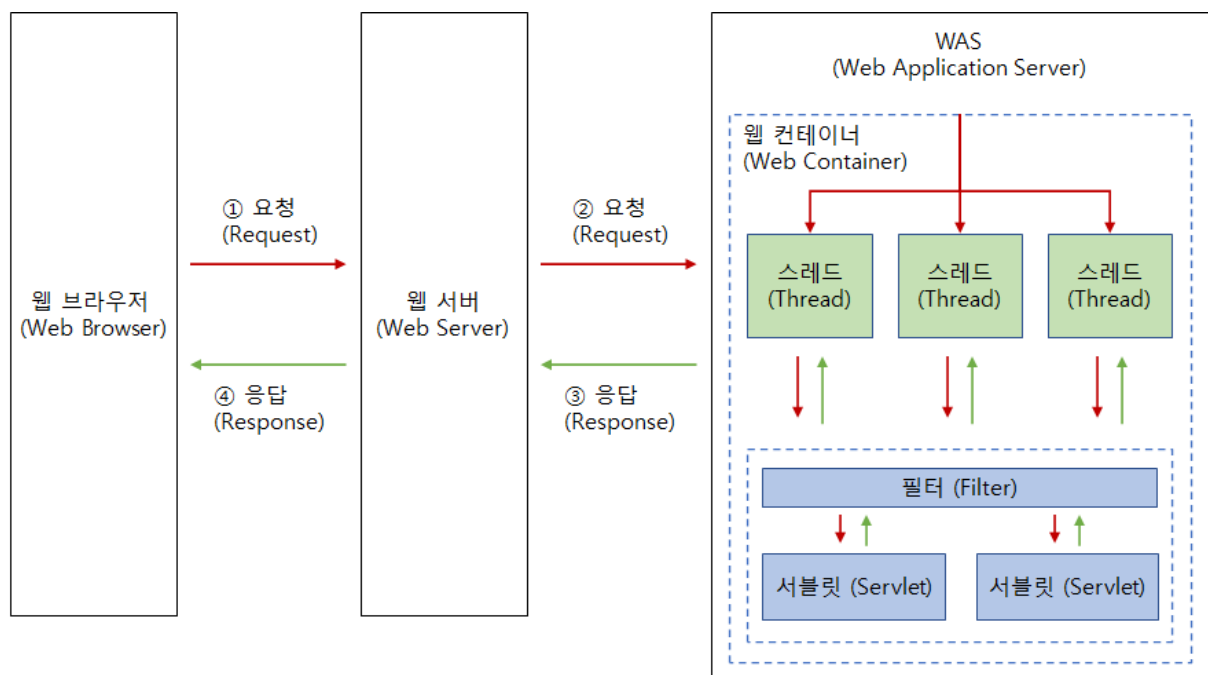


Dynamic Web project 1

Dynamic Web Project(다이나믹 웹 프로젝트)는 Java Servlet(서블릿) 기반의 웹 애플리케이션입니다.

Servlet(서블릿)은 웹 브라우저(Web Browser)에서 요청(Request)이 들어오면 웹 컨테이너(Web Container, Servlet Container)에서 웹 페이지를 동적으로 생성하고 웹 브라우저에 응답(Response)하는 Server-Side(서버사이드) 자바 프로그램입니다.



웹 서버(Web Server, HTTP Server)는 웹 브라우저(Web Browser)에서 HTTP 요청 (Request)을 받아 정적인 콘텐츠(html, image(jpg, gif, png 등등), css, script)를 응답 (Response)합니다.

웹 서버로는 Apache Server, Nginx, IIS, WebToB 등이 많이 사용됩니다.

Web Container(웹 컨테이너)는 요청되는 URL과 매핑되는 서블릿을 호출하기 위해 스레드를 생성하고 서블릿을 처리합니다. (WAS가 처리하는 것이 Web Container입니다. 설명을 위해 표시한 겁니다.)

그리고 Web Container(웹 컨테이너)는 서블릿뿐만 아니라 JSP 그리고 Server-Side(서버사이드) 코드가 있는 프로그램 파일들을 관리하고 처리합니다.

웹 응용프로그램 서버로는 Apache Tomcat, Web Logic, Web Sphere, JBOSS, JBoss 등 이 많이 사용됩니다.

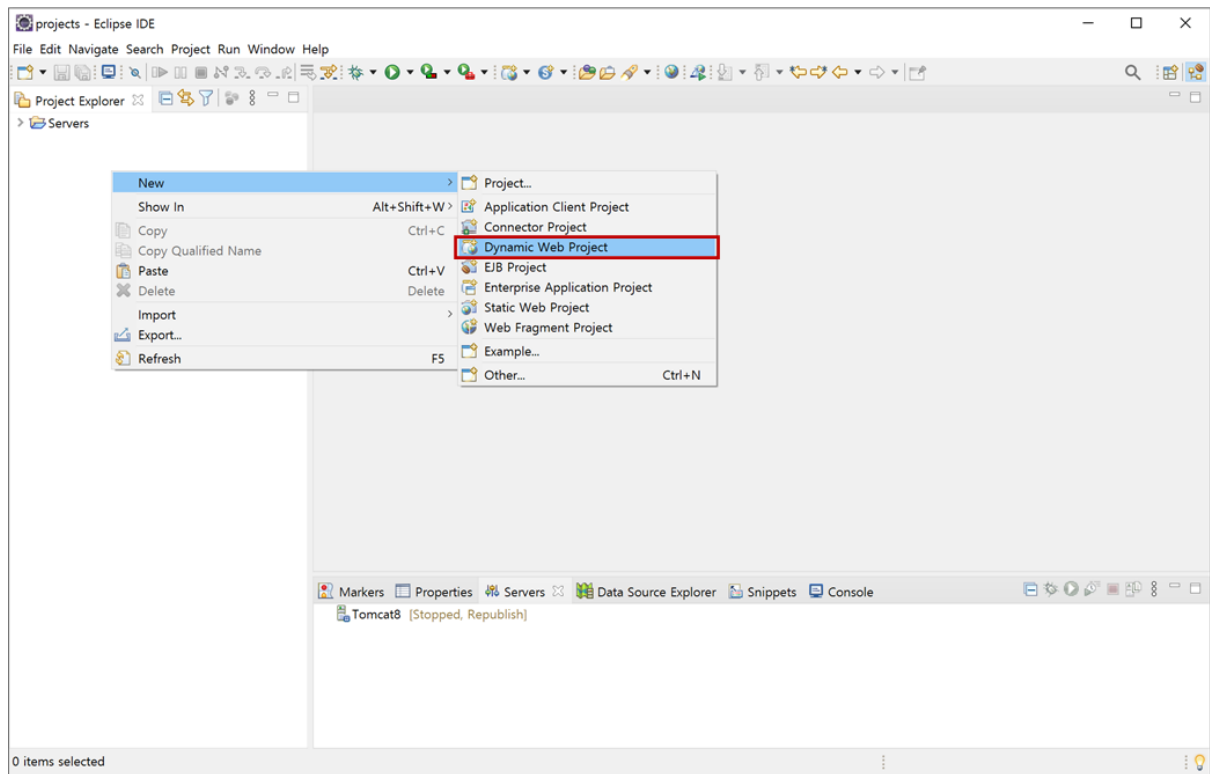
JSP vs Servlet

JSP(Java Server Pages)는 Java를 이용한 Server-Side 스크립트 언어로 HTML 안에 Scriptlet(스크립틀릿)으로 자바 소스 코드(<% %>)를 포함하고 있습니다. 파일 확장자는 ".jsp"이고 WAS에서 실행될 때 동적으로 javax.servlet.http.HttpServlet 클래스를 상속받은 Java 소스 코드로 변환되고 컴파일(Compile)되어 실행됩니다. 이처럼 JSP 파일을 Servlet 클래스로 변환하고 실행시켜 주는 역할을 하는 프로그램을 Servlet Container (서블릿 컨테이너)라고 합니다.

Servlet(서블릿)은 Java를 이용해 웹 페이지를 동적으로 생성하는 Server-Side 프로그램 (Java 클래스)로 자바 소스 코드 안에 HTML를 포함하고 있습니다. 파일 확장자는 ".java"이고 배포하기 위해서는 컴파일(Compile)을 해야 합니다. 컴파일된 파일 확장자는 ".class"입니다.

JSP(Java Server Pages)와 Servlet(서블릿)에 대해 간단하게 소개해드렸습니다. 이제 Dynamic Web Project(다이나믹 웹 프로젝트)를 만들어 보겠습니다.

1. "Project Explorer"에서 오른쪽 버튼을 클릭하여 메뉴에서 [New > Dynamic Web project]를 클릭합니다. 또는 전체 메뉴에서 [File > New > Dynamic Web project]를 클릭합니다.



2. "New Dynamic Web project"창이 나타납니다.

New Dynamic Web Project

Dynamic Web Project
Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name:

Project location
☒ Use default location
Location:

Target runtime

Dynamic web module version

Configuration

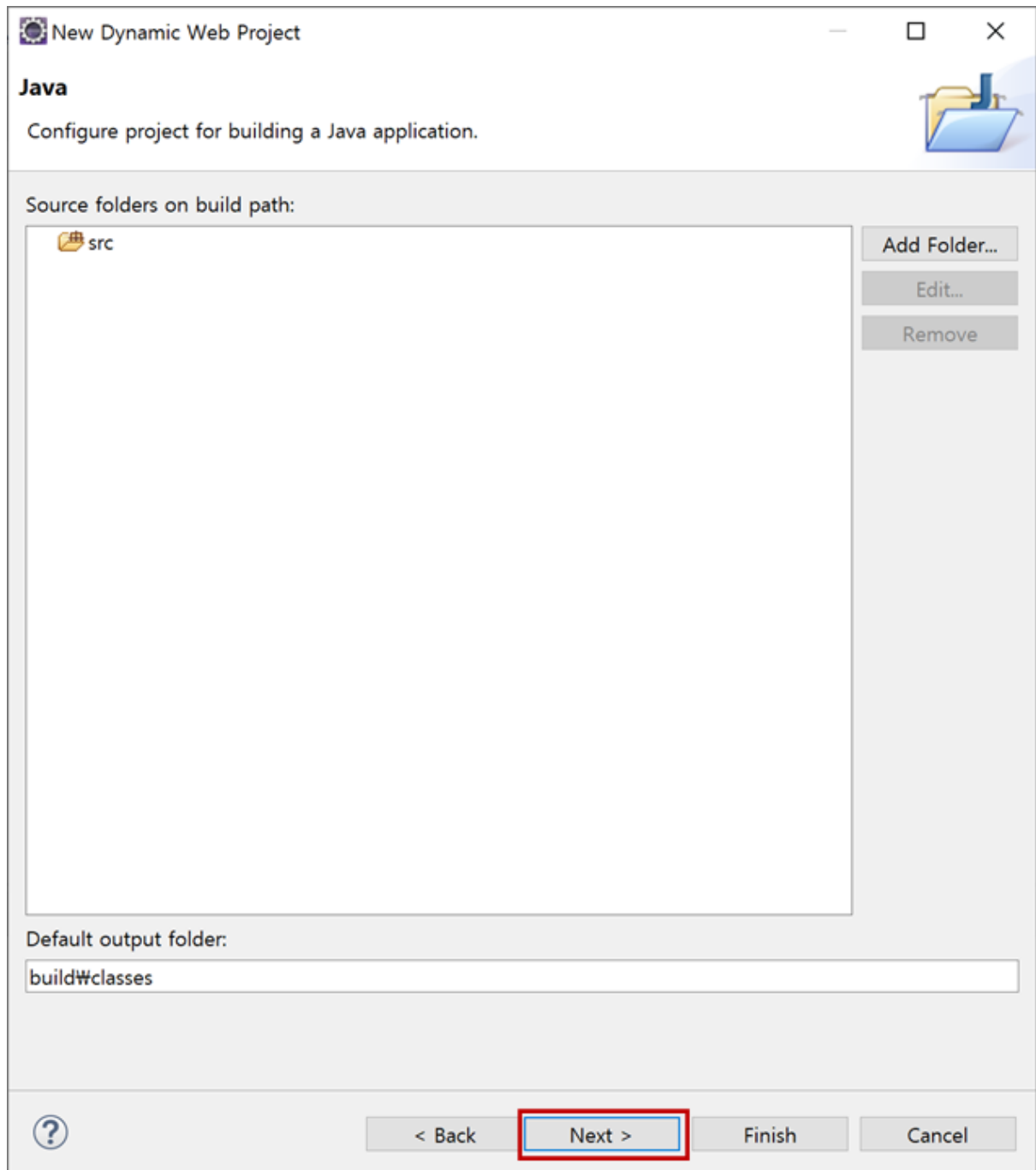
The default configuration provides a good starting point. Additional facets can later be installed to add new functionality to the project.

EAR membership
☐ Add project to an EAR
EAR project name:

Working sets
☐ Add project to working sets
Working sets:

"Project name"에 웹 프로젝트 이름을 입력합니다. "Test1"로 입력하고 "Next"버튼을 클릭합니다. 또는 "Finish"버튼을 클릭해서 완료할 수 있습니다.

자바 애플리케이션의 환경을 설정합니다.



자바 소스 폴더 경로와 출력 폴더를 변경할 수 있습니다. 자바 소스 폴더를 변경하거나 추가 또는 삭제할 수 있습니다. "Next"버튼을 클릭합니다.

웹 모듈의 환경을 설정합니다.

New Dynamic Web Project

Web Module

Configure web module settings.

Context root:

Content directory:

☐ Generate web.xml deployment descriptor

? < Back Next > **Finish** Cancel

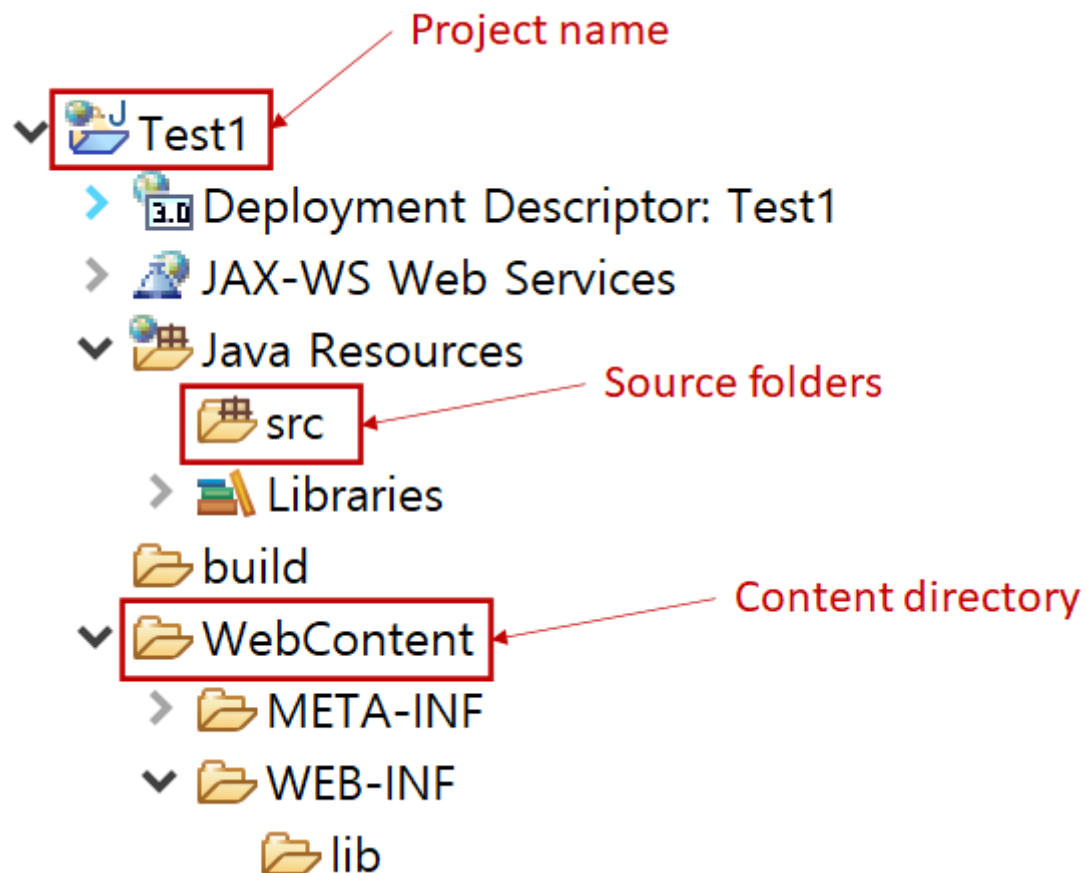
"Context root"는 Web Application Server(WAS)에서 웹 애플리케이션을 구분하기 위한 경로를 말합니다. (예를 들면, "http://localhost:8080/Test1/"입니다.)

"Content directory"는 WAS에서 호출되어 실행되는 Java Class 파일, Jsp 파일, html 파일 등등 웹 파일들이 위치하는 폴더입니다.

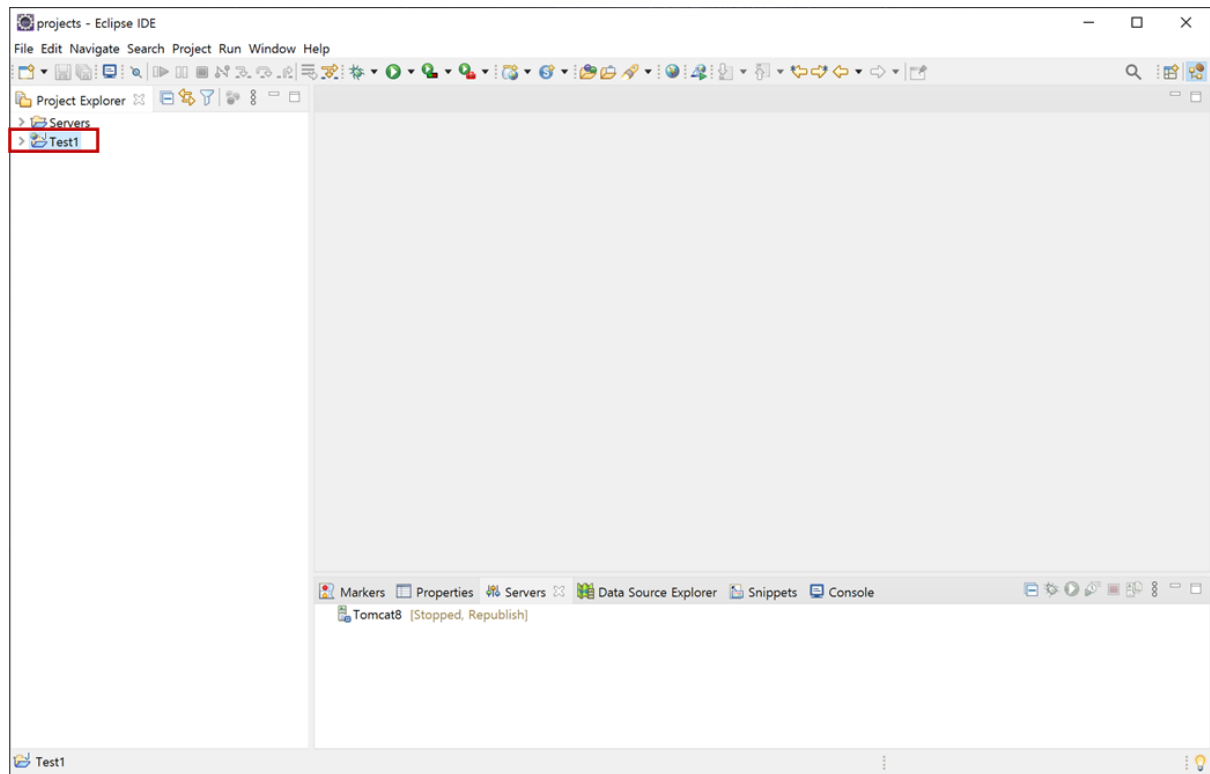
"Generate web.xml deployment descriptor"를 체크하면 "Content directory"에 "web.xml"이 자동으로 생성됩니다.

"Finish"버튼을 클릭합니다.

위와 같이 설정하면 다음처럼 웹 프로젝트 구조가 설정됩니다.



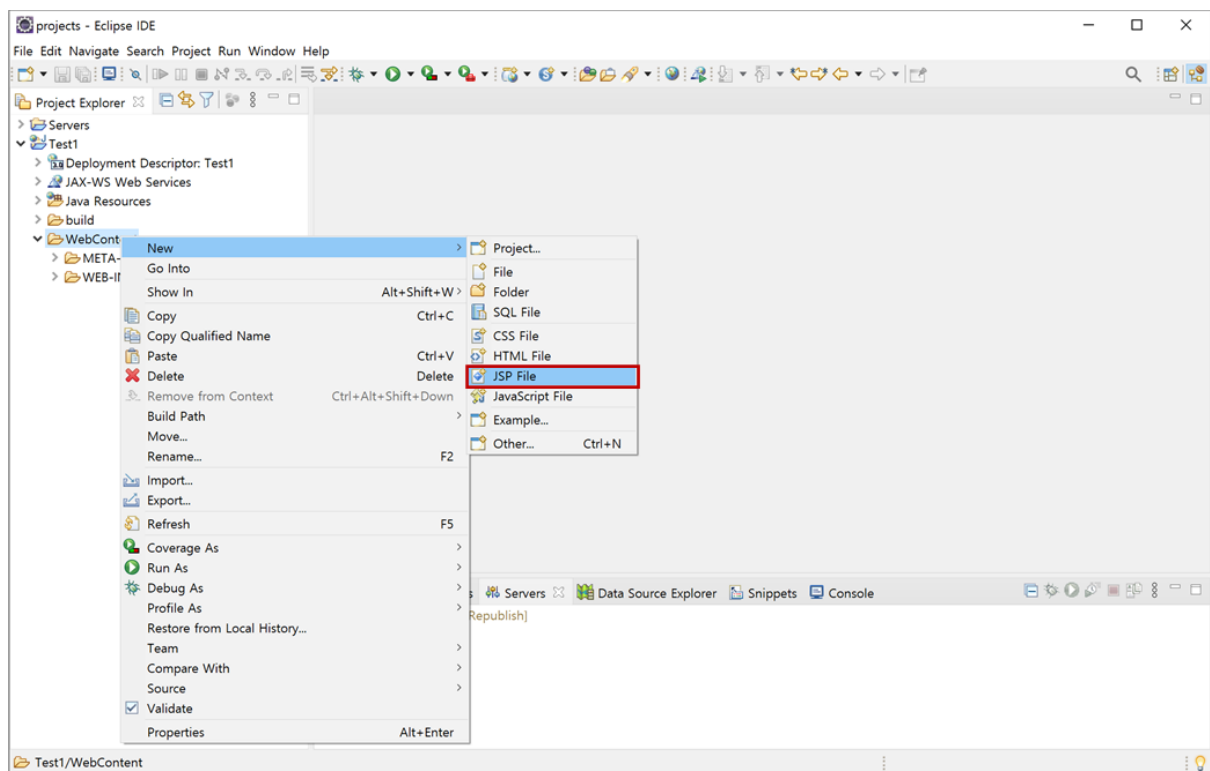
"Project Explorer"에 "Test1"으로 웹 프로젝트가 생성됩니다.



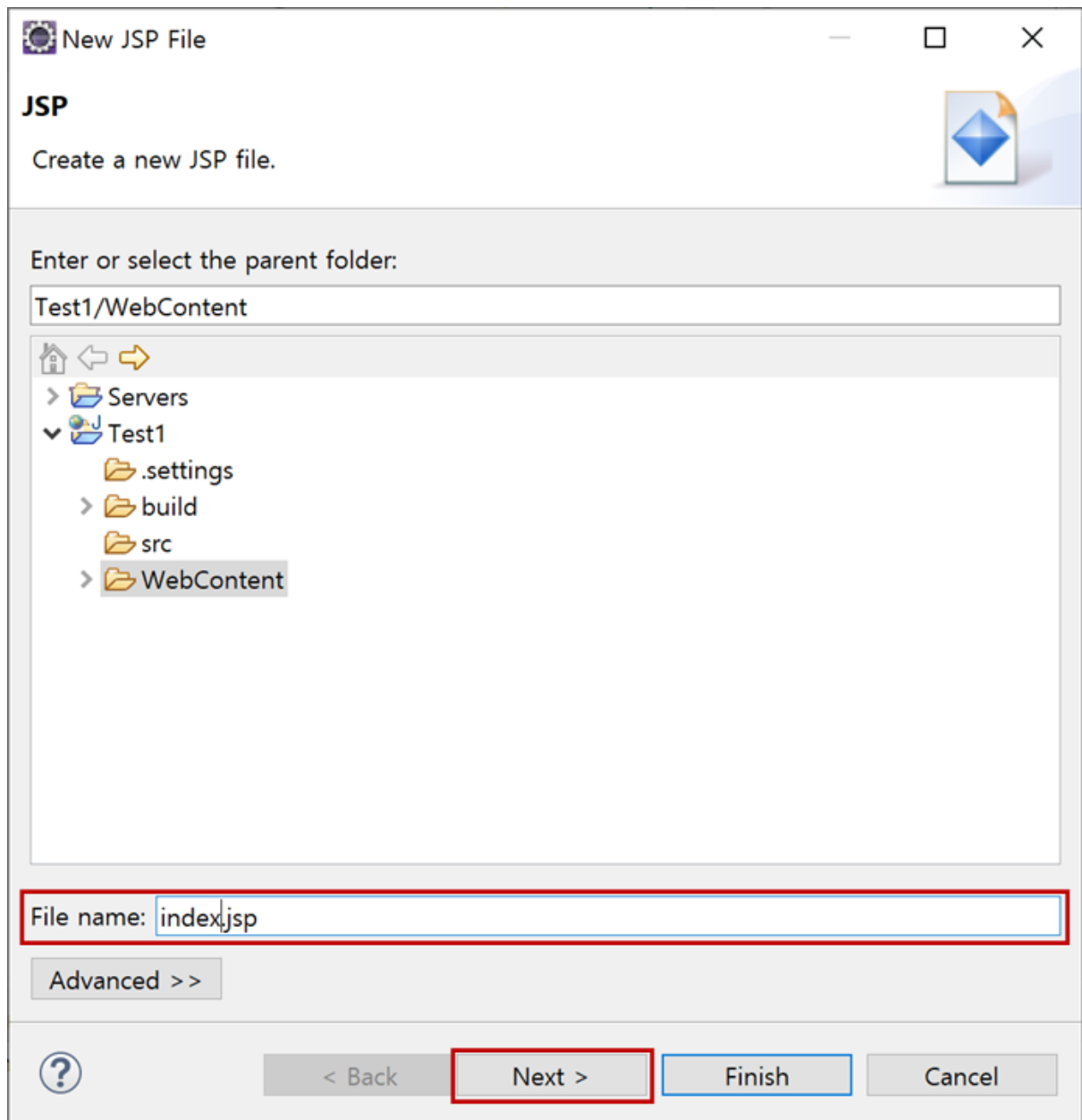
JSP파일 또는 Servlet파일을 생성합니다. 테스트를 위해 각각 생성하겠습니다.

JSP 파일 생성

3. "Test1" 웹 프로젝트를 확장시킨 후 "WebContent"에서 오른쪽 버튼을 클릭하여 [New > JSP File]를 클릭합니다.

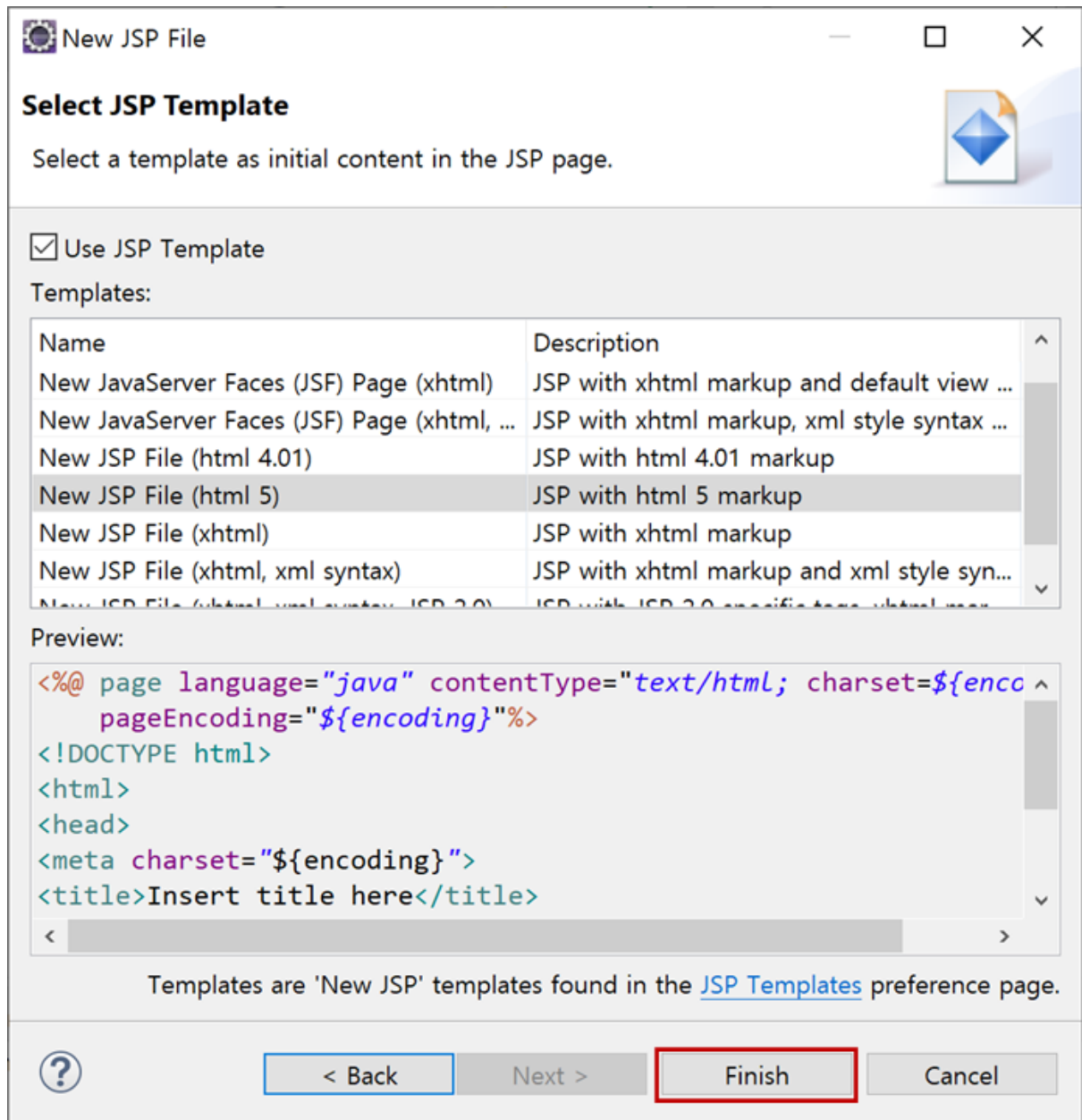


4. "New JSP File"창이 나타납니다. "File name"에 JSP 파일 이름을 입력합니다.



"index.jsp"로 입력하고 "Next"버튼을 클릭합니다. 또는 "Finish"버튼을 클릭해서 완료할 수 있습니다.

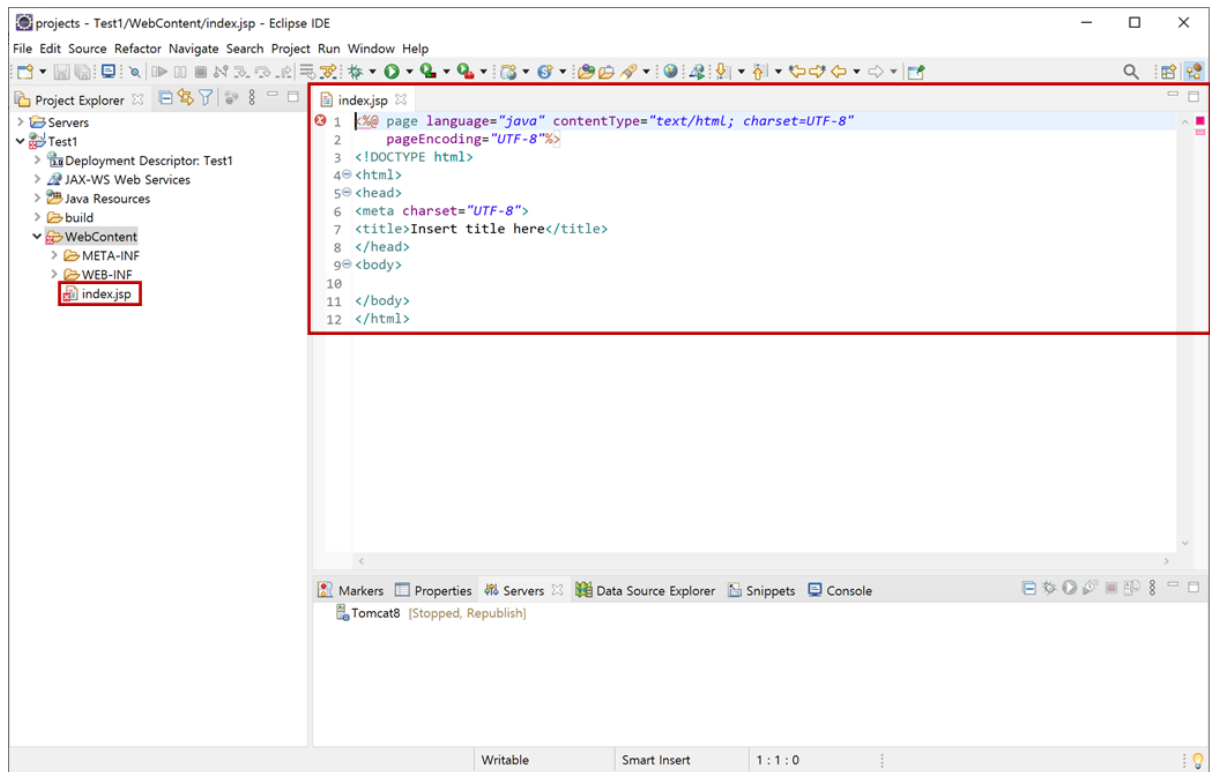
JSP 템플릿을 선택합니다.



변경하고자 하는 템플릿을 선택하고 "Finish"버튼을 클릭합니다.

만약, "Use JSP Template"를 체크하지 풀고 완료하면 내용 없이 JSP 파일이 생성됩니다.

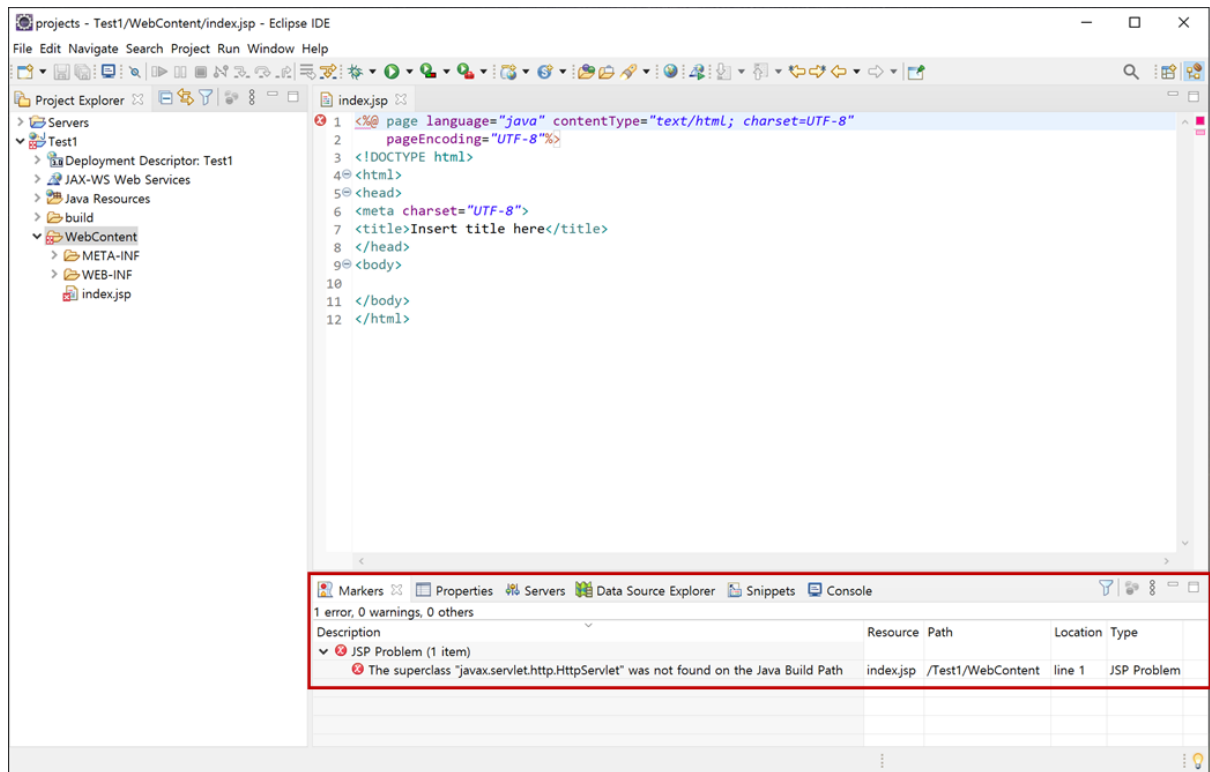
5. "WebContent"에 "index.jsp"파일이 생성되고 에디터에 템플릿이 적용되어 나타납니다.



그렇지만, 에러가 발생하였습니다.

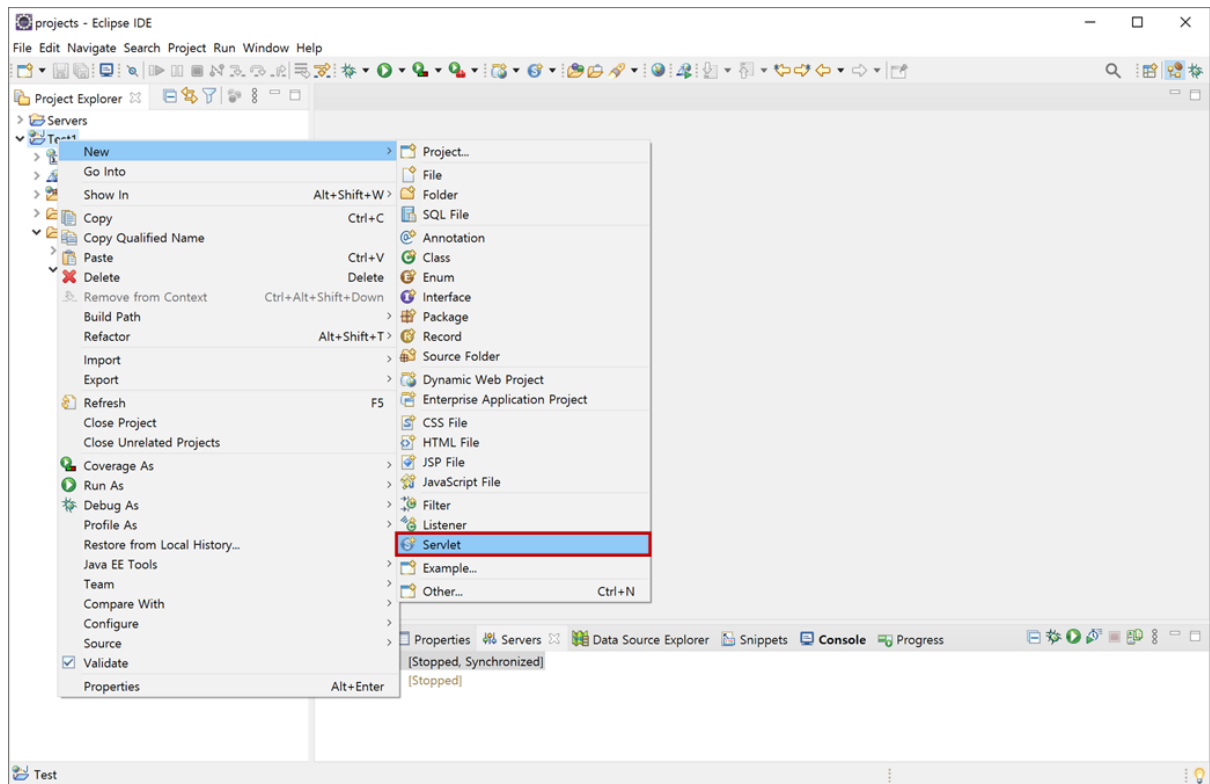
하단에 "Markers"탭을 클릭하면 에러 메시지를 자세히 볼 수 있습니다.

에러의 원인은 "The superclass "javax.servlet.http.HttpServlet" was not found on the Java Build Path"으로 Java 빌드 경로에서 수퍼 클래스인 "javax.servlet.http.HttpServlet"을 찾을 수 없다는 것입니다.

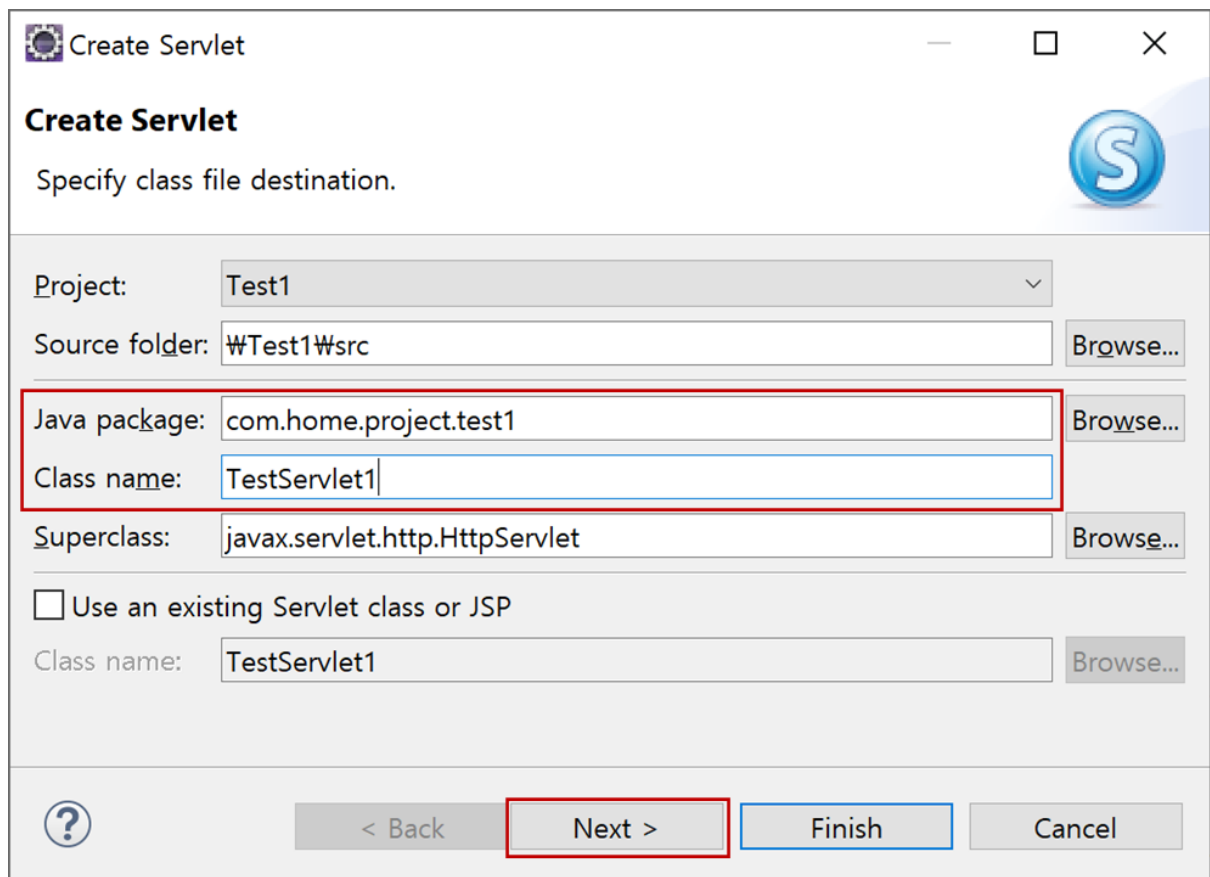


Servlet파일 생성

3. "Test1"에서 오른쪽 버튼을 클릭하여 컨텍스트 메뉴에서 [New > Servlet]를 클릭합니다.



4. "Create Servlet"창이 나타납니다. Servlet를 생성하기 위해 "Java package"와 "Class name"를 입력합니다.



"Java package"는 프로젝트 또는 라이브러리를 식별하기 위해 네임스페이스(namespace)로 클래스 파일들이 저장되는 경로(디렉터리 구조와 같음)로 입니다. 다른 패키지들과 구분되는 고유 값이어야 합니다.

패키지 명의 첫 번째는 회사나 조직 이름 또는 대표 도메인(URL 주소인 도메인을 역순으로 적용)을 사용하고 그다음은 프로젝트 또는 서비스나 모듈 명을 사용합니다. '.'(도트)을 이용하여 구분합니다.

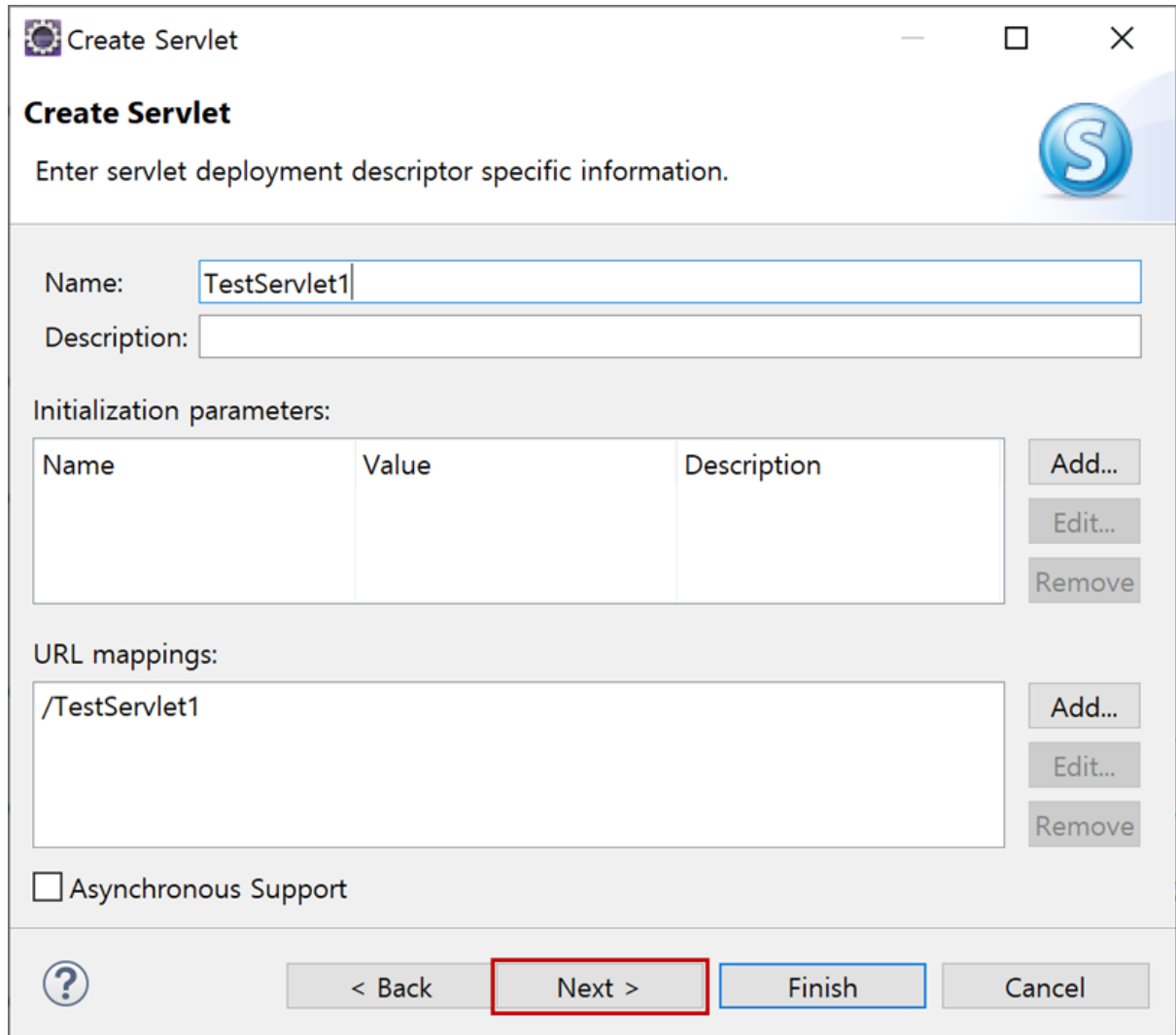
개인적인 개발 프로젝트임으로 가상 도메인인 "www.home.com"을 역순으로 적용한 "com.home"과 프로젝트 명으로 "project.test1"을 결합하여 사용하겠습니다.

"Java package"에는 "com.home.project.test1"을 입력합니다.

"Class name"은 서블릿명입니다. "TestServlet1"를 입력합니다.

입력하고 "Next >"버튼을 클릭합니다. 또는 "Finish"버튼을 클릭해서 완료할 수 있습니다.

서블릿 설명, 파라미터 설정, 웹 경로를 설정할 수 있습니다. "Next >"버튼을 클릭합니다. 다음에 자세한 설정에 대해 알려드리겠습니다.



The image shows a 'Create Servlet' dialog box with the following fields and controls:

- Name:** TestServlet1
- Description:** (empty)
- Initialization parameters:** A table with columns Name, Value, and Description. To the right are buttons: Add..., Edit..., and Remove.
- URL mappings:** /TestServlet1. To the right are buttons: Add..., Edit..., and Remove.
- ☐ Asynchronous Support
- Navigation buttons:** < Back, Next > (highlighted with a red border), Finish, and Cancel.

서블릿의 인터페이스, 메서드들을 선택하여 클래스를 생성할 수 있습니다. 기본 설정으로 클래스를 생성하였습니다. "Finish"버튼을 클릭합니다. 다음에 자세한 설정에 대해 알려드리겠습니다.

Create Servlet

Create Servlet

Specify modifiers, interfaces to implement, and method stubs to generate.

Modifiers: ☒ public ☐ abstract ☐ final

Interfaces:

Add...

Remove

Which method stubs would you like to create?

☒ Constructors from superclass

☒ Inherited abstract methods

☐ init

☐ destroy

☐ getServletConfig

☐ getServletInfo

☐ service

☒ doGet

☒ doPost

☐ doPut

☐ doDelete

☐ doHead

☐ doOptions

☐ doTrace

?

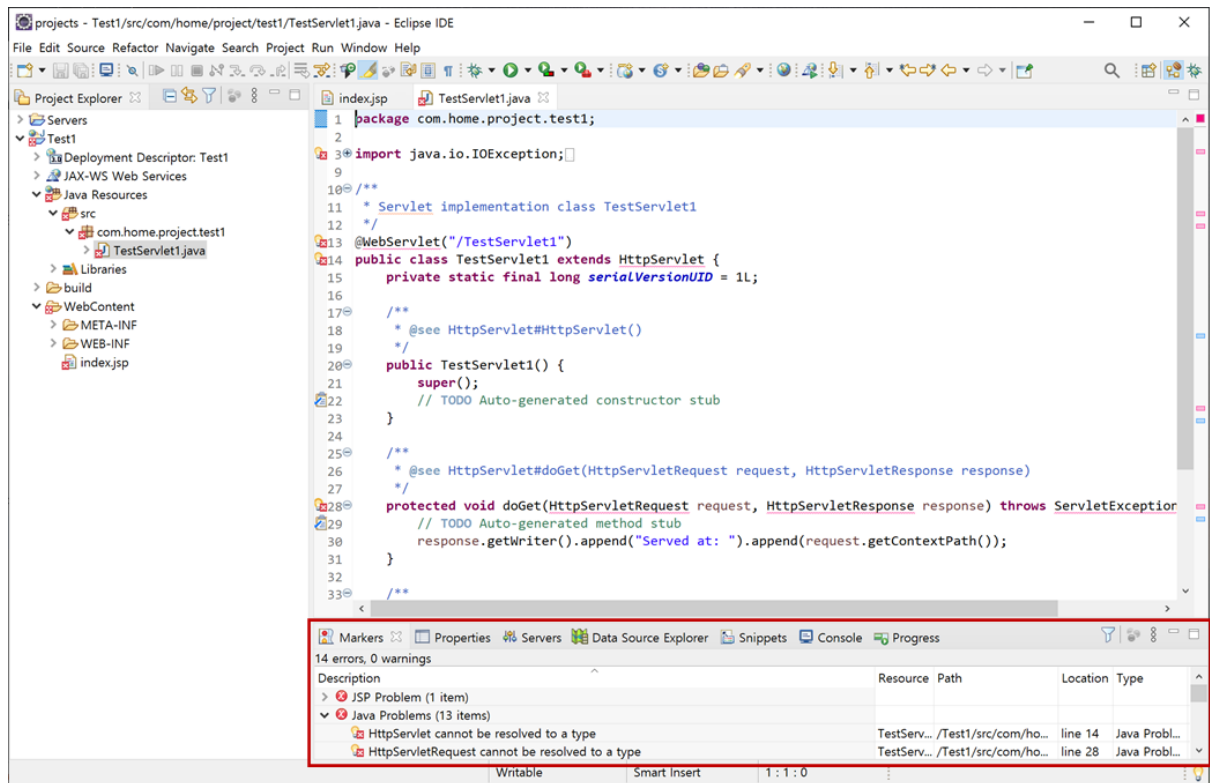
< Back

Next >

Finish

Cancel

5. "Test1" 프로젝트의 "Java Resources"에서 "src/main/java"에 "com.home.project.test1" 패키지가 생성되고 "TestServlet1.java" 파일이 생성됩니다.



그렇지만, 에러가 발생하였습니다.

하단에 "Markers"탭을 클릭하면 에러 메시지를 자세히 볼 수 있습니다.

에러의 원인은 "HttpServlet cannot be resolved to a type, HttpServletRequest cannot be resolved to a type, HttpServletResponse cannot be resolved to a type, ServletException cannot be resolved to a type, The import javax.servlet.annotation cannot be resolved, The import javax.servlet.http cannot be resolved, The import javax.servlet.ServletException cannot be resolved, WebServlet cannot be resolved to a type"으로 "HttpServlet"과 관련한 유형들을 확인할 수 없다는 것입니다.

에러는 JSP파일보다 많지만 슈퍼 클래스인 "javax.servlet.http.HttpServlet"을 찾을 수 없다는 것입니다.

에러를 처리하는 방법으로는 Tomcat 서버 런타임을 설정해서 처리하는 방법(일방적인 방법)과 "javax.servlet" JAR파일을 라이브러리에 등록하여 처리하는 방법이 있습니다.

1. 톰캣 서버 런타임(Tomcat Server Runtime) 설정하기(carrotweb.tistory.com/15)
2. javax.servlet JAR파일 등록하기(carrotweb.tistory.com/16)