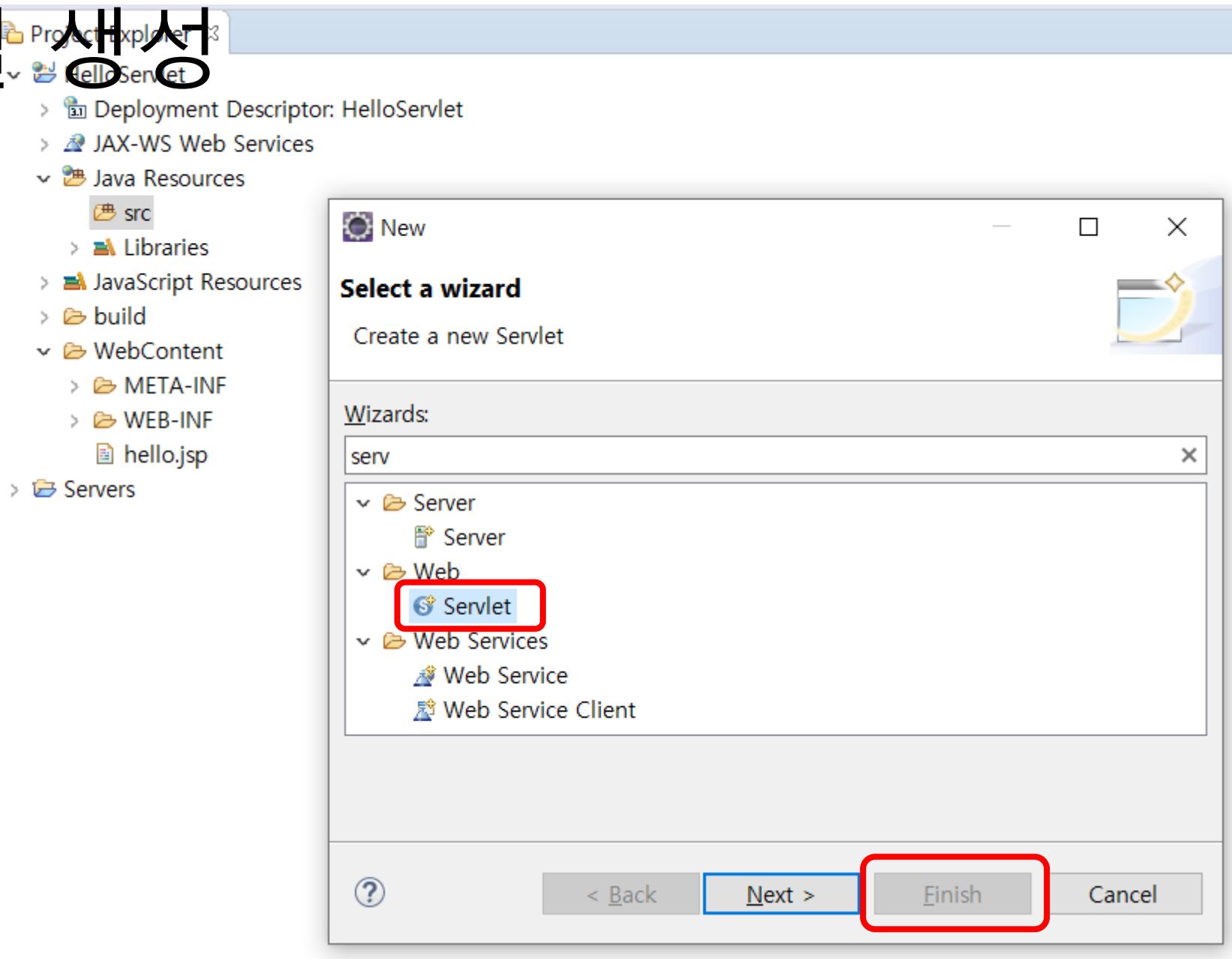


# Ex2\_Servlet

교사 최선경

# Servlet 파일 생성



The screenshot shows a Java development environment with the following details:

- Project Explorer:** A tree view on the left showing the project structure:
  - HelloServlet
  - Deployment Descriptor: HelloServlet
  - JAX-WS Web Services
  - Java Resources
    - src
      - com.web
        - HelloServlet.java
    - Libraries
    - JavaScript Resources
    - build
    - WebContent
      - META-INF
      - WEB-INF
        - hello.jsp
    - Servers

The file `HelloServlet.java` is highlighted with a pink rectangle.

**Code Editor:** The main window displays the Java code for `HelloServlet.java`:

```
1 package com.web;
2
3 import java.io.IOException;
4
5 /**
6  * Servlet implementation class HelloServlet
7  */
8 @WebServlet("/HelloServlet")
9 public class HelloServlet extends HttpServlet {
10    private static final long serialVersionUID = 1L;
11
12    /**
13     * @see HttpServlet#HttpServlet()
14     */
15    public HelloServlet() {
16        super();
17        // TODO Auto-generated constructor stub
18    }
19
20    /**
21     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
22     */
23    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
24        // TODO Auto-generated method stub
25        response.getWriter().append("Served at: ").append(request.getContextPath());
26    }
27
28    /**
29     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
30     */
31    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
32        // TODO Auto-generated method stub
33        doGet(request, response);
34    }
35
36    /**
37     * @see HttpServlet#doPut(HttpServletRequest request, HttpServletResponse response)
38     */
39    protected void doPut(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
40        doGet(request, response);
41    }
42}
```

ex2 - HelloServlet/src/com/web/HelloServlet.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

http://localhost:8090/HelloServlet/HelloServlet x

8090/HelloServlet/HelloServlet

Served at: /HelloServlet

```
1 package com.web;
2
3 import java.io.IOException;
4
5 @WebServlet("/HelloServlet")
6 public class HelloServlet extends HttpServlet {
7     private static final long serialVersionUID = 1L;
8
9     public HelloServlet() {
10         super();
11     }
12
13     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
14         IOException {
15         response.getWriter().append("Served at: ").append(request.getContextPath());
16     }
17
18     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
19         IOException {
20         doGet(request, response);
21     }
22
23 }
24
25 }
```

# Servlet의 Url 경로 축약 방법

- 1. **web.xml** 을 사용
- 2. **annotation** 을 사용 ( 서블릿 3.0 부터 사용가능)
- ??? 경로를 줄여서 사용하는 **이유** ?? 교재를 읽고 알아두자!

# Ex2\_2 LifeCycle

Project Explorer

LifeCycle

- Deployment Descriptor: LifeCycle
- JAX-WS Web Services
- Java Resources
  - src
  - Libraries
  - JavaScript Resources
  - build
  - WebContent
- Servers

Create Servlet

### Create Servlet

Specify class file destination.

Project: LifeCycle

Source folder: /LifeCycle/src

Java package: com.web

Class name: FirstServlet

Superclass: javax.servlet.http.HttpServlet

Use an existing Servlet class or JSP

Class name: FirstServlet

The screenshot shows the Eclipse IDE interface with the 'Project Explorer' view on the left. A project named 'LifeCycle' is selected. Inside 'LifeCycle', there are several nodes: 'Deployment Descriptor: LifeCycle', 'JAX-WS Web Services', 'Java Resources' (which contains 'src', 'Libraries', 'JavaScript Resources', 'build', and 'WebContent'), and 'Servers'. On the right, a 'Create Servlet' dialog is open. The dialog has a title 'Create Servlet' and a sub-instruction 'Specify class file destination.' It contains fields for 'Project' (set to 'LifeCycle'), 'Source folder' ('/LifeCycle/src'), 'Java package' ('com.web'), 'Class name' ('FirstServlet'), and 'Superclass' ('javax.servlet.http.HttpServlet'). Below these fields is a checkbox labeled 'Use an existing Servlet class or JSP'. At the bottom of the dialog are five buttons: '?', '< Back', 'Next >', 'Finish', and 'Cancel'. The 'Class name' field and the 'Next >' button are both highlighted with red boxes, indicating they are the focus of the current step in the wizard.

 Create Servlet

## Create Servlet

Enter servlet deployment descriptor specific information.

Name:

Description:

Initialization parameters:

Name	Value	Description

, ,

URL mappings:

/FirstServlet , ,

Asynchronous Support

, ,

 Create Servlet

## Create Servlet

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

Modifiers:  public  abstract  final

Interfaces:   
,

Which method stubs would you like to create?

<input checked="" type="checkbox"/> Constructors from superclass	<input checked="" type="checkbox"/> Inherited abstract methods	<input type="checkbox"/> getServletConfig
<input checked="" type="checkbox"/> init	<input checked="" type="checkbox"/> destroy	<input type="checkbox"/> doGet
<input type="checkbox"/> getServletInfo	<input checked="" type="checkbox"/> service	<input type="checkbox"/> doGet
<input checked="" type="checkbox"/> doPost	<input type="checkbox"/> doPut	<input type="checkbox"/> doDelete
<input type="checkbox"/> doHead	<input type="checkbox"/> doOptions	<input type="checkbox"/> doTrace

, ,

The screenshot shows the Eclipse IDE interface with the following components:

- Project Explorer:** Shows the project structure with a node for "Lifecycle".
- Editor:** Displays the Java code for `FirstServlet.java`. The code is annotated with Javadoc comments and annotations like `@WebServlet("/FirstServlet")`.
- Outline View:** A red box highlights the outline view on the right, which lists the class members:
  - `serialVersionUID : long`
  - `FirstServlet()`
  - `init(ServletConfig config) throws ServletException`
  - `destroy()`
  - `service(HttpServletRequest request, HttpServletResponse response) throws ServletException`
  - `doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException`
  - `doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException`

```
1 package com.web;
2
3 import java.io.IOException;
4
5 /**
6  * Servlet implementation class FirstServlet
7  */
8 @WebServlet("/FirstServlet")
9 public class FirstServlet extends HttpServlet {
10     private static final long serialVersionUID = 1L;
11
12     /**
13      * @see HttpServlet#HttpServlet()
14      */
15     public FirstServlet() {
16         super();
17         // TODO Auto-generated constructor stub
18     }
19
20     /**
21      * @see Servlet#init(ServletConfig)
22      */
23     public void init(ServletConfig config) throws ServletException {
24         // TODO Auto-generated method stub
25     }
26
27     /**
28      * @see Servlet#destroy()
29      */
30     public void destroy() {
31         // TODO Auto-generated method stub
32     }
33
34     /**
35      * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
36      */
37     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException {
38         // TODO Auto-generated method stub
39     }
40
41     /**
42      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
43      */
44     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException {
45         // TODO Auto-generated method stub
46     }
47
48     /**
49      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
50      */
51     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException {
52         // TODO Auto-generated method stub
53     }
54 }
```

# 코드 수정

```
public void init(ServletConfig config) throws ServletException {
    System.out.println("init----111-----");
}

public void destroy() {
    System.out.println("destroy-----destroy");
}

protected void service(HttpServletRequest request, HttpServletResponse resp
    System.out.println("service-----");
}
```

http://localhost:8090/LifeCycle/FirstServlet

첫번째실행  
-> 새로고침을 눌러  
하단의 console 확인

```
FirstServlet.java
1 package com.web;
2
3 import java.io.IOException;
4
5 @WebServlet("/FirstServlet")
6 public class FirstServlet extends HttpServlet {
7     private static final long serialVersionUID = 1L;
8
9     public FirstServlet() {
10         super();
11         // TODO Auto-generated constructor stub
12     }
13
14     public void init(ServletConfig config) throws ServletException {
15         System.out.println("init----111----");
16     }
17
18     public void destroy() {
19         System.out.println("destroy-----destroy");
20     }
21
22     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
23         System.out.println("service-----");
24     }
25
26     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
27         // TODO Auto-generated method stub
28         response.getWriter().append("Served at: ").append(request.getContextPath());
29     }
30
31     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
32         // TODO Auto-generated method stub
33     }
34
35     public void destroy() {
36         // TODO Auto-generated method stub
37     }
38 }
```

Markers Properties Servers Data Source Explorer Snippets Console

Tomcat v8.5 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre1.8.0\_321\bin\javaw.exe (2022. 3. 9. 오전 7:28:57)

3월 09, 2022 7:29:00 오전 org.apache.coyote.AbstractProtocol start

정보: 프로토콜 핸들러 ["http-nio-8090"]를(를) 시작합니다.

3월 09, 2022 7:29:00 오전 org.apache.catalina.startup.Catalina start

정보: Server startup in 1903 ms

init----111----

service-----

service-----

service-----

service-----

# 수정후 재실행~ 직전에 확인해보자

```
public void init(ServletConfig config) throws ServletException {  
    System.out.println("init----222-----");  
}
```

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Standard Eclipse menu items: File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Toolbar:** Includes icons for file operations (New, Open, Save, Cut, Copy, Paste), search, run, and debug.
- Quick Access:** A panel on the right side of the toolbar.
- Left Sidebar:** Shows a tree view of the project structure.
- Central Area:** A code editor window titled "FirstServlet.java" containing Java code for a servlet. The code includes imports, class definition, constructor, and a doGet method stub.
- Bottom Area:** A "Server" dialog box is displayed over the code editor. It asks if the server needs to be restarted and provides two options: "Restart server" (selected) and "Continue without restarting". There is also a "Remember my decision" checkbox and "OK" and "Cancel" buttons.
- Console Tab:** Shows logs from Tomcat v8.5 Server at localhost. The log output includes:
  - service-----
  - 3월 09, 2022 7:30:20 오전 org.apache.catalina.core.StandardContext reload  
정보: 이름이 [/LifeCycle]인 컨텍스트를 다시 로드하는 작업이 시작되었습니다.
  - destroy-----destroy
  - 3월 09, 2022 7:30:20 오전 org.apache.catalina.core.StandardContext reload  
정보: 이름이 [/LifeCycle]인 컨텍스트를 다시 로드하는 것을 완료했습니다.

Markers Properties Servers Data Source Explorer Snippets Console

Tomcat v8.5 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre1.8.0\_321\bin\javaw.exe (2022. 3. 9. 오전 7:28:57)

```
service-----  
service-----  
service-----  
service-----  
3월 09, 2022 7:30:20 오전 org.apache.catalina.core.StandardContext reload  
정보: 이름이 [/LifeCycle]인 컨텍스트를 다시 로드하는 작업이 시작되었습니다.  
destroy-----destroy  
3월 09, 2022 7:30:20 오전 org.apache.catalina.core.StandardContext reload  
정보: 이름이 [/LifeCycle]인 컨텍스트를 다시 로드하는 것을 완료했습니다.
```

# Servlet의 LifeCycle 생명주기

- 1. **init** 초기화
- 2. **service** 클라이언트의 요청 처리
- 3. **destroy** 자원의 해제시
- ??? 서블릿의 실행흐름을 이해한다
- **생성 -> 실행...실행...실행...실행... -> 소멸**