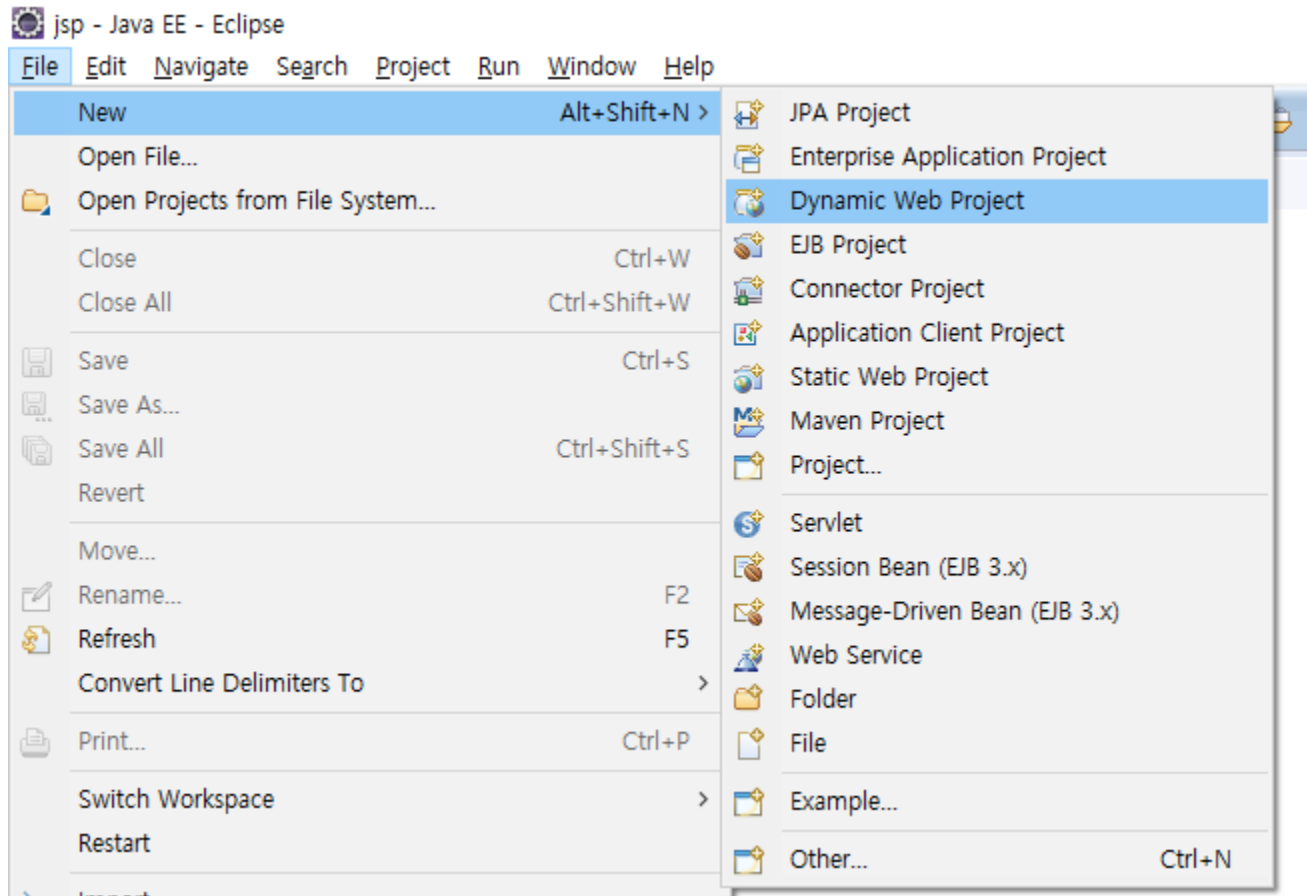


서블릿의 기초

서블릿 프로그램

❖ 프로젝트 생성

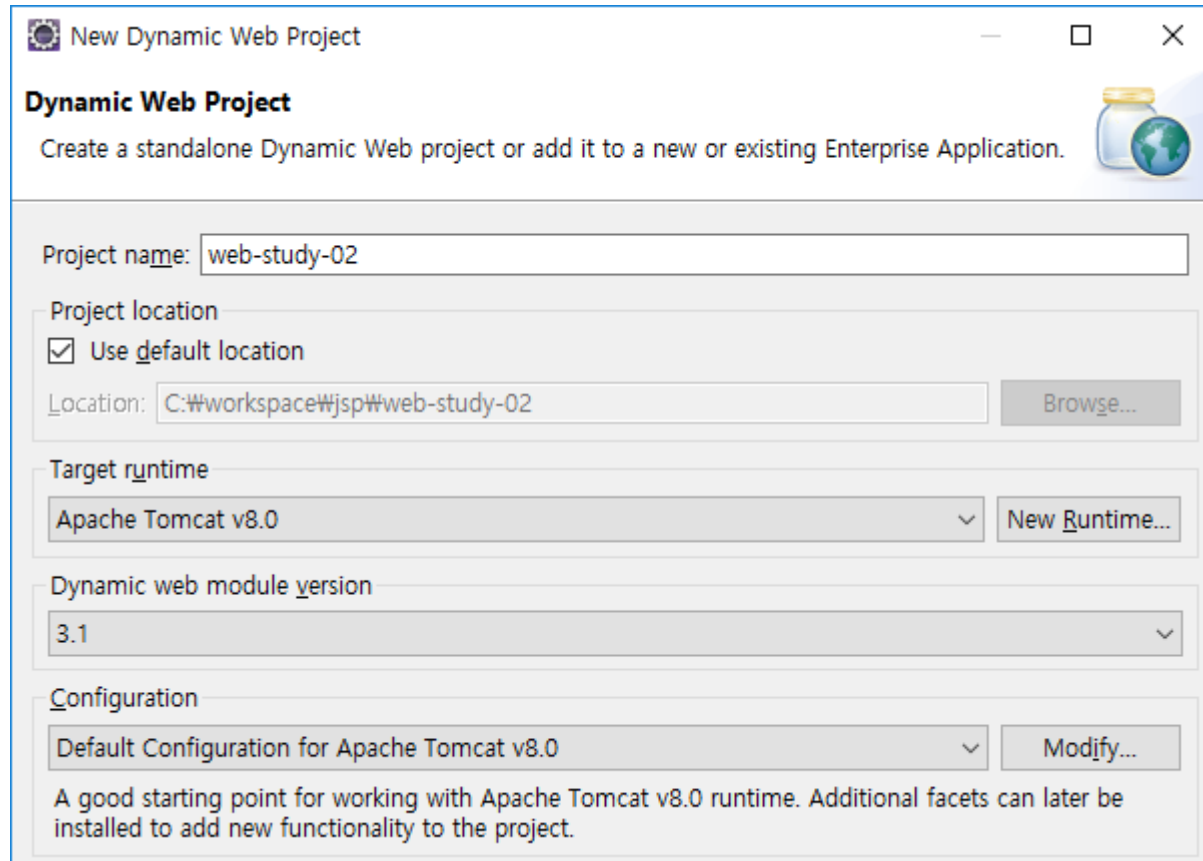
- File → New → Dynamic Web Project



서블릿 프로그램

❖ 프로젝트 생성

- 프로젝트명 : web-study-02



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: web-study-02

Project location
☒ Use default location
Location: C:\workspace\jsp\web-study-02 Browse...

Target runtime
Apache Tomcat v8.0 New Runtime...

Dynamic web module version
3.1

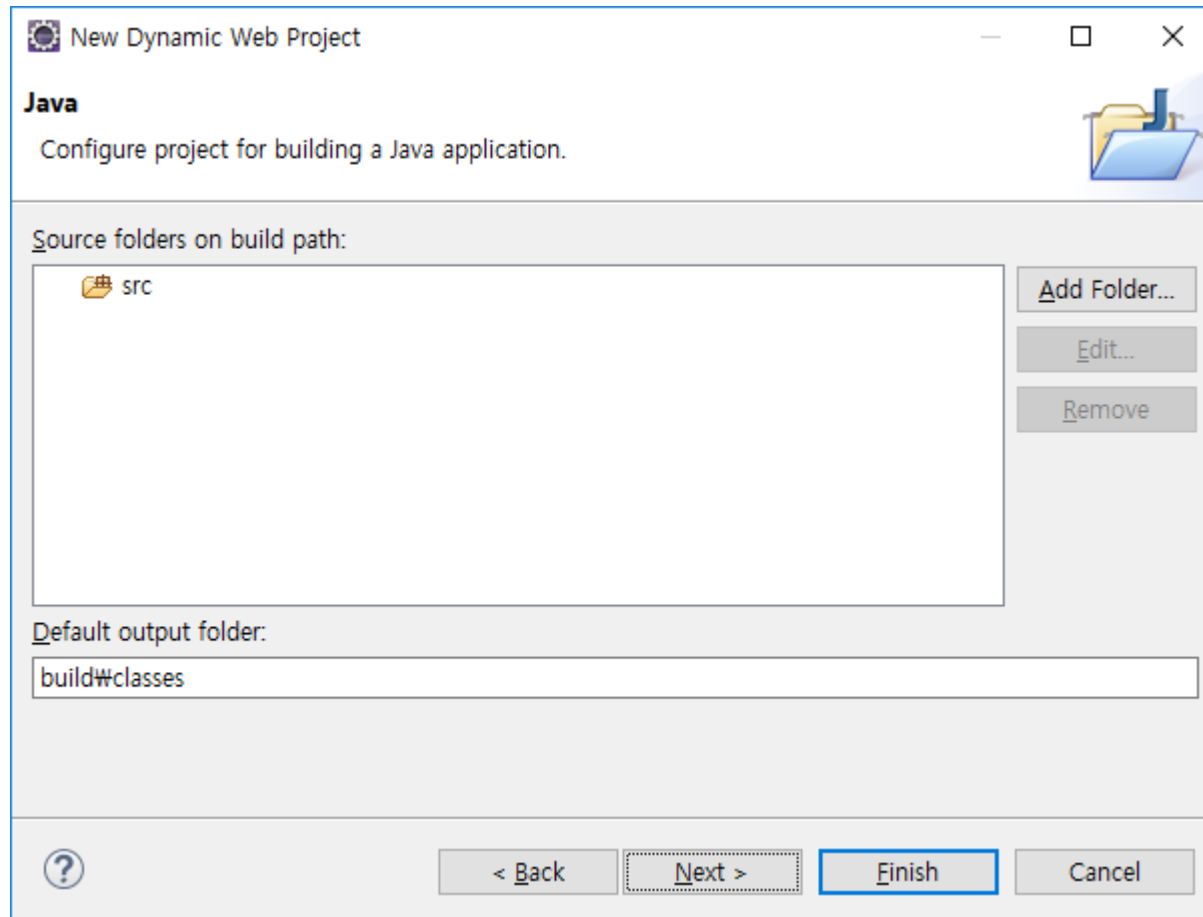
Configuration
Default Configuration for Apache Tomcat v8.0 Modify...

A good starting point for working with Apache Tomcat v8.0 runtime. Additional facets can later be installed to add new functionality to the project.

서블릿 프로그램

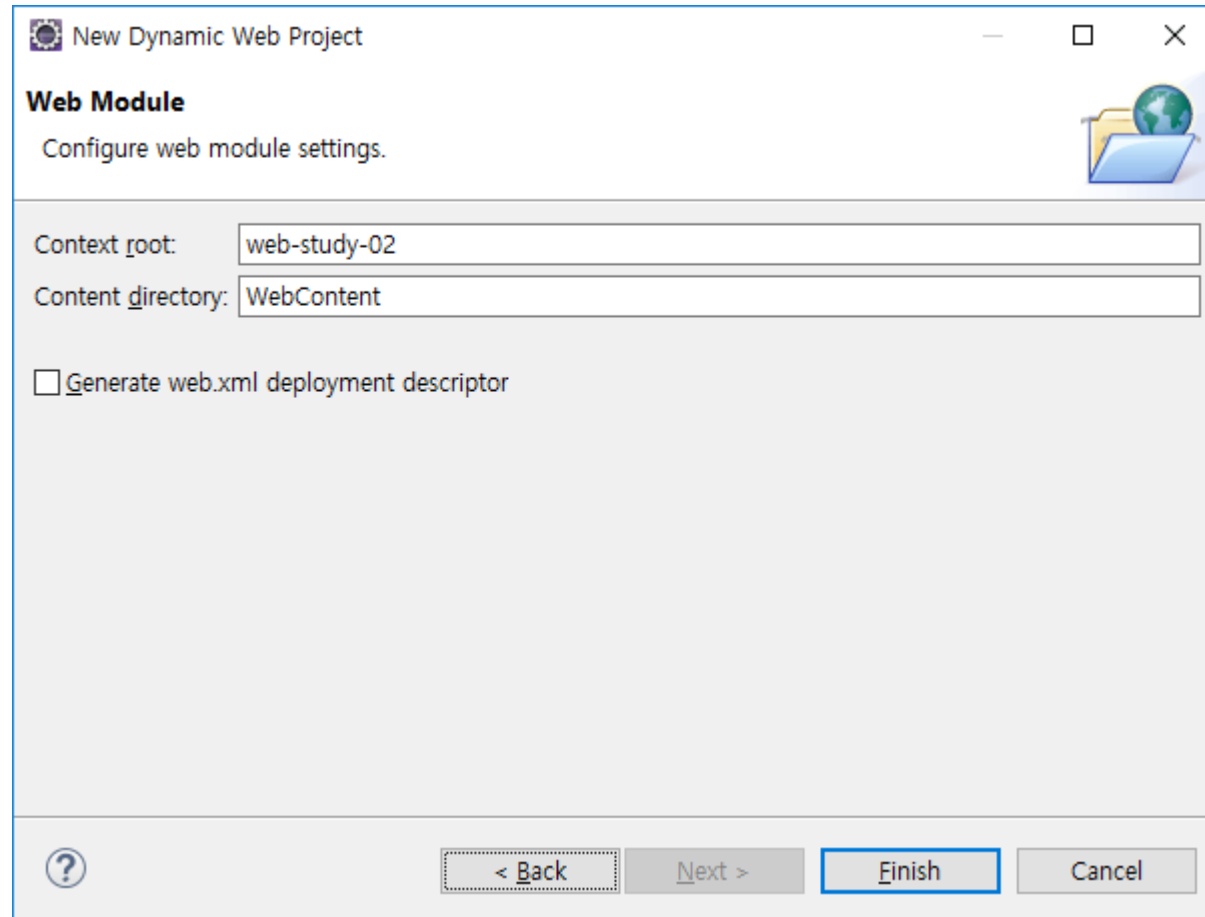
❖ 프로젝트 생성

- 기본 폴더 구조 설정
 - 디폴트 유지



❖ 프로젝트 생성

- 컨텍스트 패스 및 web.xml 파일 생성 여부 설정



New Dynamic Web Project

Web Module
Configure web module settings.

Context root: web-study-02

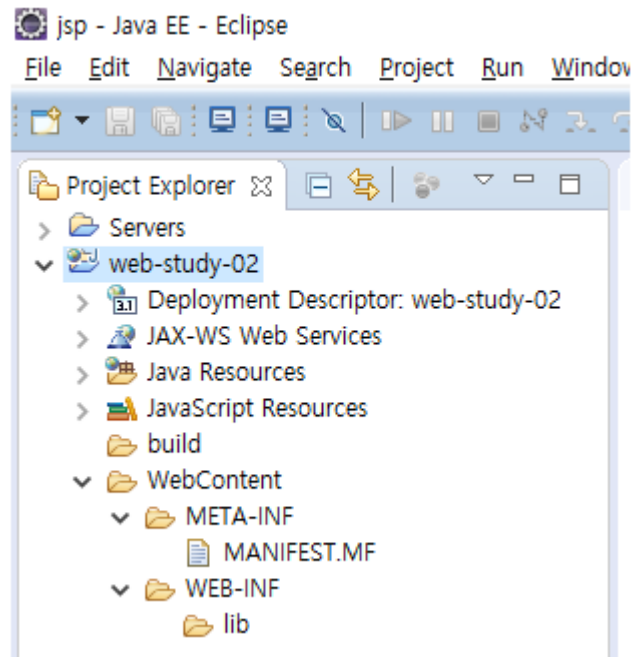
Content directory: WebContent

☐ Generate web.xml deployment descriptor

? < Back Next > Finish Cancel

서블릿 프로그램

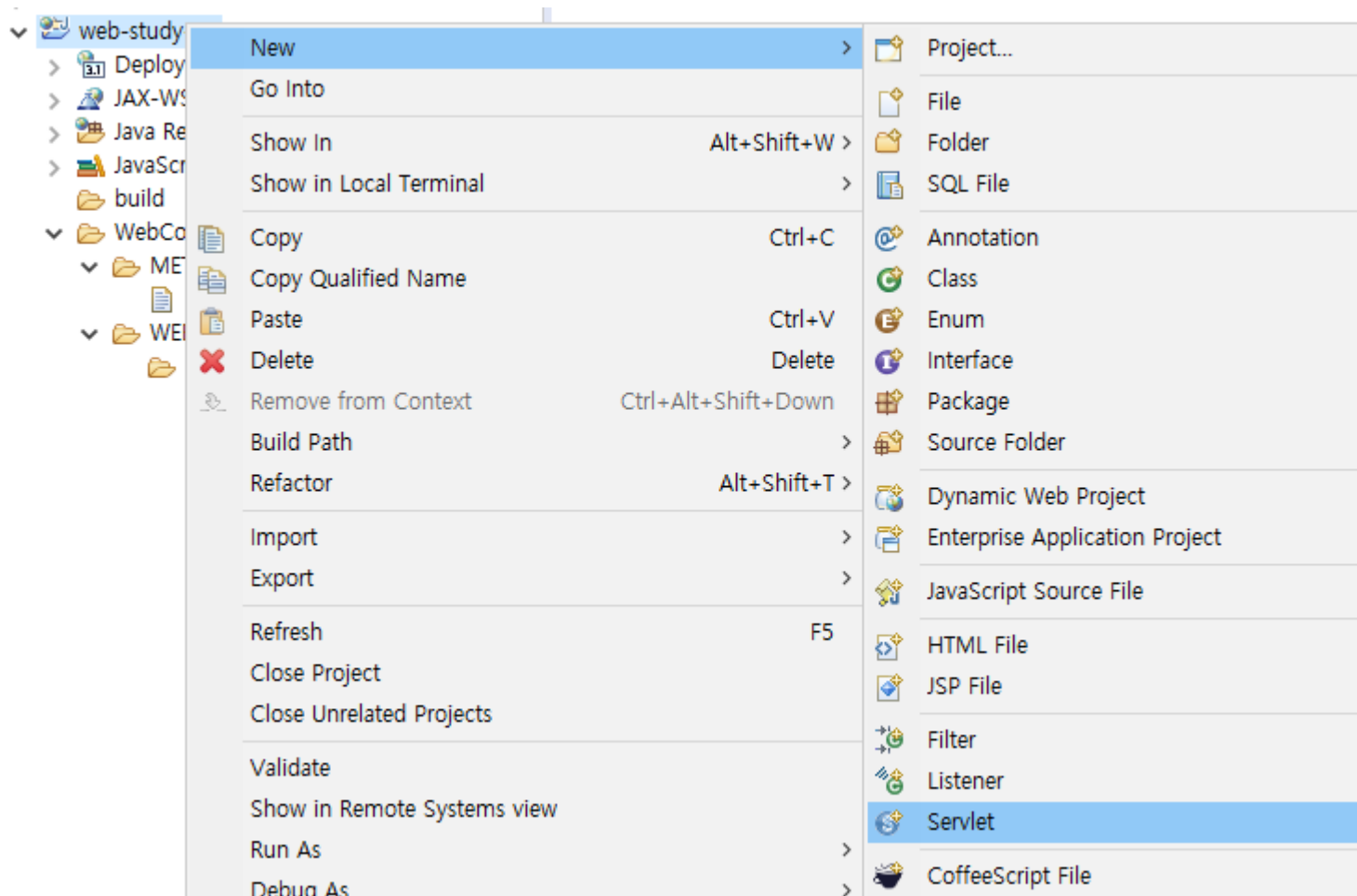
❖ 프로젝트 기본 구조



서블릿 프로그램

❖ 서블릿 생성

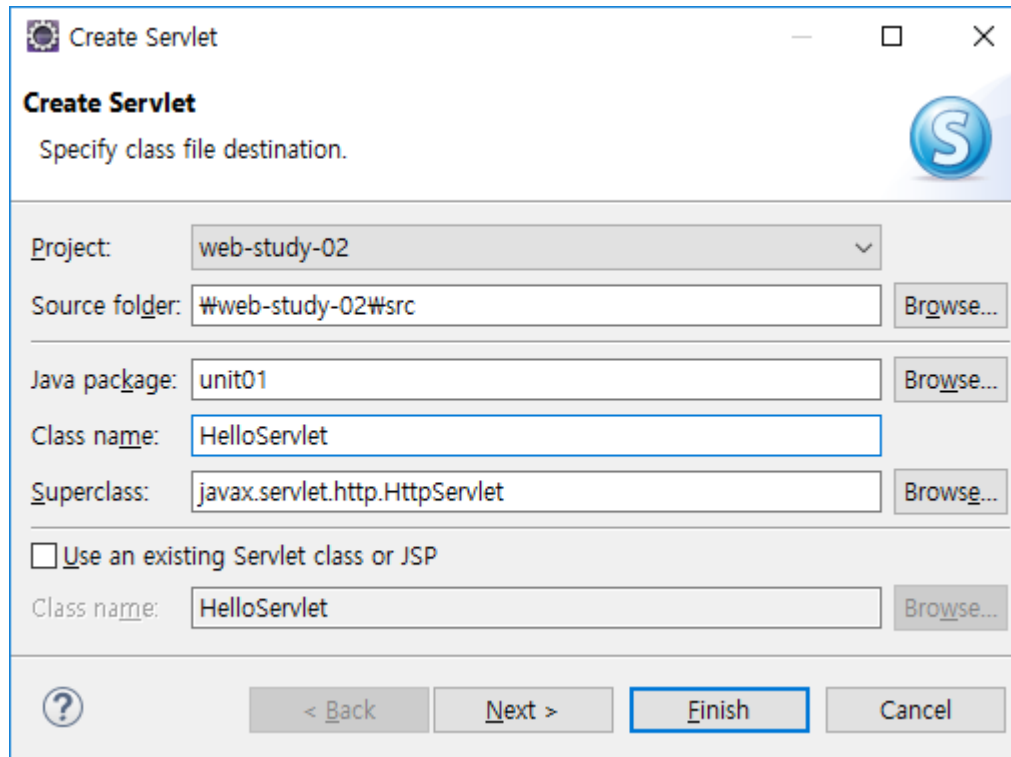
- 프로젝트 선택 → New → Servlet



서블릿 프로그램

❖ 서블릿 생성

- Java package : unit01
- Class name : HelloServlet



The image shows a 'Create Servlet' dialog box from an IDE. It has a title bar with a gear icon and the text 'Create Servlet'. Below the title bar, there's a section titled 'Create Servlet' with a sub-instruction 'Specify class file destination.' and a blue 'S' icon. The dialog contains several input fields and buttons:

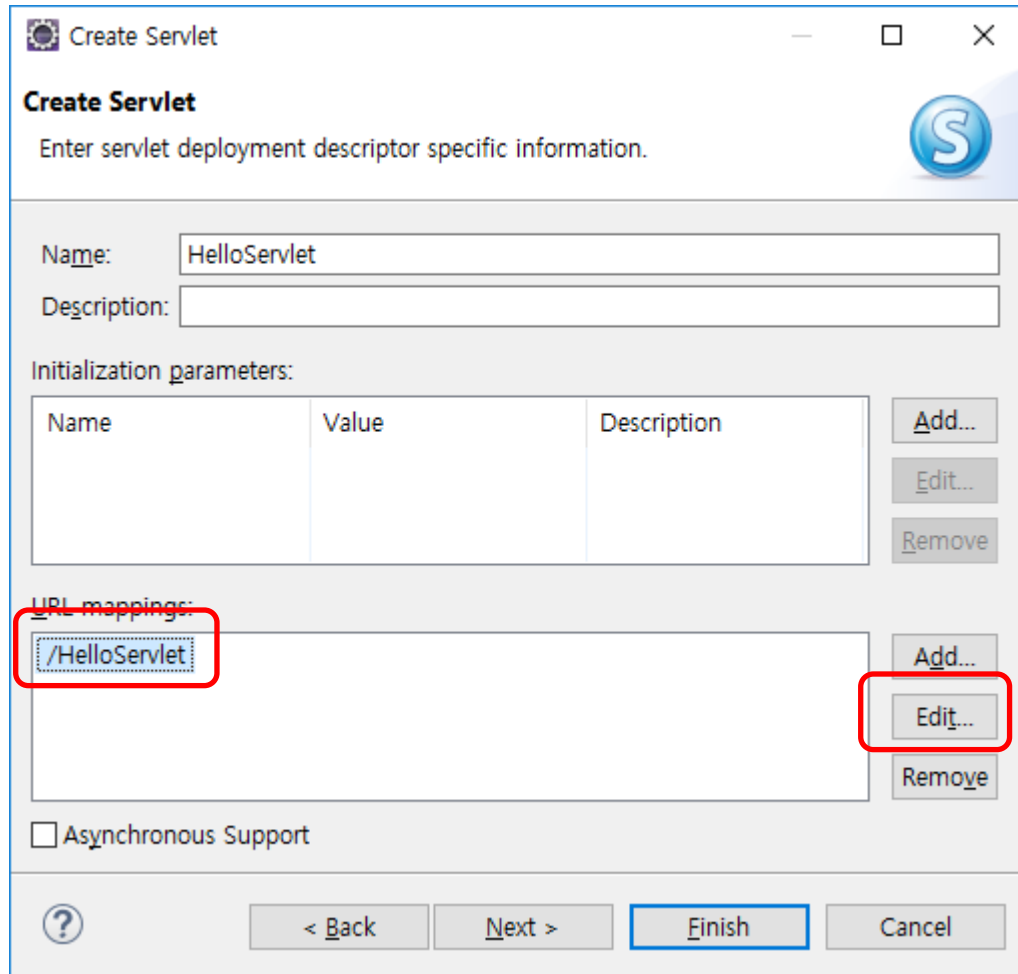
- Project:** A dropdown menu showing 'web-study-02'.
- Source folder:** A text field containing '\\web-study-02\\src' with a 'Browse...' button to its right.
- Java package:** A text field containing 'unit01' with a 'Browse...' button to its right.
- Class name:** A text field containing 'HelloServlet'.
- Superclass:** A text field containing 'javax.servlet.http.HttpServlet' with a 'Browse...' button to its right.
- Use an existing Servlet class or JSP:** An unchecked checkbox.
- Class name:** A text field containing 'HelloServlet' with a 'Browse...' button to its right.

At the bottom, there's a row of buttons: a help icon (?), '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel'.

서블릿 프로그램

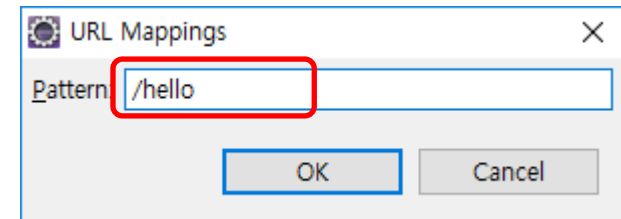
❖ 서블릿 생성

- URL 매핑명 변경 : /HelloServlet → /hello



The 'Create Servlet' dialog box is shown. It has a title bar with a gear icon and the text 'Create Servlet'. Below the title bar, it says 'Create Servlet' and 'Enter servlet deployment descriptor specific information.' There is a blue 'S' icon in the top right corner. The dialog contains several fields and buttons:

- Name:** A text field containing 'HelloServlet'.
- Description:** An empty text field.
- Initialization parameters:** A table with three columns: 'Name', 'Value', and 'Description'. To the right of the table are three buttons: 'Add...', 'Edit...', and 'Remove'.
- URL mappings:** A text field containing '/HelloServlet'. To the right of the field are three buttons: 'Add...', 'Edit...', and 'Remove'. The 'Edit...' button is highlighted with a red box.
- Asynchronous Support:** A checkbox that is currently unchecked.
- Bottom navigation:** A row of buttons: '< Back', 'Next >', 'Finish' (highlighted with a blue box), and 'Cancel'.

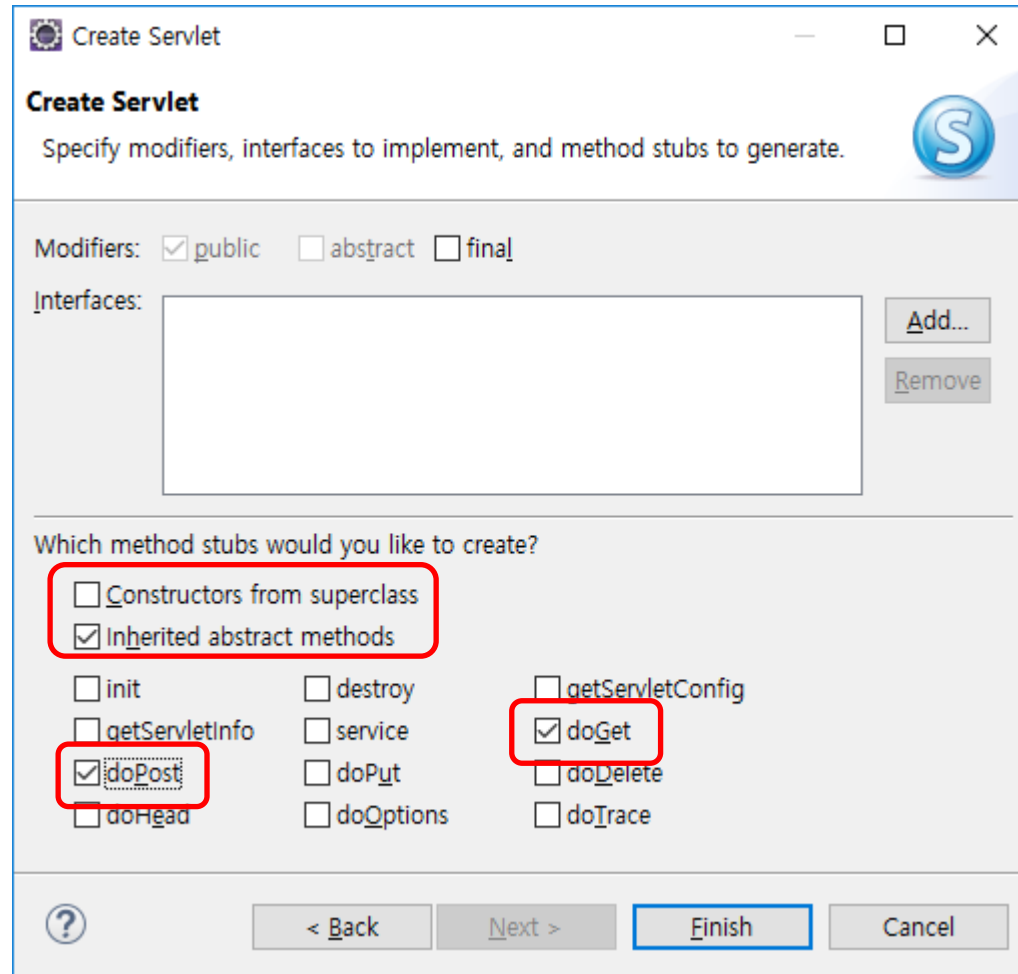


The 'URL Mappings' dialog box is shown. It has a title bar with a gear icon and the text 'URL Mappings'. Below the title bar, it says 'Pattern' followed by a text field containing '/hello'. The text field is highlighted with a red box. To the right of the text field are two buttons: 'OK' and 'Cancel'.

서블릿 프로그램

❖ 서블릿 생성

- 요청 처리 메서드 선택



The image shows the 'Create Servlet' dialog box in an IDE. It has a title bar with a gear icon and the text 'Create Servlet'. Below the title bar, there's a section titled 'Create Servlet' with a subtitle 'Specify modifiers, interfaces to implement, and method stubs to generate.' and a blue 'S' icon. The 'Modifiers' section has three checkboxes: 'public' (checked), 'abstract' (unchecked), and 'final' (unchecked). The 'Interfaces' section has a text box and two buttons: 'Add...' and 'Remove'. The 'Which method stubs would you like to create?' section has a grid of checkboxes. A red box highlights the first two options: 'Constructors from superclass' (unchecked) and 'Inherited abstract methods' (checked). Another red box highlights 'doPost' (checked) in the second column. A third red box highlights 'doGet' (checked) in the third column. At the bottom, there are four buttons: a help button (question mark), '< Back', 'Next >', and 'Finish' (highlighted with a blue border), and a 'Cancel' button.

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

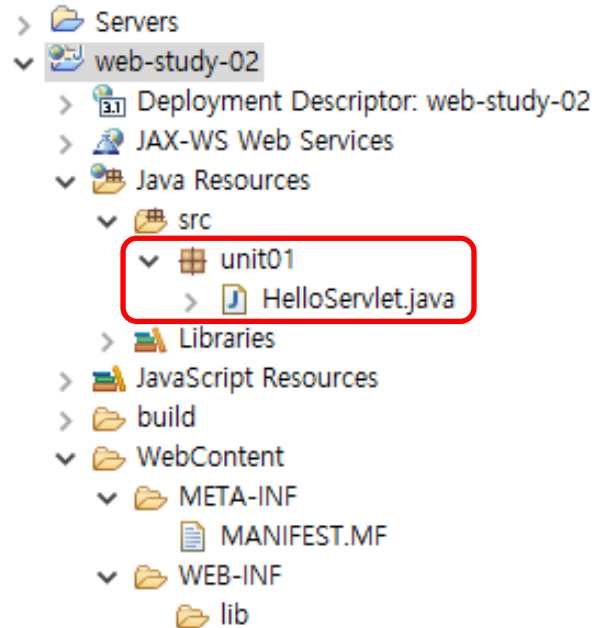
☒ doPost ☐ doPut ☐ doDelete

☐ doHead ☐ doOptions ☐ doTrace

? < Back Next > Finish Cancel

서블릿 프로그램

❖ 서블릿 생성



서블릿 프로그램

❖ 디폴트 서블릿 코드

```
package unit01;

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("/hello")
public class HelloServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
```

서블릿 프로그램

❖ 디폴트 서블릿 코드

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

    // TODO Auto-generated method stub
    response.getWriter()
        .append("Served at: ")
        .append(request.getContextPath());
}

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

    // TODO Auto-generated method stub
    doGet(request, response);
}

}
```

서블릿 프로그램

❖ 서블릿 코드 수정

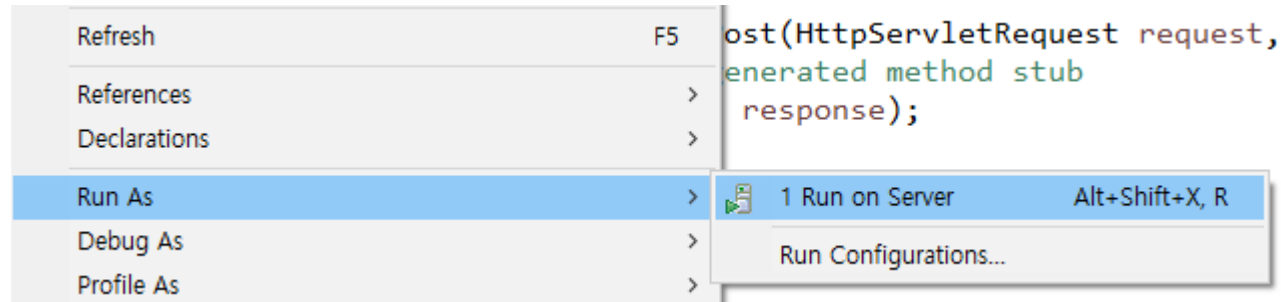
```
:
protected void doGet(HttpServletRequest request,
                      HttpServletResponse response)
                      throws ServletException, IOException {

    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.print("<html><body><h1>");
    out.print("Hello Servlet");
    out.print("</h1></body></html>");
    out.close();
}
:
```

서블릿 프로그램

❖ 실행

- 서블릿 클래스 선택 → Run As → Run On Server (단축키 : Ctrl+F11)



서블릿 프로그램

❖ 실행

○ 서버 선택

Run On Server

Run On Server

Select which server to use

How do you want to select the server?

☒ Choose an existing server

☐ Manually define a new server

Select the server that you want to use:

type filter text

Server	State
▼ localhost	
Tomcat v8.0 Server at localhost	Stopped

Apache Tomcat v8.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5, 6, and 7 Web modules.

☐ Always use this server when running this project

?

< Back

Next >

Finish

Cancel

Run On Server

Add and Remove

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

Available:

Configured:

Add >

< Remove

Add All >>

<< Remove All

web-study-02

?

< Back

Next >

Finish

Cancel

서블릿 프로그램

❖ 실행



○ url

- http://localhost:포트번호/web-study-02/hello

컨텍스트 패스

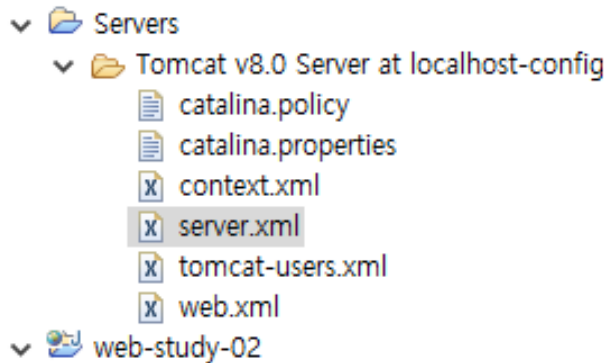
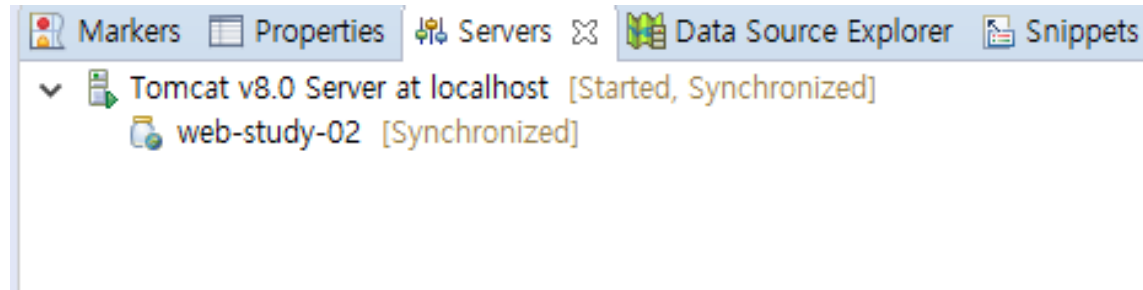
서블릿 요청 URL 패턴

○ 컨텍스트 패스(Context Path)

- 한 웹 서버에서 웹 애플리케이션을 구분

서블릿 프로그램

❖ 서버 환경 변화

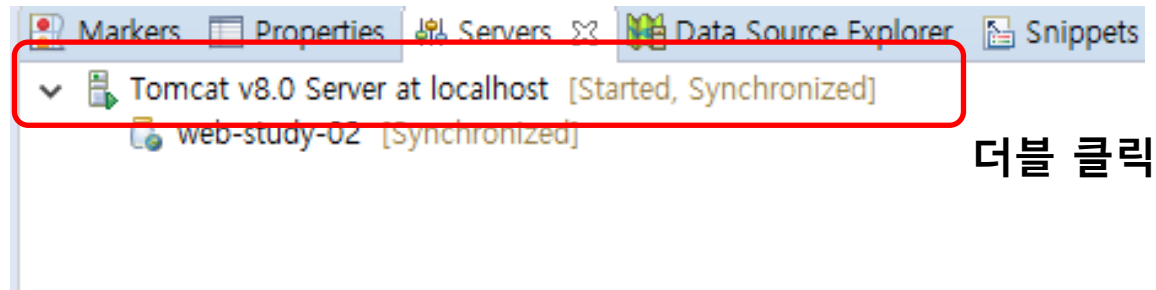


```
<?xml version="1.0" encoding="UTF-8"?>
:
```

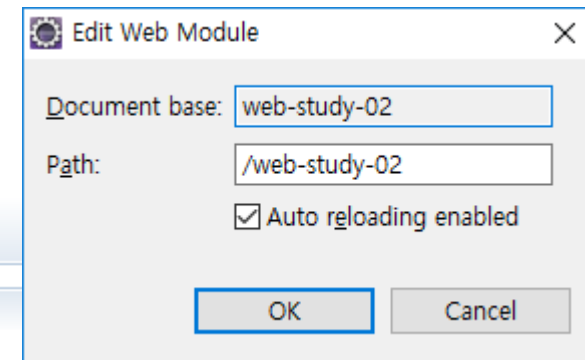
```
<Context docBase="web-study-02"
  path="/web-study-02"
  reloadable="true"
  source="org.eclipse.jst.jee.server:web-study-02"/>
</Host>
</Engine>
</Service>
</Server>
```

서블릿 프로그램

❖ 컨텍스트 패스 변경



더블 클릭



Web Modules

Web Modules

Configure the Web Modules on this server.

Path	Document Base
/web-study-02	web-study-02

Add Web Module...

Add External Web Module...

Edit...

Remove

Overview Modules

서블릿 프로그램

❖ @WebServlet 어노테이션

- URL 매핑 패턴 지정

```
package unit01;

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("/hello")
public class HelloServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
```

서블릿 프로그램

❖ 서블릿 클래스 구조

접근 제한자는 반드시
public이어야 함

```
public class HttpServlet extends HttpServlet {  
    }  
}
```

서블릿 클래스 이름

HttpServlet의 상속을 받아야 함

서블릿 프로그램

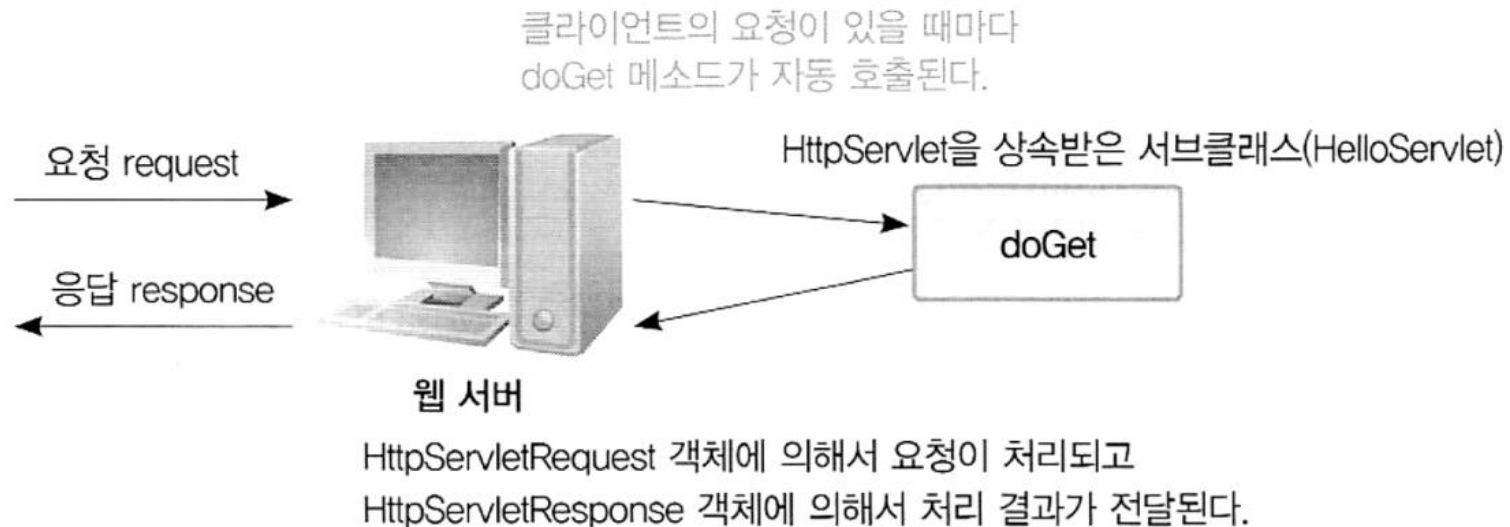
❖ 요청 처리 메서드

- doGet
 - HTTP의 GET 요청 처리시 호출
- doPost
 - HTTP의 POST 요청 처리시 호출
- 매개변수
 - HttpServletRequest request : 요청 정보 처리
 - HttpServletResponse response : 응답 처리

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

서블릿 프로그램

❖ 서블릿의 HTTP 요청 처리 과정



서블릿 프로그램

❖ 응답 보내기

- 응답 내용 형태 지정(MIME 타입 지정)
 - `response.setContentType()` 메서드로 지정

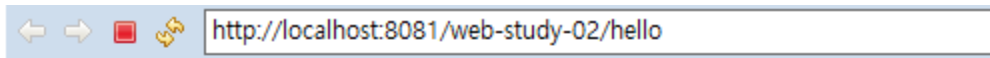
```
:
protected void doGet(HttpServletRequest request,
                      HttpServletResponse response)
                      throws ServletException, IOException {

    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.print("<html><body><h1>");
    out.print("안녕 Servlet");
    out.print("</h1></body></html>");
    out.close();
}
:
```


서블릿 프로그램

❖ 응답 보내기

- 응답 내용 형태 지정(MIME 타입 지정)
 - 한글 응답시 문자셋 문제 발생
 - 응답 문자셋은 UTF-8이지만 브라우저는 이를 모름
 - 디폴트 EUC-KR로 처리
 - 콘텐츠 타입에 사용 문자셋을 지정



?? Servlet

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

    response.setContentType("text/html; charset=UTF-8");
    :
}
:
}
```

서블릿 프로그램

❖ 응답 보내기

- 출력 스트림을 얻어 출력

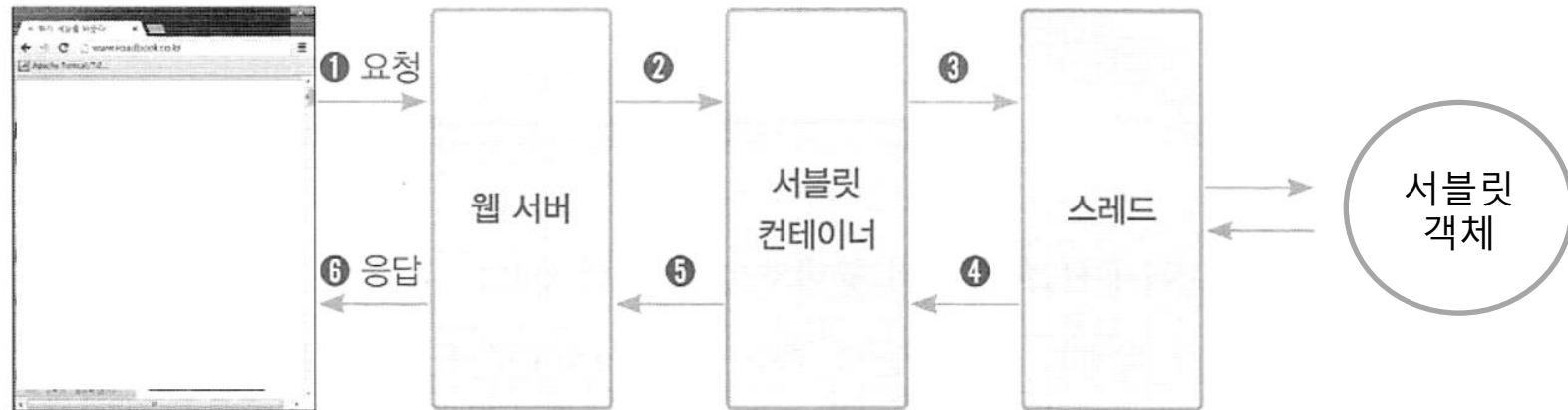
```
PrintWriter out = response.getWriter();
```

- 출력하기
 - response.setContentType("text/html;charset=UTF-8")으로 출력 형태를 html로 지정했으므로 html 코드를 작성하여 출력

```
out.print("<html><body><h1>");  
out.print("안녕 Servlet");  
out.print("</h1></body></html>");  
out.close();
```

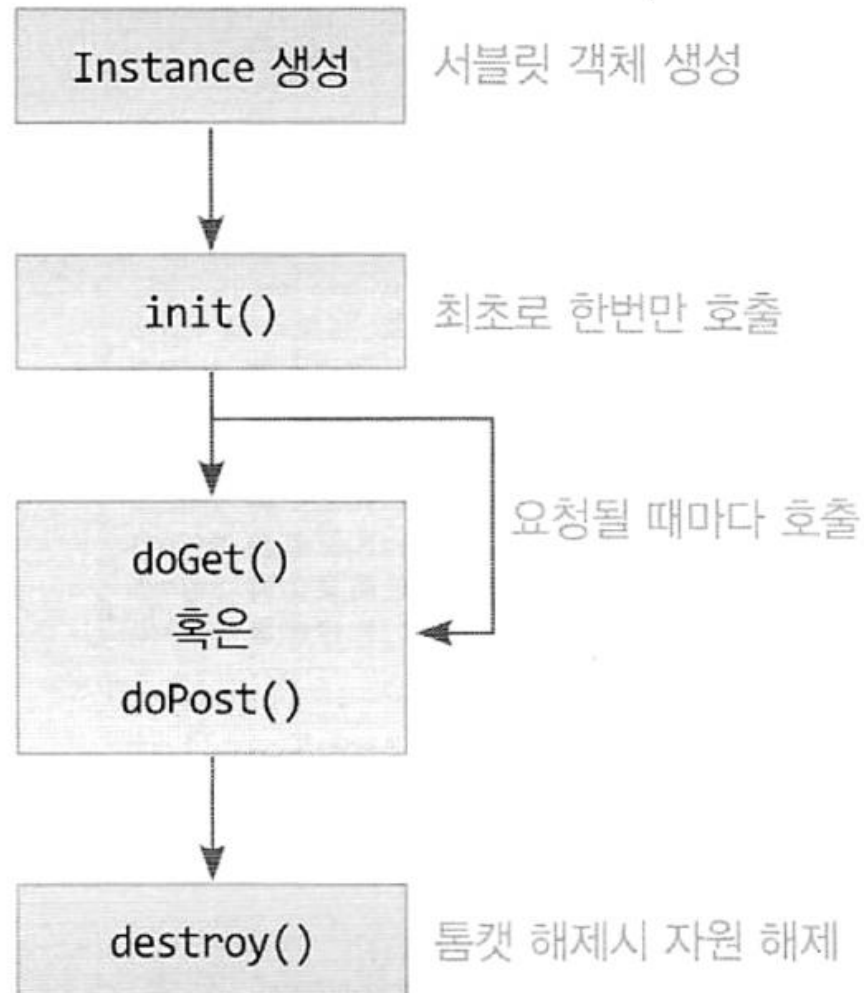
서블릿 프로그램

❖ 서블릿 작동 절차



서블릿 프로그램

❖ 서블릿 라이프 사이클



서블릿 프로그램

❖ 서블릿 라이프 사이클

- 서블릿 클래스 추가
 - 클래스명 : Lifecycle
 - URL 패턴 : /Lifecycle
 - 메서드 선택
 - init, destroy, doGet, doPost

```
@WebServlet("/Lifecycle")
public class Lifecycle extends HttpServlet {
    private static final long serialVersionUID = 1L;

    int initCount =1;
    int doGetCount = 1;
    int destroyCount =1;

    public Lifecycle() {
        super();
    }
}
```

서블릿 프로그램

```
public void init(ServletConfig config) throws ServletException {  
    System.out.println("init 메서드는 첫 요청만 호출됨 : " + initCount++);  
}
```

```
public void destroy() {  
    System.out.println("destroy 메서드는 톰캣이 종료될 때만 호출됨 : " +  
        destroyCount++);  
}
```

```
protected void doGet(HttpServletRequest request,  
    HttpServletResponse response)  
    throws ServletException, IOException {  
  
    System.out.println("doGet 메서드가 요청될 때마다 호출됨 : " +  
        doGetCount++);  
}
```

```
:  
}
```