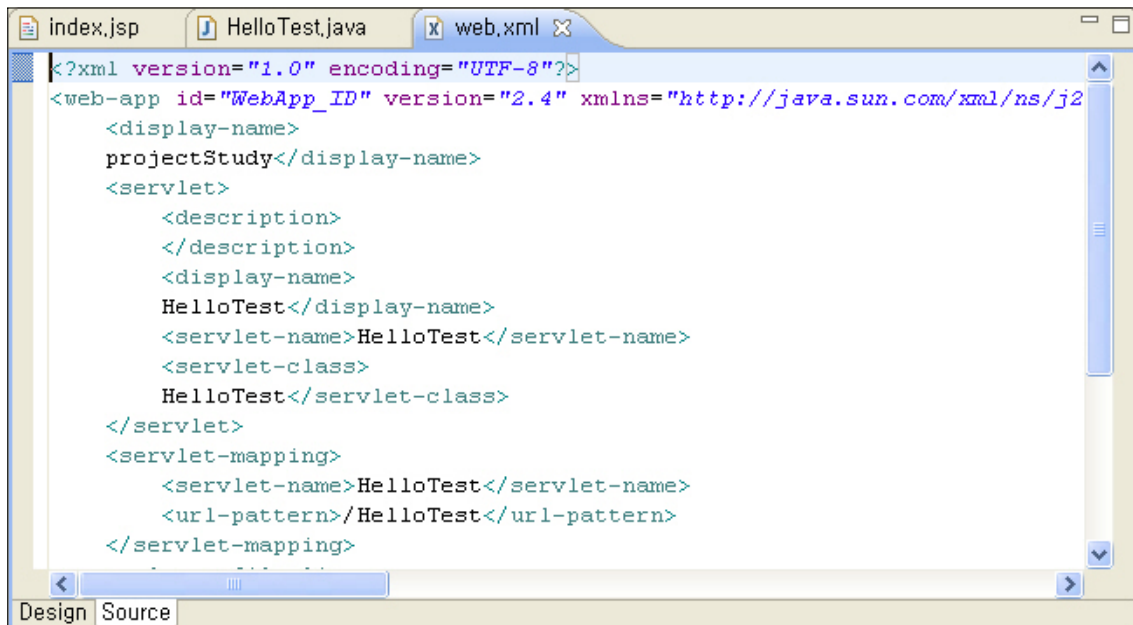
<이클립스에서 서블릿 페이지 작성 3>

⑤ [projectStudy]-[Java Resource : src]-[default package]항목안에 HelloTest.java파일이 위치한 것을 확인할 수 있다. 모든 .java파일(서블릿 포함)은 모두 [Java Resource : src]항목안에 위치한다. 패키지 없는 예제여서 [WebContent]폴더를 선택 후 작성했으나, 패키지가 있는 경우에는 [Java Resource : src]항목에 패키지를 작성후 해당패키지를 선택 후 서블릿을 작성해야 한다.



<이클립스에서 서블릿 페이지 작성 4>

또한 서블릿이 추가되면, 이 추가된 서블릿은 [projectStudy]프로젝트의 [WebContent]-[WEB-INF]안에 있는 web.xml에 추가된다.



HelloTest.java의 내용을 다음과 같이 수정 후 저장한다.

```
1 import java.io.IOException;
```

```

2  import java.io.PrintWriter;
3
4  import javax.servlet.ServletException;
5  import javax.servlet.http.HttpServlet;
6  import javax.servlet.http.HttpServletRequest;
7  import javax.servlet.http.HttpServletResponse;
8
9  /**
10   * Servlet implementation class HelloTest
11   */
12  public class HelloTest extends HttpServlet {
13      private static final long serialVersionUID = 1L;
14
15      /**
16       * @see HttpServlet#HttpServlet()
17       */
18      public HelloTest() {
19          super();
20          // TODO Auto-generated constructor stub
21      }
22
23      /**
24       * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
25       */
26      protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
27          // TODO Auto-generated method stub
28          response.setContentType("text/html;charset=euc-kr");
29          try{
30              PrintWriter out = response.getWriter();
31              out.println("<html>");
32              out.println("<head><title>Servlet 연습</title></head>");
33              out.println("<body>");
34              out.println("하하하.. Servlet 연습입니다.");
35              out.println("</body>");
36              out.println("</html>");
37          }catch(Exception e){
38              getServletContext().log("Error in HelloTest: ", e);
39          }
40      }

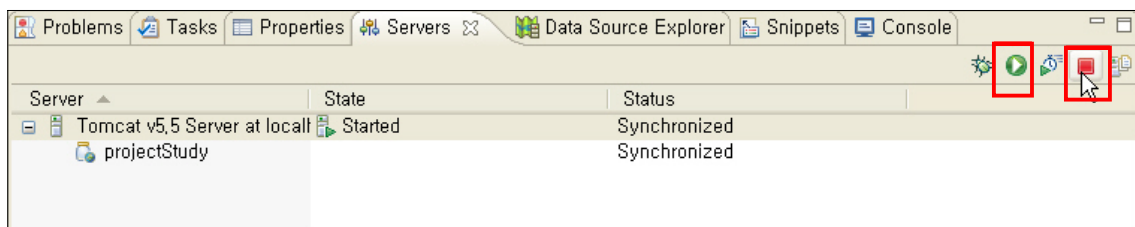
```

```

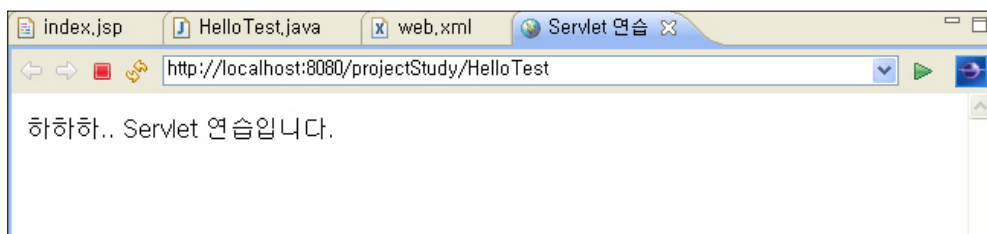
41
42     /**
43     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
44     */
45     protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
46         // TODO Auto-generated method stub
47     }
48
49 }

```

⑥ 서블릿의 경우 소스가 변경된 경우 서버를 내렸다 다시 올려야한다.



서버가 올라오면, HelloTest.java를 선택후 오른쪽 마우스 단추를 클릭해서 [Run As]-[Run on Server]메뉴를 선택한다. [Run On Server]창이 표시되면 [localhost]-[Tomcat v5.5 Server at localhost]를 선택 후 [Next],[Next],[Finish]단추를 클릭한다. 그러면 다음과 같 이 서블릿이 서비스 되는 것을 확인할 수 있다.



<이클립스에서 서블릿 페이지 작성 5>

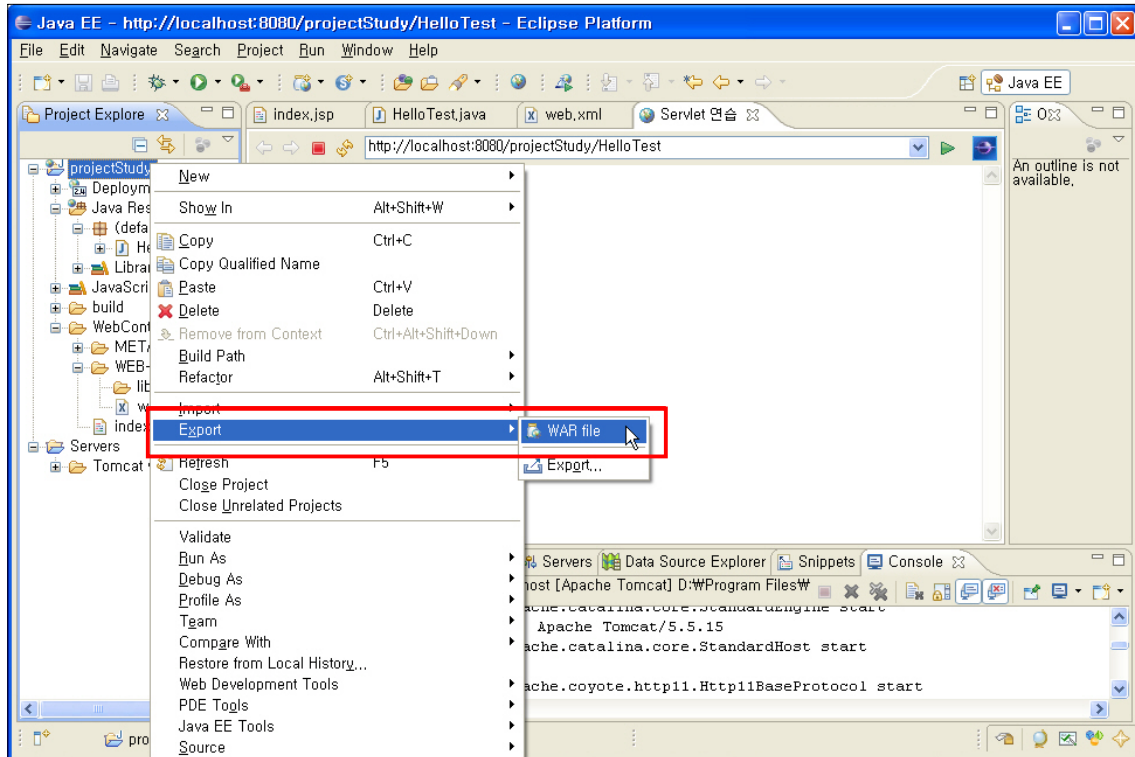
이제, 이클립스내의 환경에서 실행되고 있는 웹 어플리케이션을 실제로 서비스하기 위해 배포하는 것에 대해 알아보자.

5. WAR 배포

이클립스내의 환경에서 실행되는 것은 가상환경에서 실행되는 것이라 할 수 있다. 물론 실행결과는 실제로 서비스하는 것과 같다. 그러나 실제로 서비스 하기 위해서는 실제환경으로 웹 어플리케이션을 내보내서 실제환경에서 실행되도록 해야한다. 웹 어플리케이션을 서비스 하기 위해 WAR파일을 사용하는데, 이클립스에서는 이파일을 자동으로 생성해주는 기능이 있다.

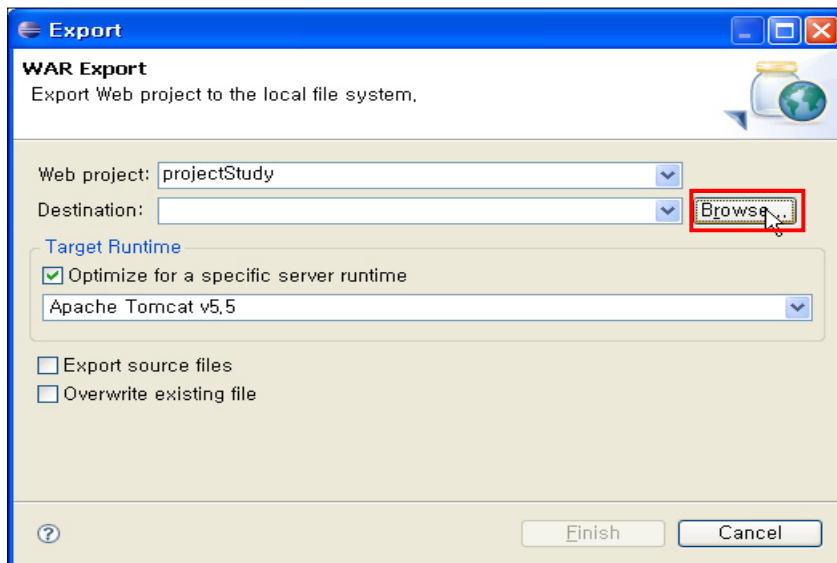
지금까지 작성한 [projectStudy] 프로젝트를 WAR파일로 내보내보자.

① 톰캣 서버가 올라와 있는 경우에 서버를 내린 후 [projectStudy] 프로젝트를 선택한 후 [Export]-[WAR file]메뉴를 선택한다.



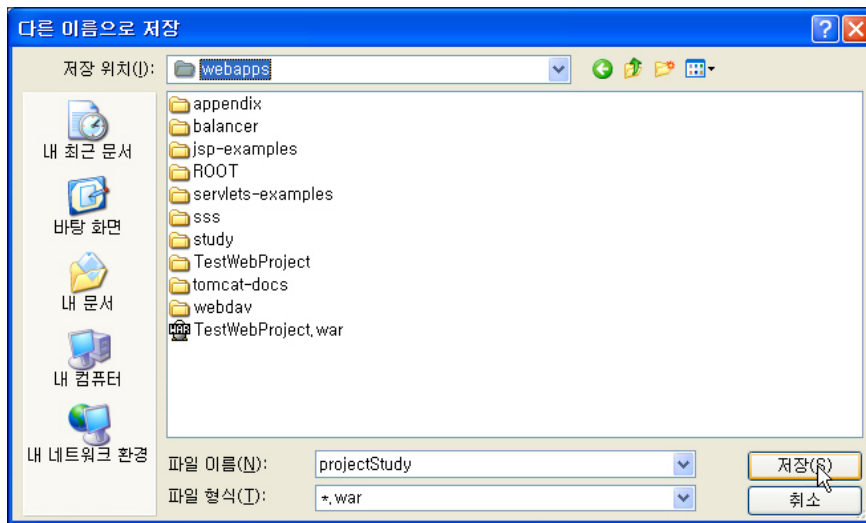
<[projectStudy] 프로젝트의 WAR 배포 1>

② [Export]창이 표시되면 [Browse]단추를 클릭해서 내보낼 위치를 지정한다.



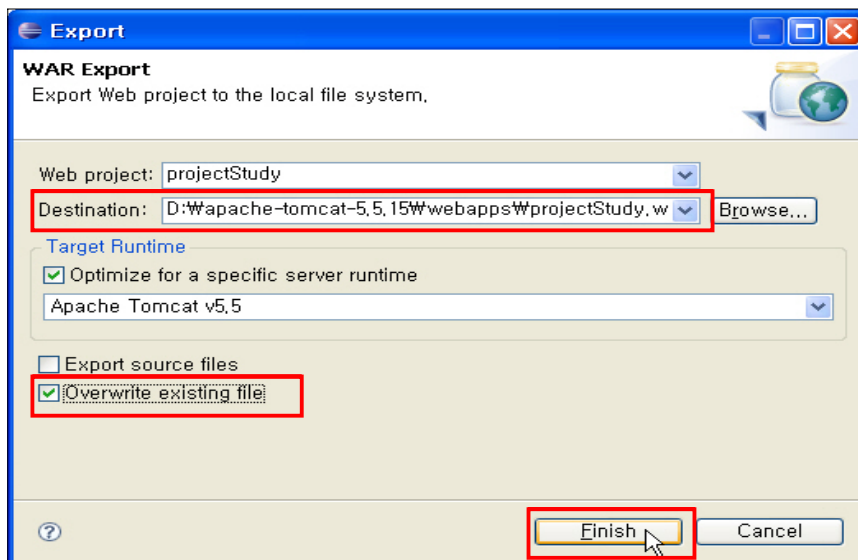
<[projectStudy] 프로젝트의 WAR 배포 2>

③ [다른 이름으로 저장] 대화상자가 표시되면 톰캣홈\webapps폴더를 선택 후 [저장] 단추를 클릭한다.



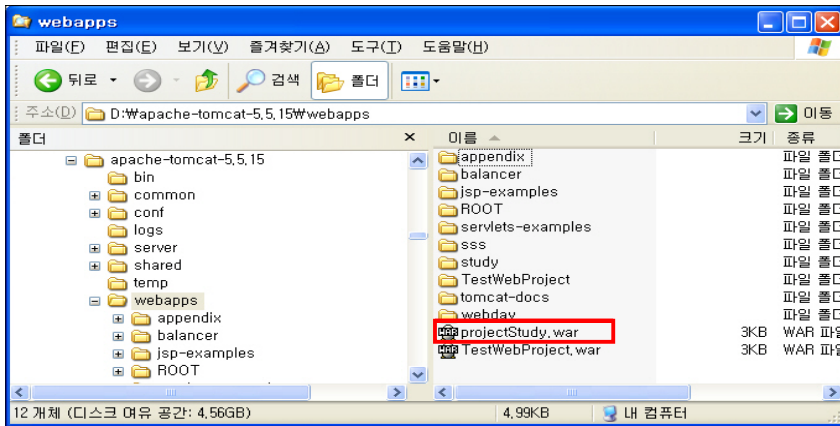
<[projectStudy]프로젝트의 WAR 배포 3>

④ 다음과 같이 선택된 부분을 확인 후 [Finish] 단추를 클릭한다. 이때 [Overwrite existing file] 항목은 기존에 같은 이름의 WAR파일이 있는 경우에는 덮어쓰기를 할 것인가의 여부를 지정하는 것으로 필자의 경우는 웹어플리케이션의 내용이 업데이트된 경우, 업데이트된 내용이 적용되도록 이 항목을 선택했다.



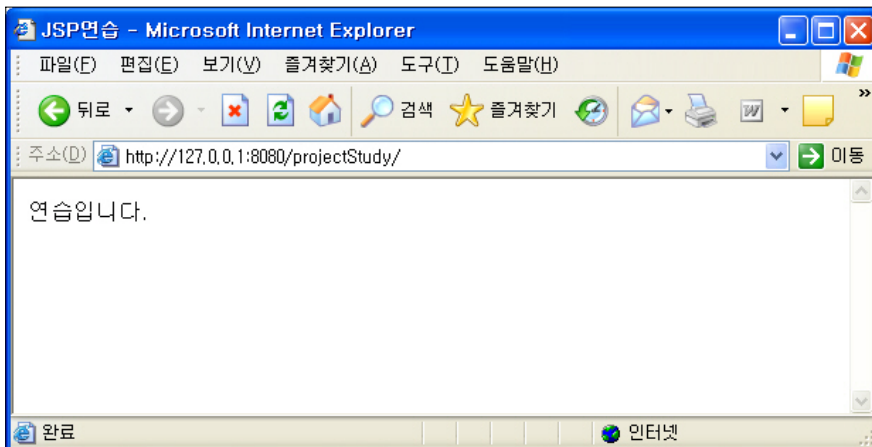
<[projectStudy]프로젝트의 WAR 배포 4>

⑤ 탐색기에서 톰캣홈\webapps폴더에 projectStudy.war파일이 생성된것을 확인할 수 있다.



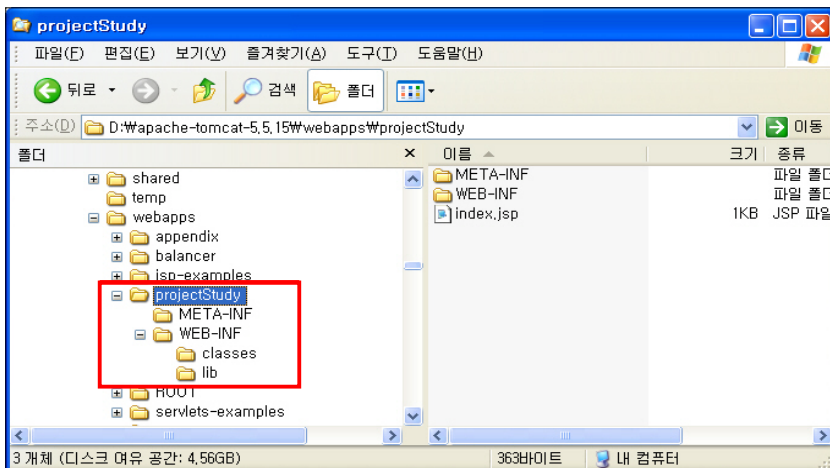
<[projectStudy]프로젝트의 WAR 배포 5>

⑥ 톰캣서버를 올린 후 웹 브라우저에 `http://127.0.0.1:8080/projectStudy/` 를 입력해서 실행하면, 이클립스에서 실행한 것과 같은 결과가 표시된다. 다만 차이는 아래의 화면은 실제로 사용자에게 서비스를 하는 화면인 것이다.



<[projectStudy]프로젝트의 WAR 배포 6>

⑦ 이렇게 WAR파일을 웹에서 실행하고 나면 톰캣홈\webapps폴더에 projectStudy.war파일의 내용이 풀려서 [projectStudy]웹 어플리케이션 폴더가 생성된 것을 확인 할 수 있다.



<[projectStudy]프로젝트의 WAR 배포 7>

6.동적 웹 프로젝트에서 패키지 작성과 작업폴더 추가

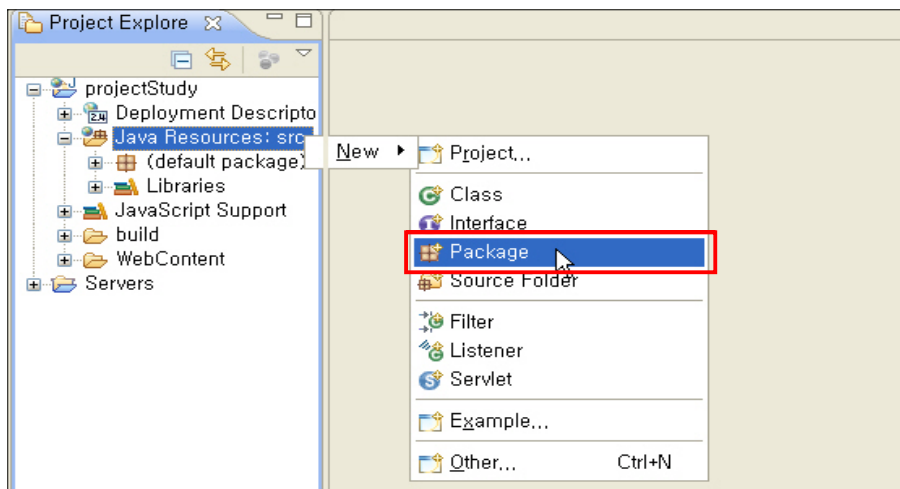
(1) 동적 웹 프로젝트에서 패키지 작성

일반적으로 로직을 갖는 자바파일이나 서블릿파일등의 .java파일은 디폴트(default)패키지에 그냥 저장하지 않는다. 웹어플리케이션이 동작되는데는 각각의 작업에 관련되는 여러개의 로직파일들이 있는데, 이들은 폴더와 같은것을 만들어서 그 폴더에 관련 파일들을 저장하면 관리하기도 편하다.

이 폴더와 같은 역할을 하는 것이 패키지이다. 일반적으로 상용을 목적으로 만드는 패키지의 경우에는 유일한 이름을 갖는 것을 목적으로 하기 때문에 자사의 URL이 주로 사용된다. 예를 들어 회사의 URL이 `http://www.test.com` 이라면 패키지명은 `com.test`와 같은 형식으로 만들어지게 된다.

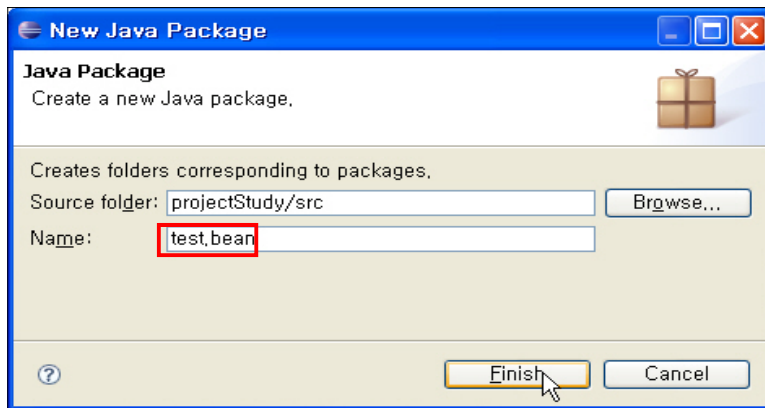
이클립스의 동적 웹프로젝트에도 패키지를 추가할 수 있는데, 여기서는 그방법에 대해 알아보자. 자바빈에서 데이터를 저장하는 빈을 작성하는 것을 예시로써 작성해보도록 하겠다.

① [Java Resource : src]를 선택하고 마우스 오른쪽 단추를 클릭해서 [New]-[Package] 메뉴를 선택한다.



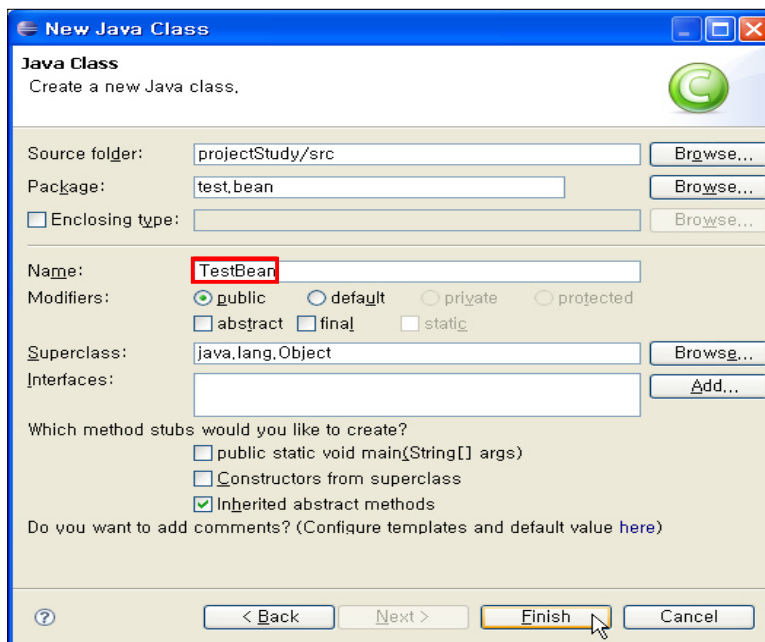
<동적 웹 프로젝트에서 패키지 작성 1>

② [New Java Package]창이 표시되면 [Name]항목에 패키지명을 'test.bean'으로 입력하고 [Finish]단추를 클릭한다.



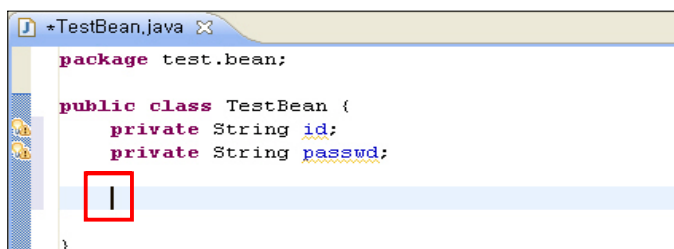
<동적 웹 프로젝트에서 패키지 작성 2>

③ [Java Resource : src] 아래에 [test.bean]패키지가 생성되면, [test.bean]패키지를 선택하고 [File]-[New]-[Other] 메뉴를 선택후 [Java]-[Class]항목을 선택한다. [New Java Class]창이 표시되면 클래스명을 기술하는 [Name]항목에 클래스명으로 'TestBean'을 입력 후 [Finish]단추를 클릭한다.



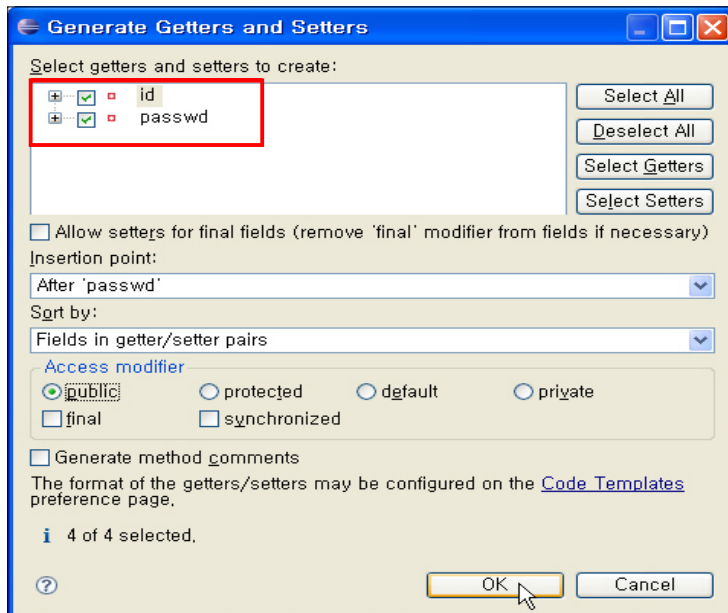
<동적 웹 프로젝트에서 패키지 작성 3>

④ 편집기 뷰에 TestBean.java의 파일이 표시되면 커서를 다음과 같이 위치 후 [Source]-[Generate Getters and Setters]메뉴를 선택한다.



<그림3판용53.jpg><동적 웹 프로젝트에서 패키지 작성 4>

⑤ [Generate Getters and Setters]창이 표시되면 Getters/ Setters메소드를 작성할 프로퍼티를 선택후 [OK]단추를 클릭한다. 그러면 화면에 소스가 완성된다. 자바빈에서 데이터를 저장하는 빈을 만들때 이와같이 작성한다.

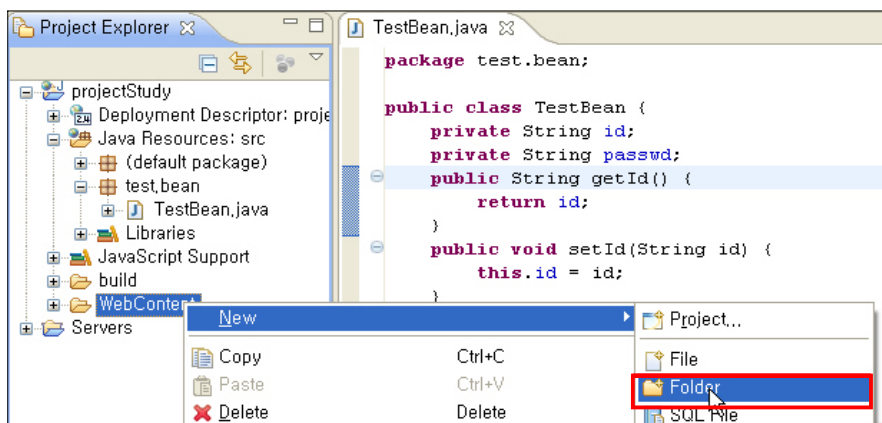


<그림3판용54.jpg><동적 웹 프로젝트에서 패키지 작성 5>

(2) 동적 웹 프로젝트에 작업폴더 추가

마찬가지로 JSP웹페이지도 [WebContent]폴더에 그냥 저장하는 것이 아니라, 파일의 관리를 위해서 작업폴더를 작성할 후 그 작업폴더에 관련파일들을 저장한다. 그럼 작업폴더를 추가하는 방법에 대해서도 알아보자.

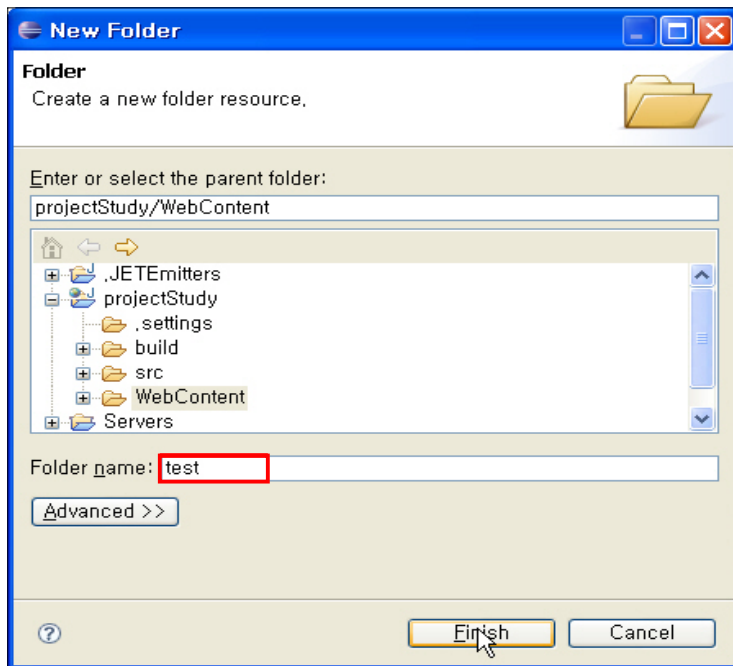
① 프로젝트의 [WebContent]폴더를 선택하고 마우스 오른쪽 단추를 클릭해서 [New]-[Folder]메뉴를 선택한다.



<동적 웹 프로젝트에 작업폴더 추가 1>

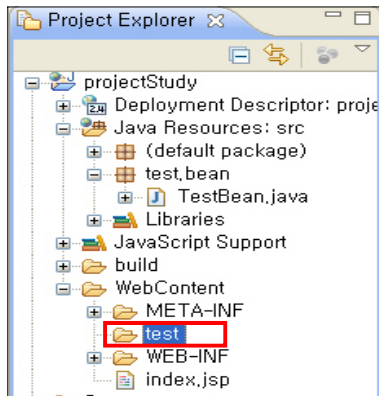
② [New Folder]창이 표시되면 [Enter or select the parent folder]항목의 값이

'projectStudy/WebContent'이면 [Folder name]항목에 추가할 폴더명으로 'test'를 입력 하고 [Finish]단추를 클릭한다.



<동적 웹 프로젝트에 작업폴더 추가 2>

③ 작업폴더가 다음과 같이 추가된 것을 알 수 있다. 해당 폴더에 JSP페이지를 추가하려면 대상작업폴더를 선택 후 마우스 오른쪽 단추를 클릭해서 [New]-[JSP]메뉴를 선택해서 작성한다.



<동적 웹 프로젝트에 작업폴더 추가 3>