

2024년 상반기 K-디지털 트레이닝

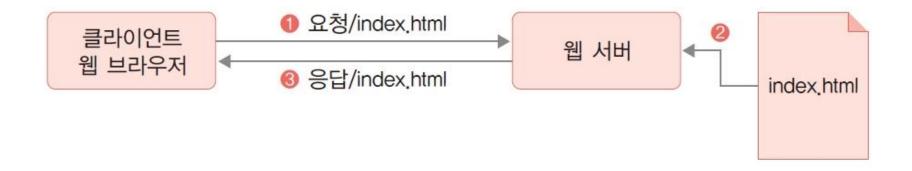
서블릿의 이해

[KB] IT's Your Life



서블릿(Servlet) 개요

☑ 기본적인 웹의 요청과 응답 과정



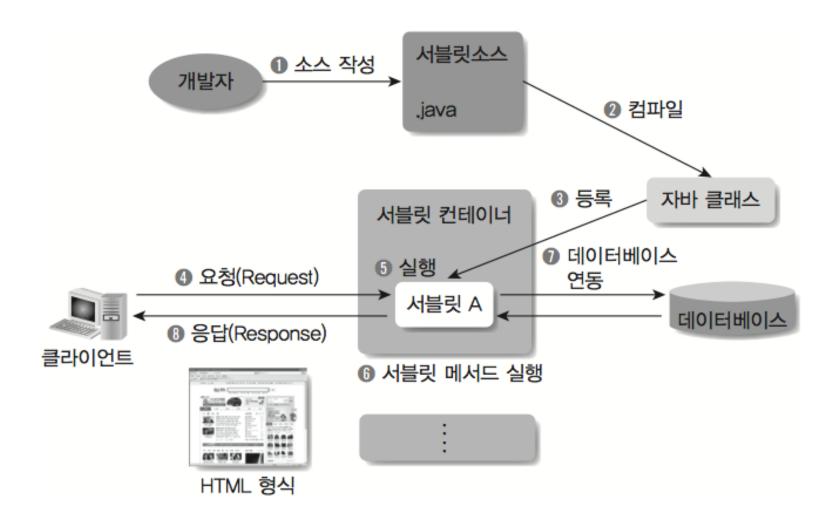
서블릿(Servlet) 개요

☑ 서블릿

- 웹 컨테이너에 의해 관리
- 다양한 클라이언트 요청에 의해 컨텐츠로 응답 가능한 자바 기반의 웹 컴포넌트



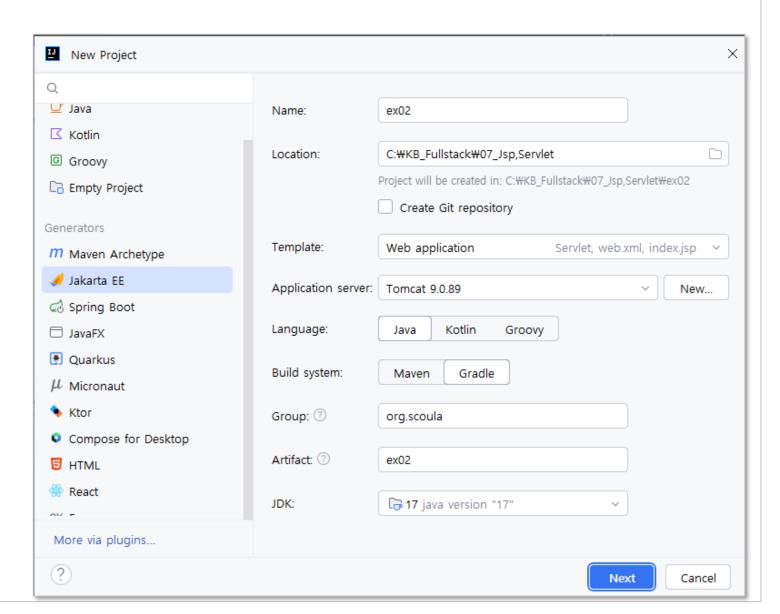
서블릿(Servlet) 개요



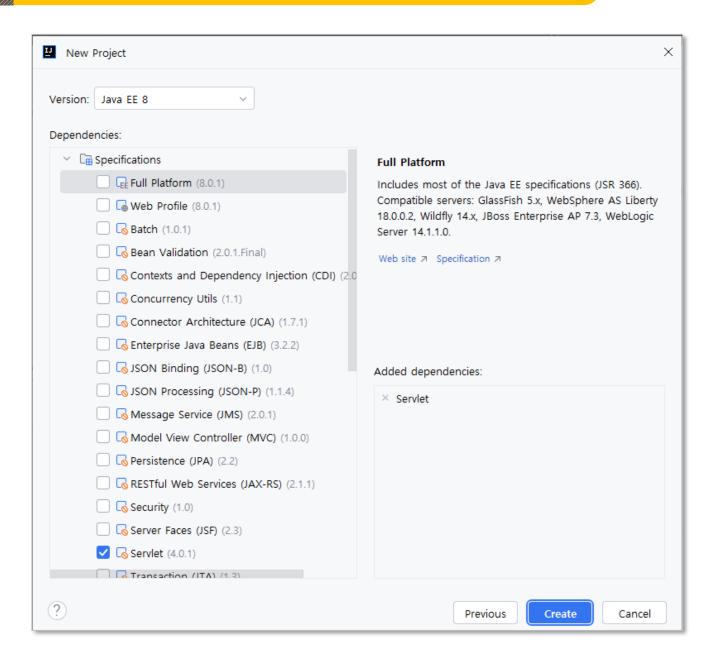
프로젝트 만들기

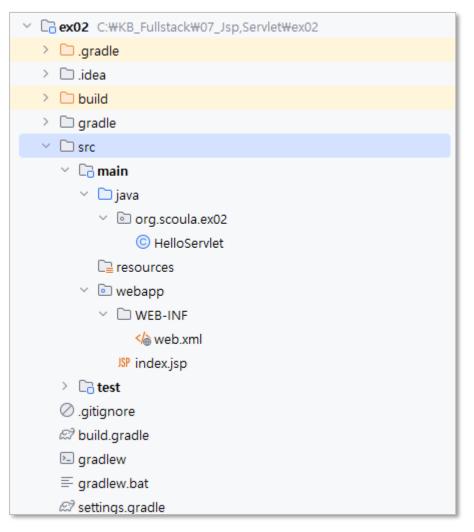
☑ 프로젝트 생성

- o Templates: Jakarta EE
- o Project name: ex02



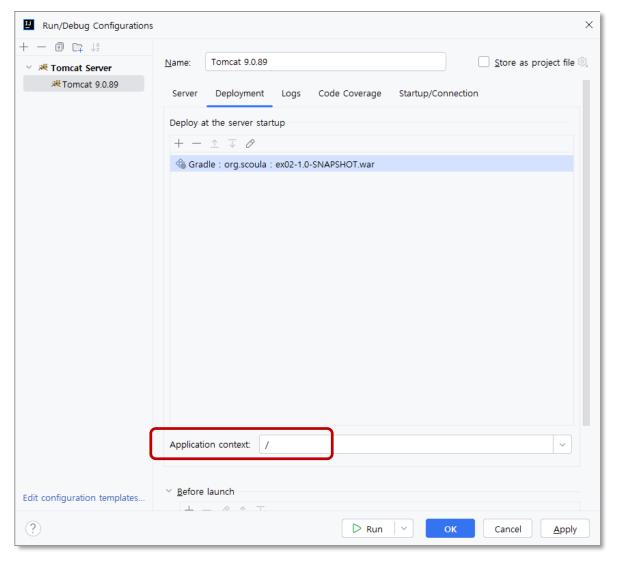
프로젝트 만들기

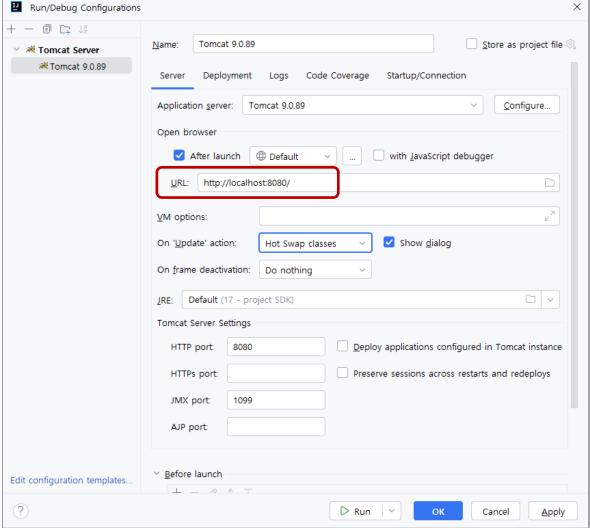




프로젝트 만들기

Tomcat 설정





index.jsp

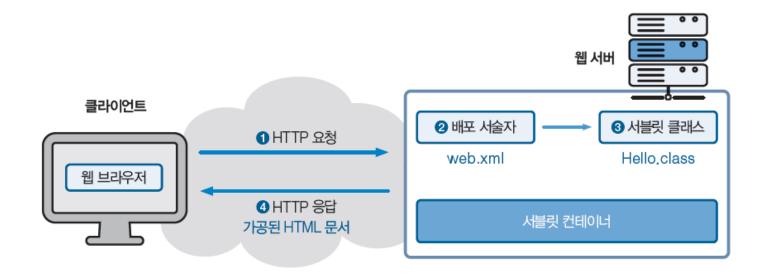
http://localhost:8080/

Hello World!

Hello Servlet

서블릿 맵핑

🗸 서블릿 동작 과정



HelloServlet.java

```
package org.scoula.ex02;
import java.io.*;
import javax.servlet.http.*;
import javax.servlet.annotation.*;
// @WebServlet(name = "helloServlet", value = "/hello-servlet")
public class HelloServlet extends HttpServlet {
    private String message;
    public void init() {
        message = "Hello World!";
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html><body>");
        out.println("<h1>" + message + "</h1>");
        out.println("</body></html>");
    public void destroy() {
```

서블릿 맵핑


```
《servlet-name》서블릿별명《/servlet-name》
《servlet-class》패키지를 포함한 서블릿명《/servlet-class》
《/servlet》
《servlet-mapping》
《servlet-name》서블릿별명《/servlet-name》
《url-pattern》/맵핑명《/url-pattern》
《/servlet-mapping》
```

```
package org scoula.ex02;
...
// @WebServlet(name = "helloServlet", value = "/hello-servlet")
public class HelloServlet extends HttpServlet {
    ...
}
```

서블릿 맵핑

web.xml

o web.xml 수정본을 반영하려면 웹서버 재기동

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</pre>
         xmlns="http://xmlns.jcp.org/xml/ns/javaee"
         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
         id="WebApp_ID" version="3.1">
    <servlet>
        <servlet-name>helloServlet</servlet-name>
        <servlet-class>org.scoula.ex02.HelloServlet</servlet-class>
    </servlet>
                                                                              클라이언트
    <servlet-mapping>
                                                                                                                           3 서블릿 클래스
                                                                                                            ② 배포 서술자
        <servlet-name>helloServlet</servlet-name>
                                                                                            ① HTTP 요청
                                                                                                                            Hello, class
                                                                                                             web.xml
        <url-pattern>/hello-servlet</url-pattern>
                                                                             웹 브라우저
    </servlet-mapping>
                                                                                            4 HTTP 응답
                                                                                                                   서블릿 컨테이너
                                                                                           가공된 HTML 문서
</web-app>
```

http://localhost:8080/

Hello World!

Hello Servlet

http://localhost:8080/hello-servlet

Hello World!

@WebServlet

☑ @WebServlet 어노테이션 이용 방법

- o Xml 파일 대신 자바 코드에 기술
- o Servlet 3.0의 어노테이션

• @WebServlet	• @Resources
• @WebFilter	• @PersistenceContext
• @WebInitParam	• @PersistenceContexts
• @WebListener	• @PersistenceUnit
• @MultipartConfig	• @PersistenceUnits
• @DeclareRoles	• @PostConstruct
• @EJB	• @PreDestroy
• @EJBs	• @RunAs
• @Resource	

@WebServlet

💟 서블릿 맵핑 설정

```
@WebServlet("/맵핑명")
public class MyServlet extends HttpServlet { ... }
```

😕 서블릿 별명과 urlPatterns 속성으로 여러 개의 맵핑명 지정

HelloServlet.java 어노테이션을 이용한 서블릿 맵핑 등록

```
OwebServlet(name="MyServlet", urlPatterns={"/xxx", "/yyy" })

public class HelloServlet extends HttpServlet {

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

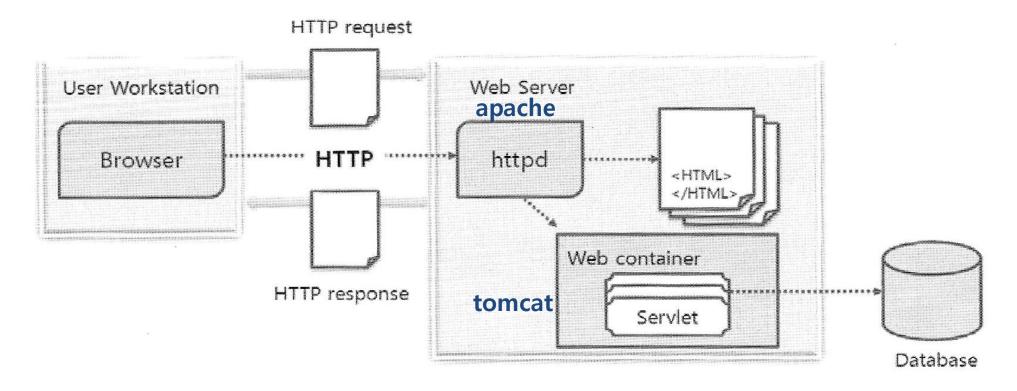
    System.out.println("HelloServlet 요청");
    PrintWriter out = resp.getWriter();
    out.println("<h1>Hello Servlet</h1>");

  }
}
```

- o http://localhost/xxx
- o http://localhost/yyy

서블릿 아키텍처 및 핵심 API

서블릿을 사용하는 경우 웹 아키텍처



서블릿 아키텍처 및 핵심 API

♡ 웹브라우저의 URL 요청 → 해당 서블릿 호출

- o 웹 컨테이너에서 서블릿을 실행
- 결과값을 HTML로 구성하여 클라이언트에 응답

- o HTTP Request(요청)
 - javax.servlet.http.HttpServletRequest
- o HTTP Response(응답)
 - javax.servlet.http.HttpServletResponse

서블릿 아키텍처 및 핵심 AP

HttpServletRequest API

- getHeader(String name):String
- getHeaderNames():Enumeration
- getCookies():Cookie[]
- getRequestURI():String
- getServletPath():String
- getSession(boolean):HttpSession
- getSession():HttpSession

- setCharacterEncoding(String encoding)
- setAttriute(String name, Object obj)
- getAttribute(String name):Object
- removeAttribute(String name)
- getParameter(String name)
- getParameterNames():Enumeration
- getParameterValues(String name):String[]

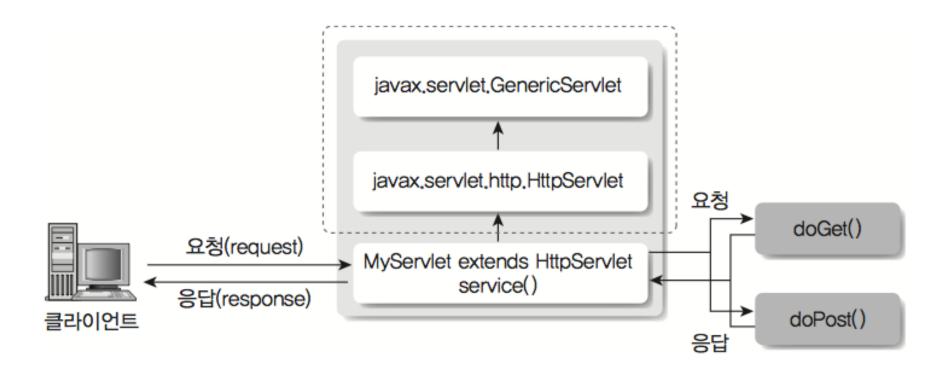
서블릿 아키텍처 및 핵심 AP

HttpServletResponse

- addCookie(Cookie c)
- addHeader(String name, String value)
- encodeURL(String url)
- getStatus()

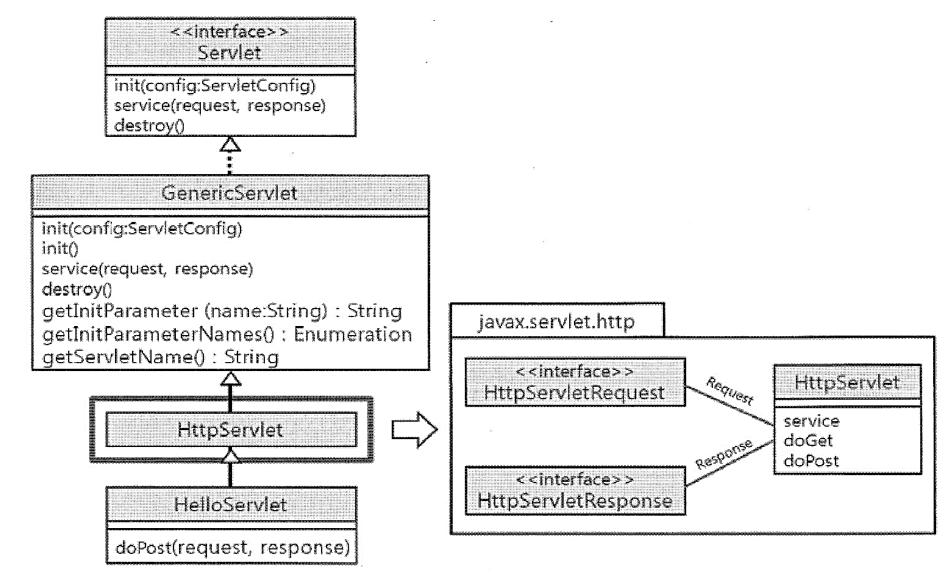
- sendRedirect(String loc)
- getWriter():PrintWriter
- getOutputStream():ServletOutputStream
- setContentType(String type)

▽ javax.servlet.http.HttpServlet을 상속받은 서블릿 동작 구조



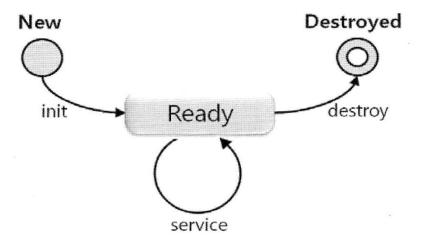
서블릿 아키텍처 및 핵심 API

HttpServlet API



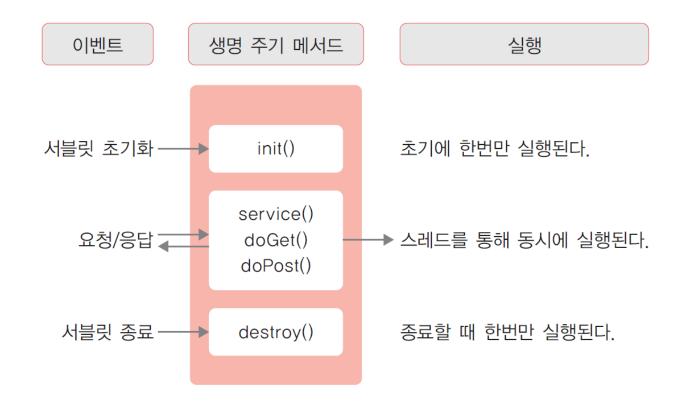
◎ 서블릿의 인스턴스를 init, service, destroy 메서드로 관리

- o init 메서드
 - 인스턴스가 처음 실행될 때, 단 한번 호출
 - 서블릿에서 필요한 초기화 작업 시 사용
- o service 메서드
 - 클라이언트가 요청할 때마다 호출
 - doGet 또는 doPost에서 주로 작업
- o destroy 메서드
 - 인스턴스가 웹 컨테이너에서 제거될 때 호출

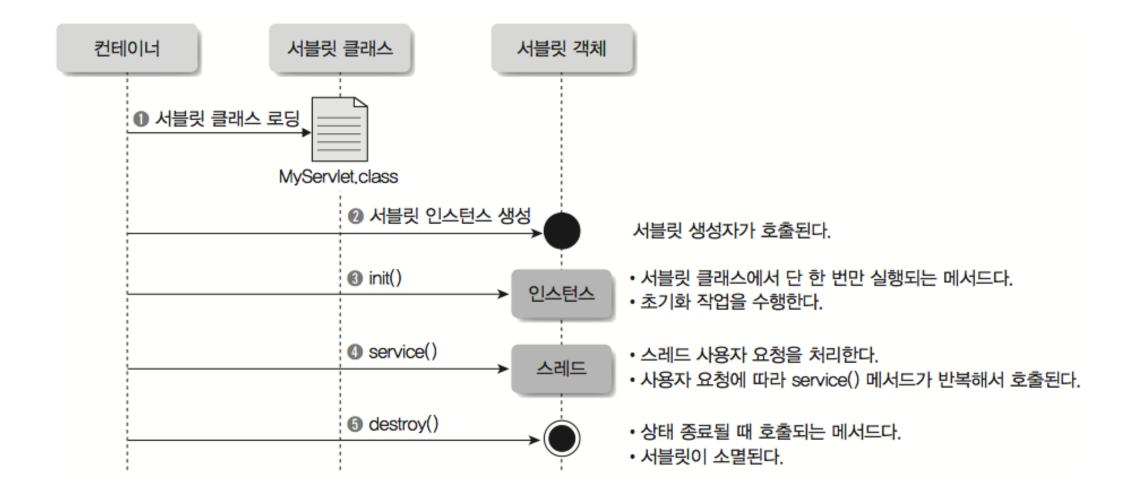


☑ 서블릿의 생명 주기

○ 객체의 생성에서 종료에 이르는 과정

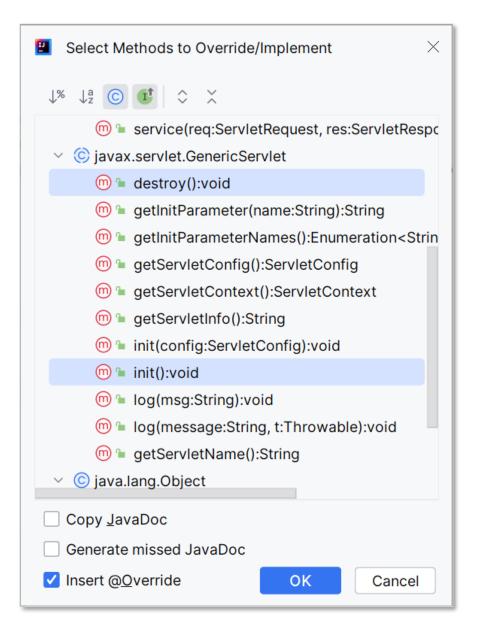


💟 서블릿의 동작 과정



🧿 init, destroy 메서드 추가

- o HelloServlet.java에서 Ctrl+O
- o init(): void, destroy(): void 선택



HelloServlet

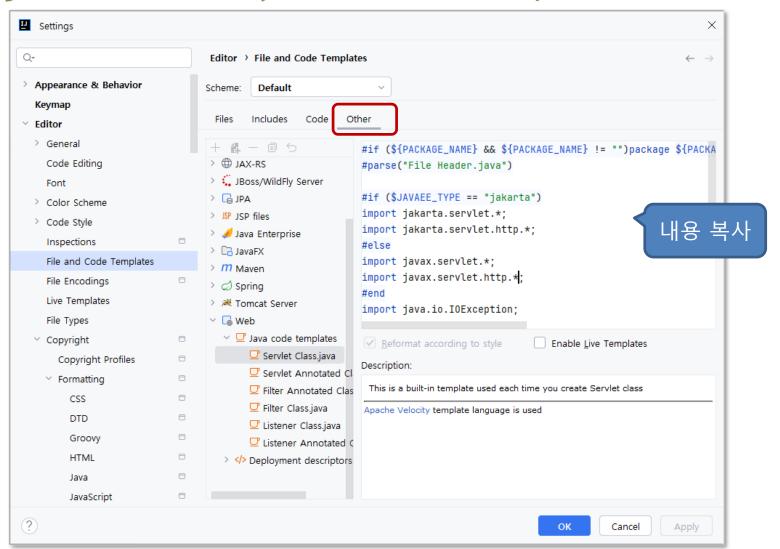
init, destroy 메서드 추가

```
@WebServlet(name="MyServlet", urlPatterns={"/xxx", "/yyy" })
public class HelloServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
            throws ServletException, IOException {
    @Override
    public void destroy() {
        System.out.println("destroy 호출");
    @Override
    public void init() throws ServletException {
        System.out.println("init호출");
```

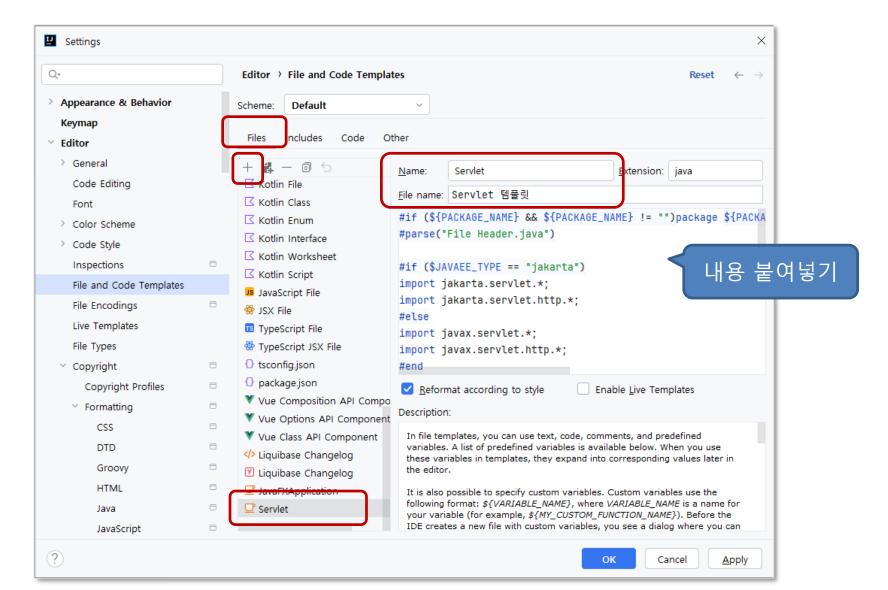
- 💟 클라이언트에서 서블릿으로 요청
- 서블릿은 처리한 결과를 html 형식으로 응답 처리
- ♡ 문자셋 설정
 - o response.setContentType("text/html;charset=UTF-8");
 - 응답 데이터의 MIME 타입
- ◎ 응답 데이터의 전송
 - 문자 데이터 응답
 - response.getWriter()로 PrintWriter 클래스 사용
 - o 바이너리 데이터 응답
 - response.getOutputStream()으로 ServletOutputStream 클래스 사용

😕 Servlet 템플릿 만들기

File > Settings... > Editor > Code Style > File and Code Templates



Servlet 템플릿 만들기



✓ Servlet 템플릿 수정

```
#if (${PACKAGE_NAME} && ${PACKAGE_NAME} != "")package ${PACKAGE_NAME};#end
#parse("File Header.java")

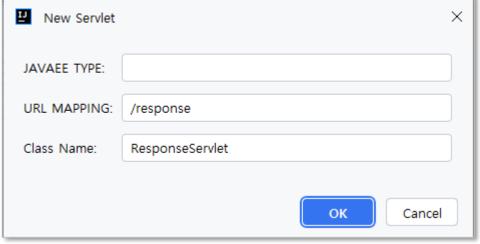
#if ($JAVAEE_TYPE == "jakarta")
import jakarta.servlet.*;
import jakarta.servlet.http.*;
import jakarta.servlet.annotation.WebServlet;
#else
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;

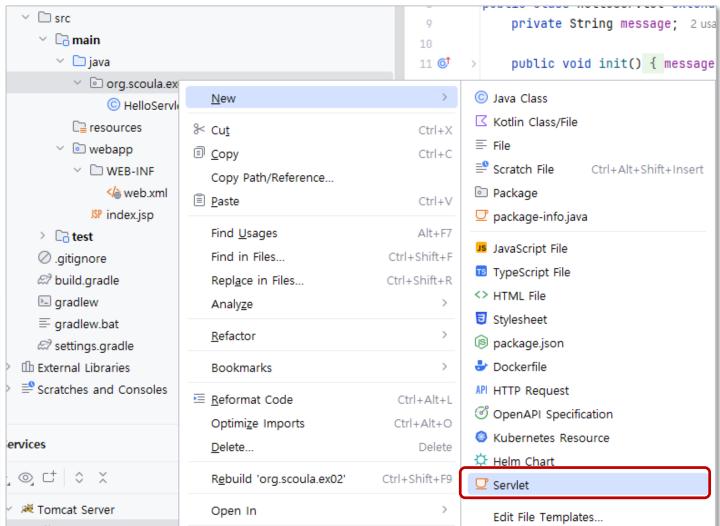
#end
import java.io.IOException;
import java.io.PrintWriter;
```

✓ Servlet 템플릿 수정

```
@WebServlet("$URL_MAPPING")
public class ${Class_Name} extends HttpServlet {
   @Override
   protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
       // MIME 타입 설정
       response.setContentType("text/html;charset=UTF-8");
       // 자바 I/0
       PrintWriter out = response.getWriter();
       // html 작성 및 출력
       out.print("<html><body>");
       out.print("");
       out.print("</body></html>");
```

ResponseServlet 클래스 만들기





ResponseServlet.java

```
@WebServlet("/response")
public class ResponseServlet extends HttpServlet {
  protected void doGet(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    // MIME 타입 설정
    response.setContentType("text/html;charset=UTF-8");
    // 자바 I/0
    PrintWriter out = response.getWriter();
    // html 작성 및 출력
    out.print("<html><body>");
    out.print("ResponseServlet 요청 성송");
    out.print("</body></html>");
```

http://localhost:8080/response

ResponseServlet 요청 성공

ResponseServlet ?? ??