

삼성 청년 SW 아카데미

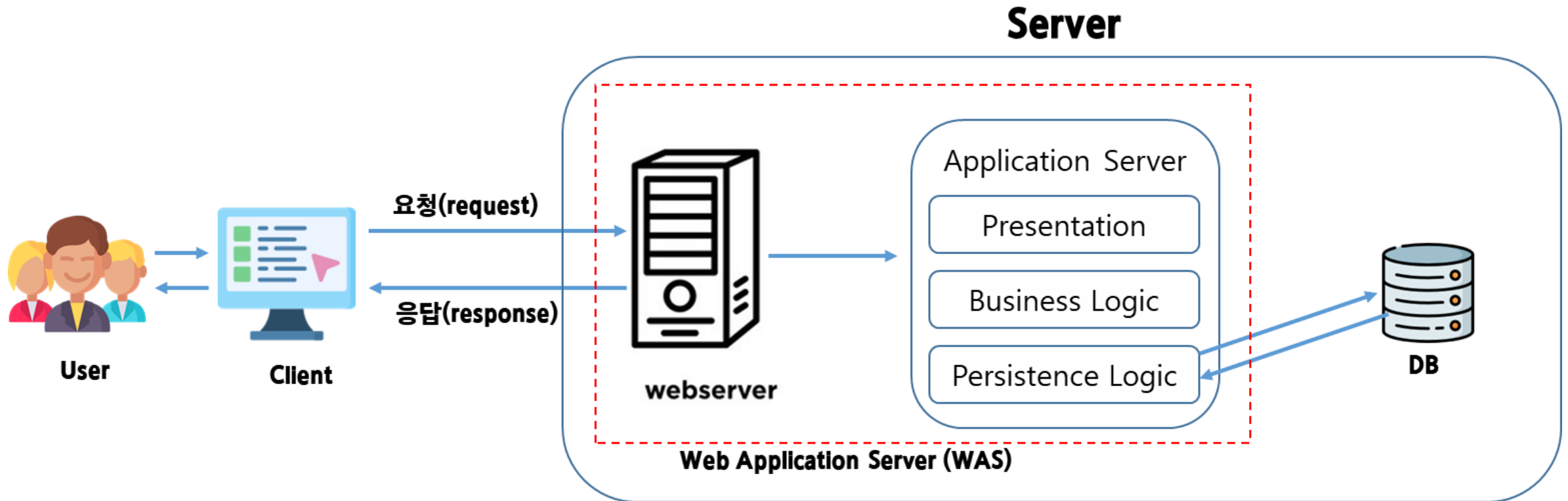
Web-Backend

Servlet

- 웹 프로그래밍
- Servlet
- Servlet Parameter

웹프로그래밍


✓ Web Architecture



✓ 웹과 웹 프로그래밍

- URL (Uniform Resource Locator) - 웹 상의 자원을 참조하기 위한 웹 주소
- 웹 페이지 (Web page) - 웹 브라우저를 통해서 보여지는 화면
- 웹 서버 (Web Server) - 클라이언트 요청에 맞는 응답(웹 페이지)을 제공
- 웹 어플리케이션 (Web Application) - 웹 서버를 기반으로 실행되는 응용 소프트웨어
- 웹 어플리케이션 서버 (Web Application Server, WAS) - 요청이 오면 알맞은 프로그램을 실행하여 응답 만들고 제공하는 서버

✓ Tomcat 설치 및 등록



Apache Tomcat®

Apache Tomcat

Home
Taglibs
Maven Plugin

Download

Which version?
Tomcat 11 (alpha)
Tomcat 10
Tomcat 9
Tomcat 8
Tomcat Migration Tool for Jakarta EE
Tomcat Connectors
Tomcat Native
Taglibs
Archives

Apache Tomcat

The Apache Tomcat® software is an open-source implementation of the Java Servlet, Java Server Pages (JSP), Java Expression Language (EL), and WebSocket technologies. It is developed as part of the Jakarta EE platform.

The Jakarta EE platform is developed as part of the Java EE specification.

The Apache Tomcat software is the best-of-breed development of the Java EE specification.

Apache Tomcat software is the best-of-breed development of the Java EE specification.

the [PoweredBy](#) wiki page

Apache Tomcat, Tomcat

Tomcat 10.1.6 Release

The Apache Tomcat Project

9.0.72



Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

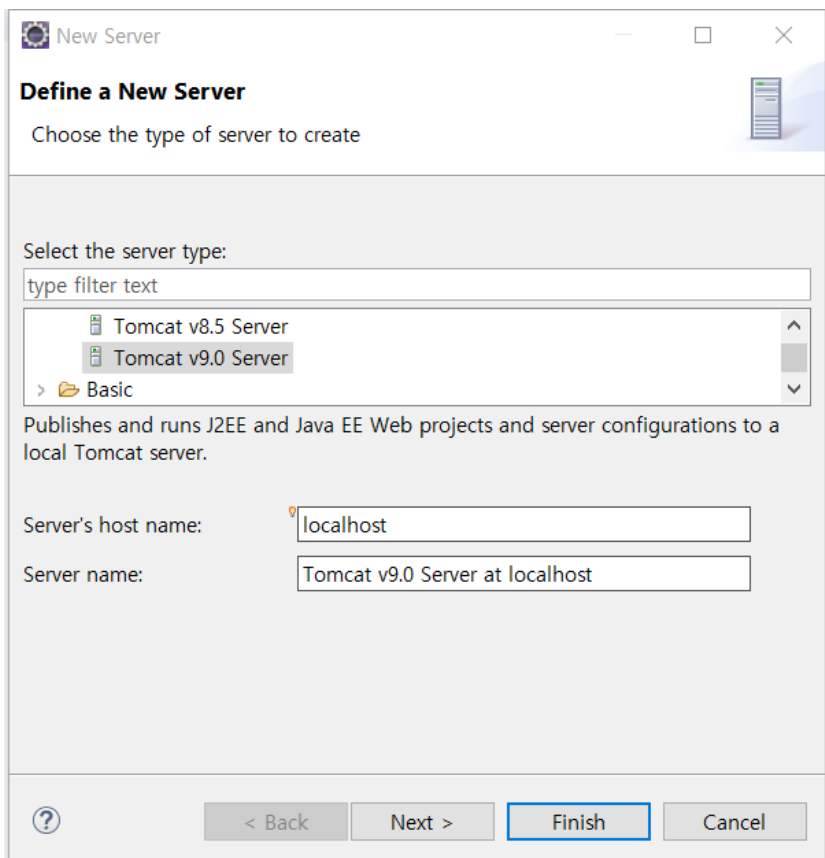
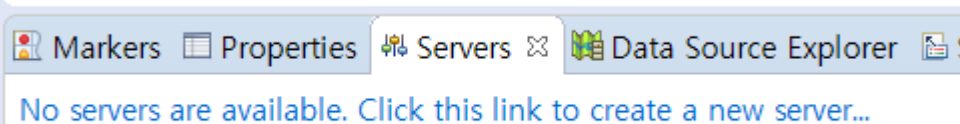
- Core:
 - [zip](#) ([pgp](#), [sha512](#))
 - [tar.gz](#) ([pgp](#), [sha512](#))
 - [32-bit Windows zip](#) ([pgp](#), [sha512](#))
 - [64-bit Windows zip](#) ([pgp](#), [sha512](#))
 - [32-bit/64-bit Windows Service Installer](#) ([pgp](#), [sha512](#))
- Full documentation:
 - [tar.gz](#) ([pgp](#), [sha512](#))
- Deployer:
 - [zip](#) ([pgp](#), [sha512](#))
 - [tar.gz](#) ([pgp](#), [sha512](#))
- Embedded:
 - [tar.gz](#) ([pgp](#), [sha512](#))
 - [zip](#) ([pgp](#), [sha512](#))

Source Code Distributions

- [tar.gz](#) ([pgp](#), [sha512](#))
- [zip](#) ([pgp](#), [sha512](#))

 [apache-tomcat-9.0.72-windows-x64.zip](#)
 [apache-tomcat-9.0.72](#)

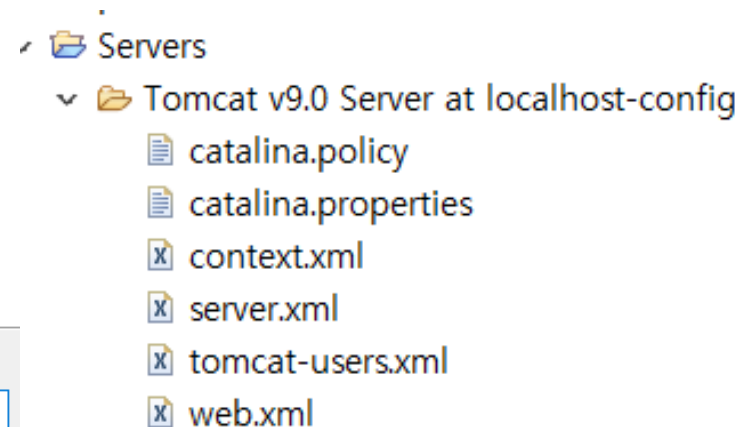
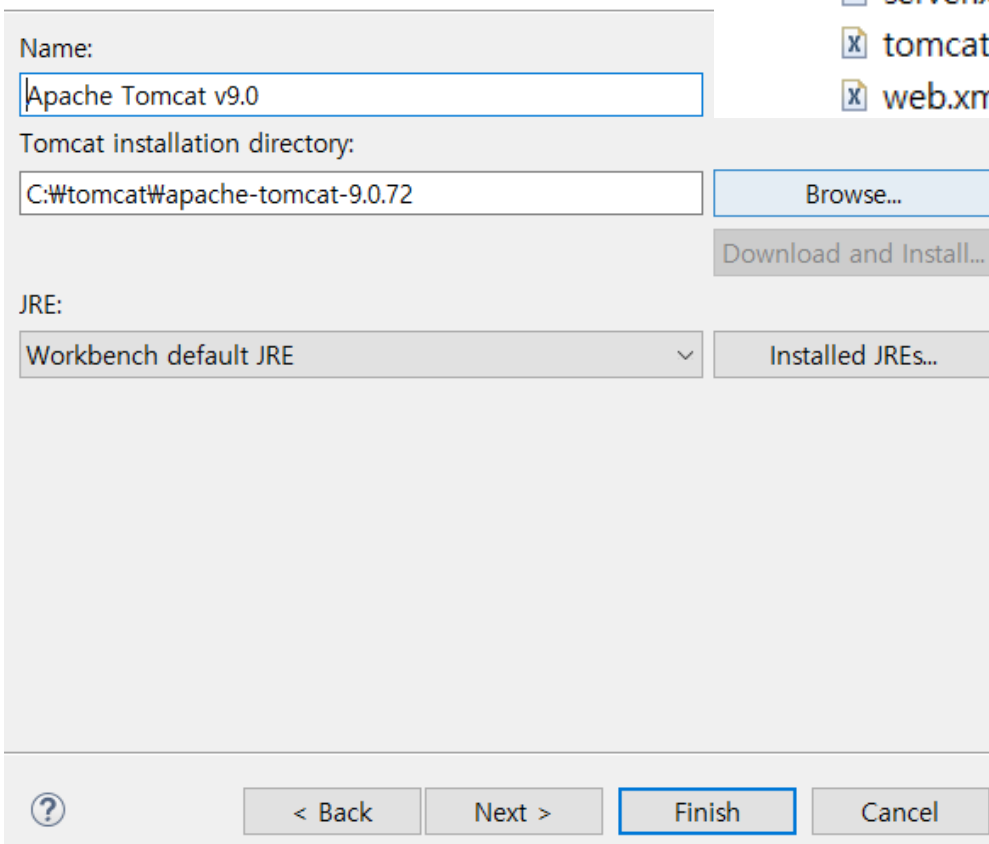
✓ Tomcat 설치 및 등록



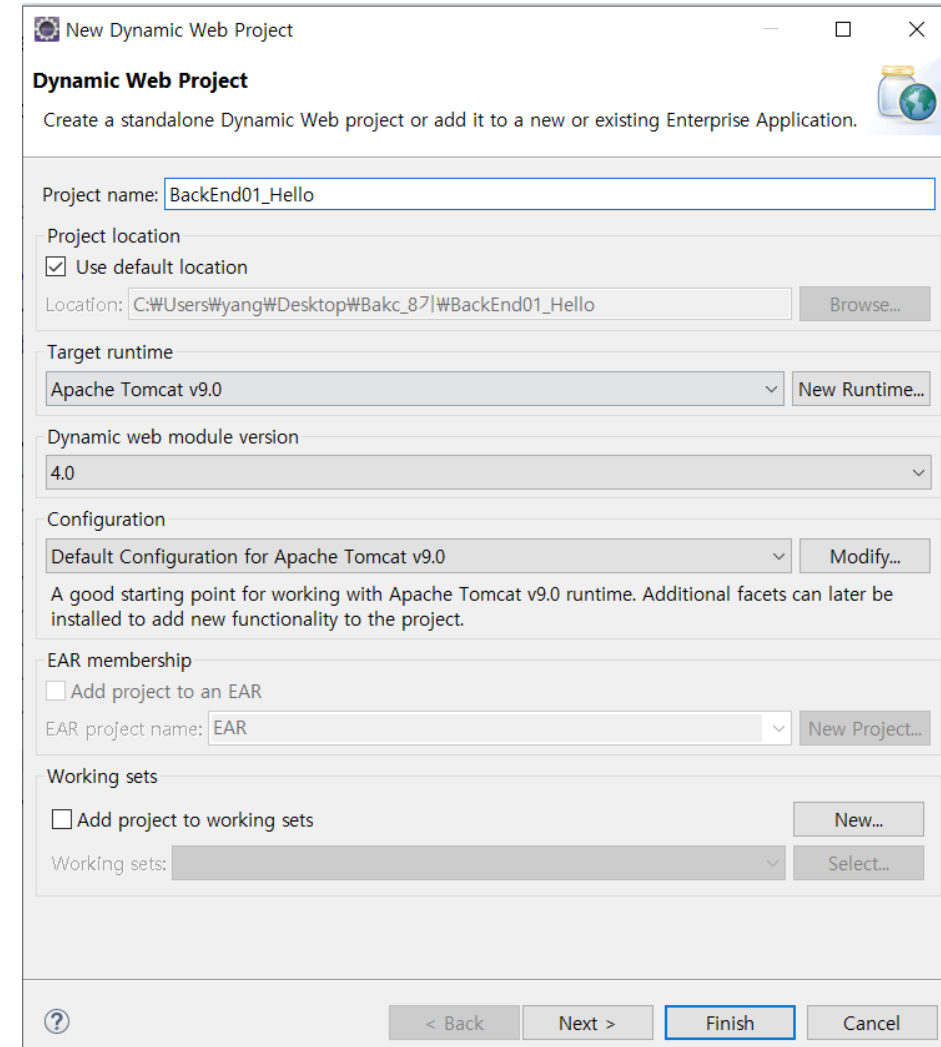
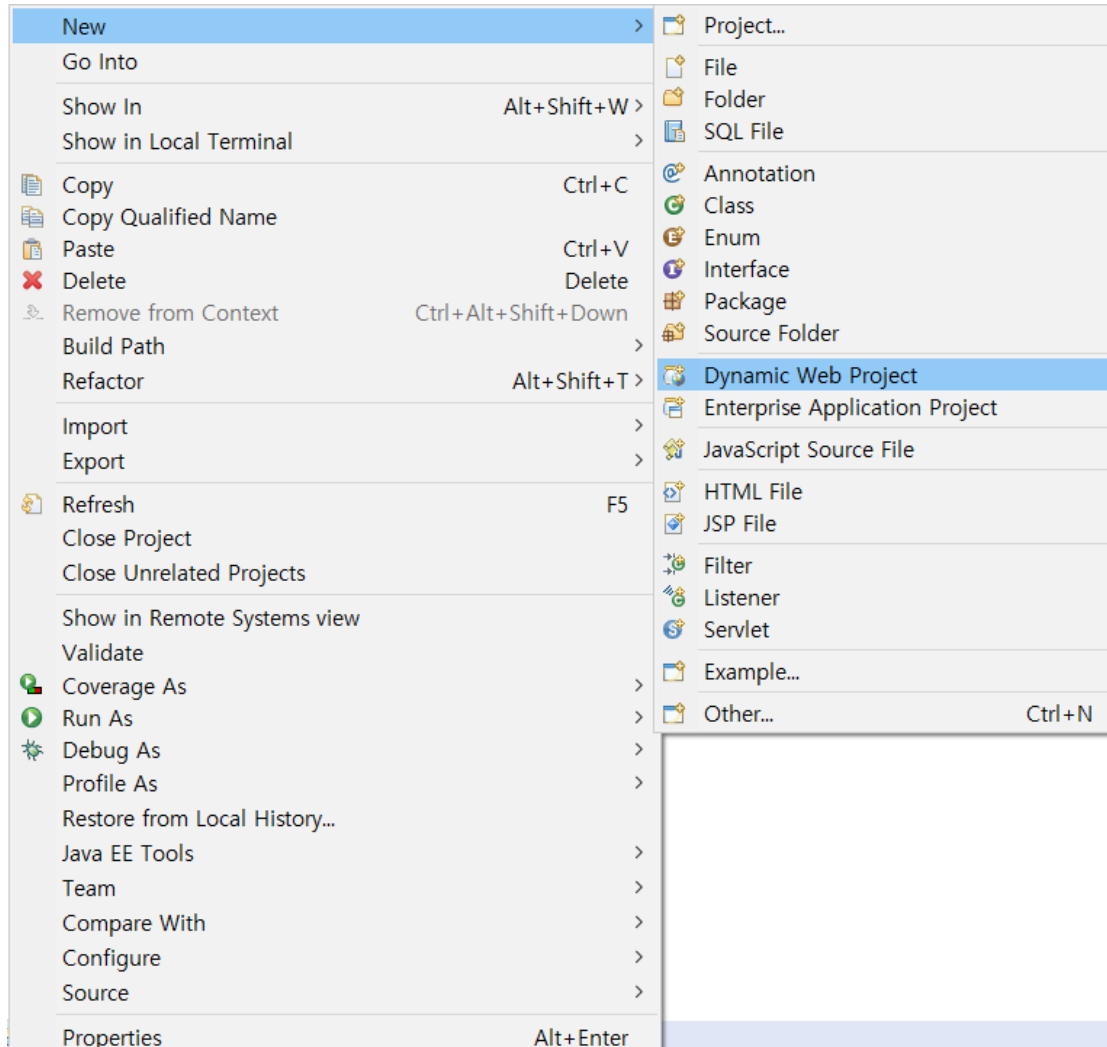
New Server

Tomcat Server

Specify the installation directory



✓ Dynamic Web Project



✓ Dynamic Web Project 구조

📁 BackEnd01_Hello

- > 📄 Deployment Descriptor: BackEnd01_Hello
- > 🌐 JAX-WS Web Services
- ▼ 📁 Java Resources ①
 - 📁 src
 - ▼ 📁 Libraries
 - > 📄 Apache Tomcat v9.0 [Apache Tomcat v9.0]
 - > 📄 JRE System Library [zulu-8]
- > 📄 JavaScript Resources
- > 📁 build
- ▼ 📁 WebContent ②
 - > 📁 META-INF
 - ▼ 📁 WEB-INF
 - 📁 lib

✓ Dynamic Web Project 구조

1. Web Application 실행에 필요한 Java 관련 Resource를 포함
2. Web Application 실행에 필요한 html, javascript, css, JSP, Image 등 웹 콘텐츠를 포함.
웹 어플리케이션 설정파일인 web.xml도 WebContent/WEB-INF/에 위치한다.

✓ HelloServlet 작성

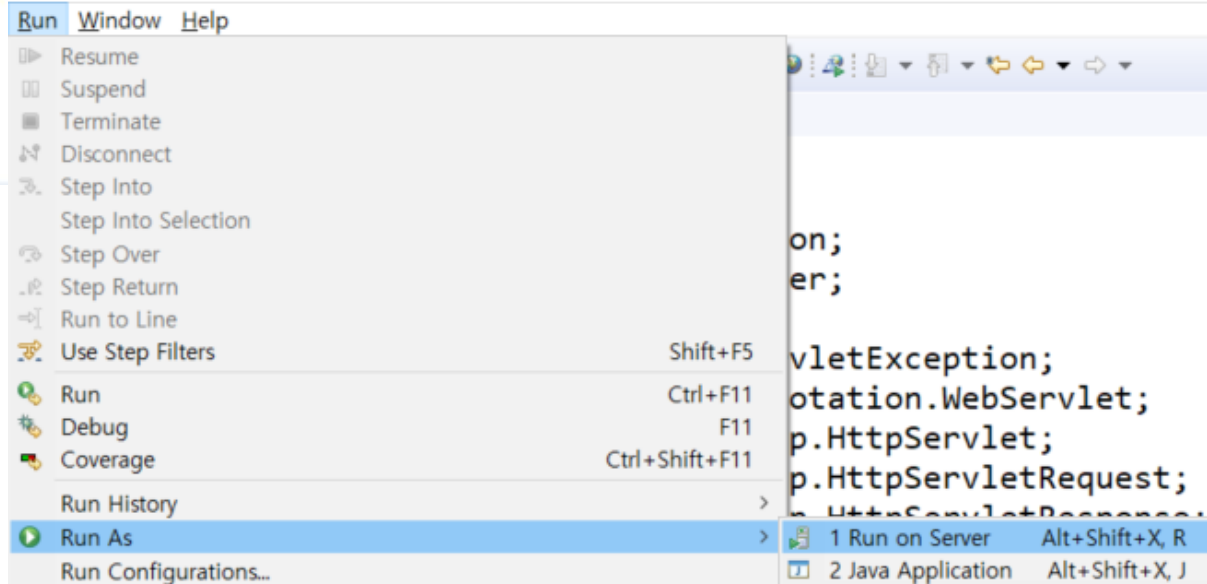
The screenshot shows an IDE interface. On the left, the 'New' menu is open, and 'Servlet' is selected under the 'Other...' category. The 'Create Servlet' dialog box is open on the right, showing the following configuration:

- Project: BackEnd01_Hello
- Source folder: /BackEnd01_Hello/src
- Java package: com.ssafy.hello
- Class name: HelloServlet
- Superclass: javax.servlet.http.HttpServlet
- ☐ Use an existing Servlet class or JSP
- Class name: HelloServlet

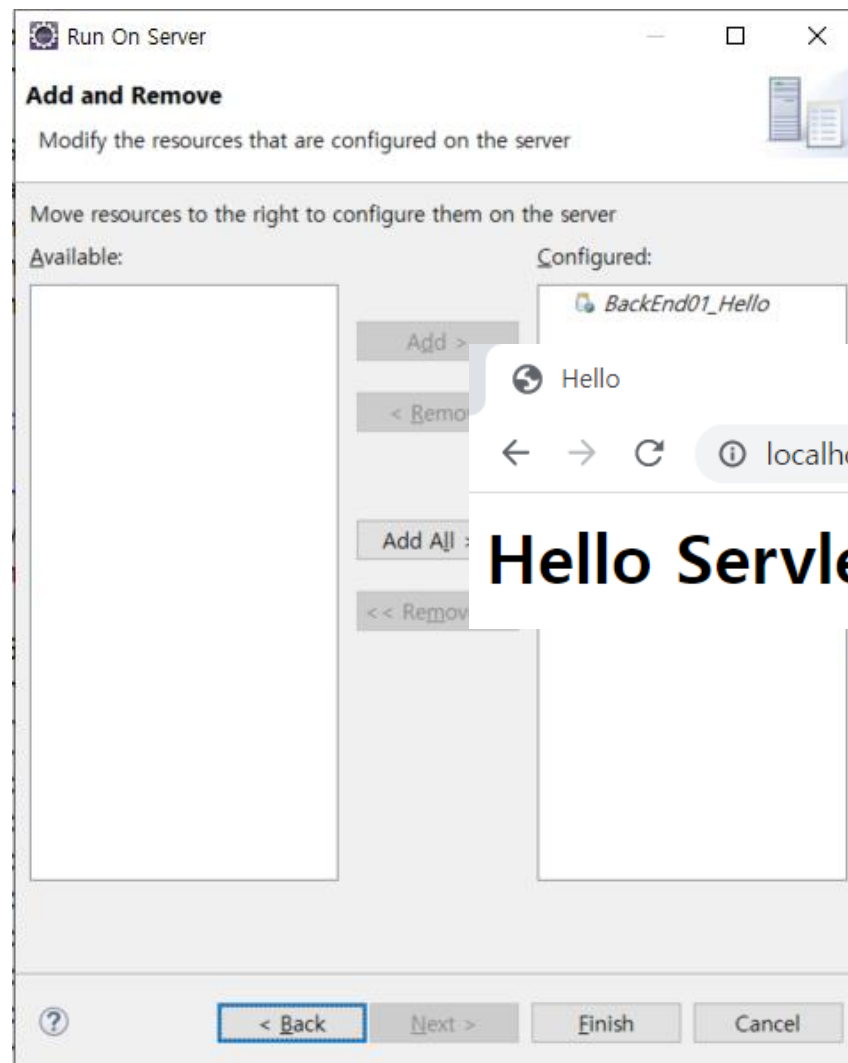
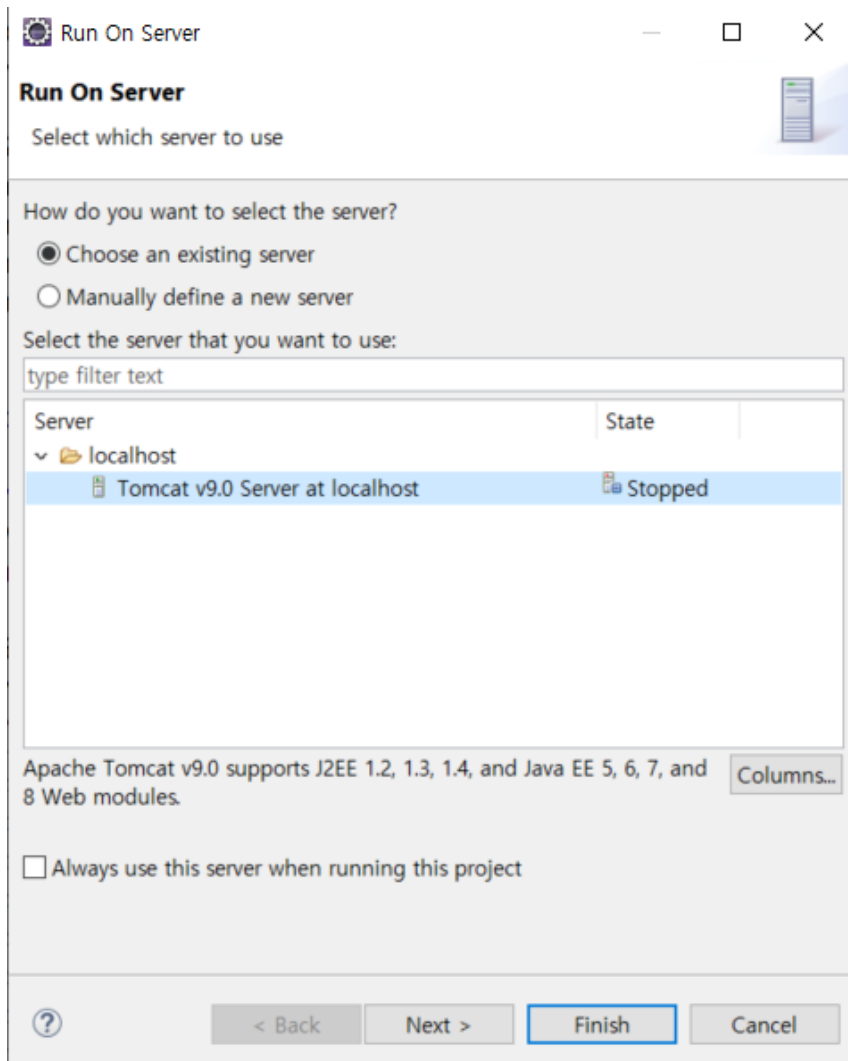
At the bottom of the dialog, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'. The 'Finish' button is highlighted. In the background, a code editor shows a line of code with the number '31'.

✓ HelloServlet 작성

```
15 @WebServlet("/HelloServlet")
16 public class HelloServlet extends HttpServlet {
17     private static final long serialVersionUID = 1L;
18
19     protected void doGet(HttpServletRequest request, HttpServletResponse response)
20         throws ServletException, IOException {
21         PrintWriter writer = response.getWriter();
22         writer.append("<html>");
23         writer.append("<head>");
24         writer.append("<title>Hello</title>");
25         writer.append("</head>");
26         writer.append("<body>");
27         writer.append("<h1>Hello Servlet!</h1>");
28         writer.append("</body>");
29         writer.append("</html>");
30     }
31
32 }
```



✓ HelloServlet 작성



Servlet

✓ Servlet

- Server + Applet 의 합성어
- 자바를 이용하여 웹에서 실행되는 프로그램을 작성하는 기술
- 자바를 이용하여 웹페이지를 동적으로 생성할 수 있음.
- Servlet 은 자바 코드 안에 HTML을 포함.

✓ Servlet API

javax.servlet

Interface Servlet

All Known Subinterfaces:

HttpJspPage, JspPage

All Known Implementing Classes:

FacesServlet, GenericServlet, Http

javax.servlet

Class GenericServlet

java.lang.Object

javax.servlet.GenericServlet

All Implemented Interfaces:

Serializable, Servlet, ServletConfig

Direct Known Subclasses:

HttpServlet

javax.servlet.http

Class HttpServlet

java.lang.Object

javax.servlet.GenericServlet

javax.servlet.http.HttpServlet

All Implemented Interfaces:

Serializable, Servlet, ServletConfig

✓ Servlet 등록방법 (web.xml)

BackEnd02_Servlet

> Deployment Descriptor: BackEnd02_Servlet

> JAX-WS Web Services

▼ Java Resources

▼ src

> com.ssafy.myservlet

> Libraries

> JavaScript Resources

> build

▼ WebContent

> META-INF

▼ WEB-INF

lib

web.xml

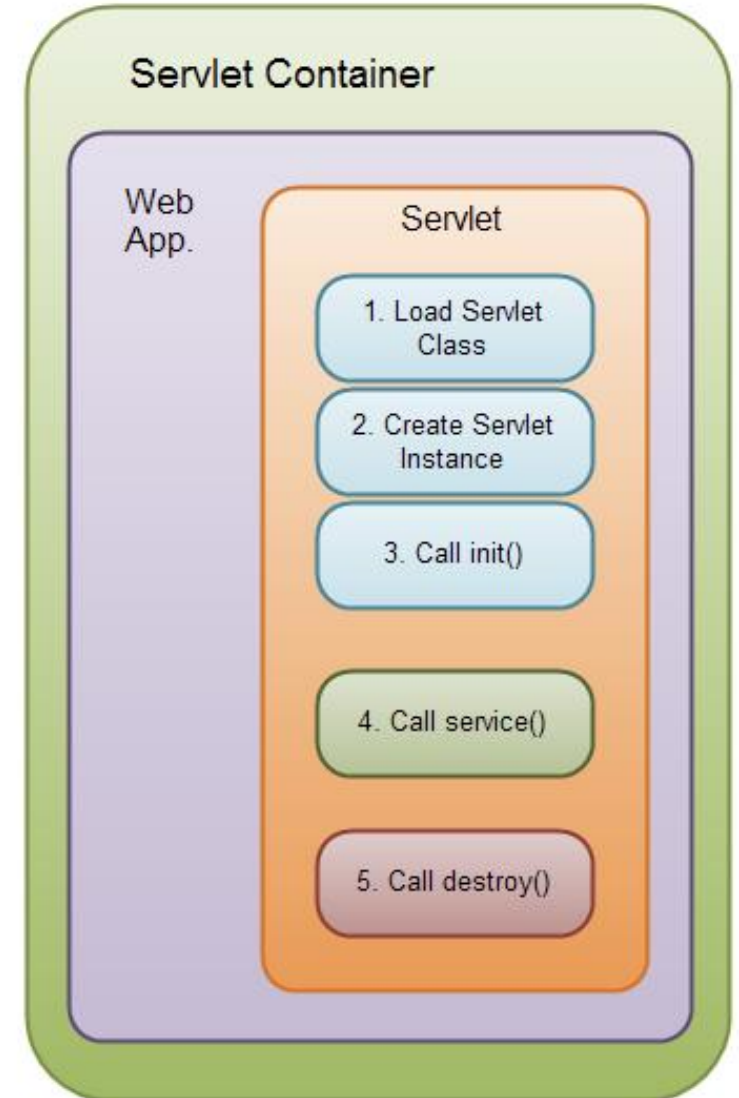
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns
3   <display-name>BackEnd02_Servlet</display-name>
4   <welcome-file-list>
5     <welcome-file>index.html</welcome-file>
6     <welcome-file>index.htm</welcome-file>
7     <welcome-file>index.jsp</welcome-file>
8     <welcome-file>default.html</welcome-file>
9     <welcome-file>default.htm</welcome-file>
10    <welcome-file>default.jsp</welcome-file>
11  </welcome-file-list>
12
13  <servlet>
14    <servlet-name>MyServlet</servlet-name>
15    <servlet-class>com.ssafy.myservlet.MyServlet04</servlet-class>
16  </servlet>
17  <servlet-mapping>
18    <servlet-name>MyServlet</servlet-name>
19    <url-pattern>/MyServlet</url-pattern>
20  </servlet-mapping>
21 </web-app>
```

✓ Servlet 등록방법 (Annotation)

```
@WebServlet("/MyServlet2")
public class MyServlet05 extends HttpServlet{
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        PrintWriter writer = response.getWriter();
        writer.append("<html>");
        writer.append("<head>");
        writer.append("<title>Hello</title>");
        writer.append("</head>");
        writer.append("<body>");
        writer.append("<h1>MyServlet5</h1>");
        writer.append("</body>");
        writer.append("</html>");
    }
}
```

✓ Servlet 생명주기 (Life-Cycle)

- 서블릿 인스턴스는 서블릿이 포함된 웹컨테이너에 의해 제어된다.
- 서블릿 인스턴스가 존재하지 않으면 다음과 같은 작업을 수행한다
 - 서블릿클래스 로드
 - 서블릿 클래스 인스턴스 생성
 - 서블릿 인스턴스 초기화
 - 웹 컨테이너에 의한 서비스 메서드 호출
 - destroy 메서드를 호출하여 서블릿 종료
- 서비스 메서드는 요청이 들어올 때 마다 호출된다.



Servlet Parameter

✓ GET 과 POST

GET	POST
지정된 리소스에서 데이터를 요청하는 데 사용	리소스를 생성/업데이트하기 위해 서버에 데이터를 보내는 데 사용
query string(name/value 쌍)이 URL에 포함되어 전송됨. POST와 비교하여 보안에 취약함	HTTP header 의 body에 파라미터를 포함하여 전송 데이터 길이에 대한 제한 없음 매개변수가 브라우저나 웹 서버에 저장되지 않음
URL이 길이 제한이 있으므로, 전송가능한 데이터의 길이가 제한적 (URL maximum characters : 2048) ASCII 문자만 가능	제한 없음. 바이너리 데이터도 허용

✓ URL 구성요소

프로토콜 **서버** **경로**

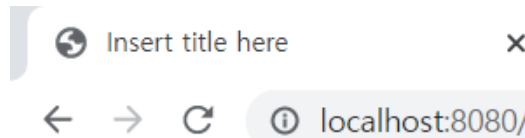
https://www.google.com/search?q=ssafy

URL Query String

구성요소	설명
프로토콜	절차를 포함한 통신규약
서버	웹 페이지를 요청할 서버의 주소, 실제 IP 주소나 도메인을 입력할 수 있다.
경로	서버 내의 상세 경로
쿼리 스트링	추가로 서버로 데이터를 전송하기 위해서 사용한다. '?' 마크를 적어 시작을 표시한다. parameter_name=value 형태로 작성하며, 파라미터가 여러 개 일 경우 '&' 로 구분하여 작성한다.

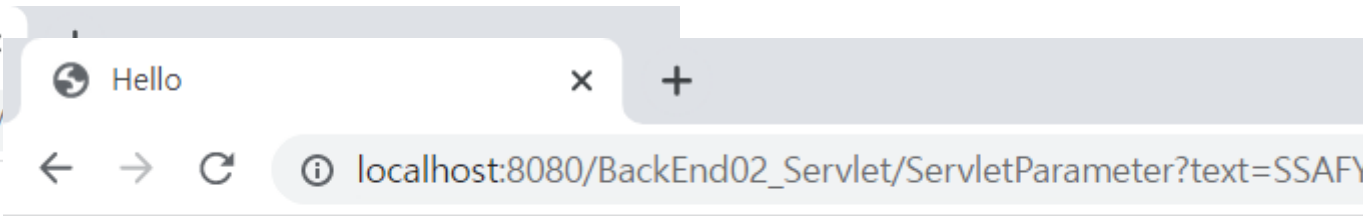
✓ input tag 실습

- text (영어 / 한글)
- number
- radio
- checkbox
- select
- textarea
- hidden

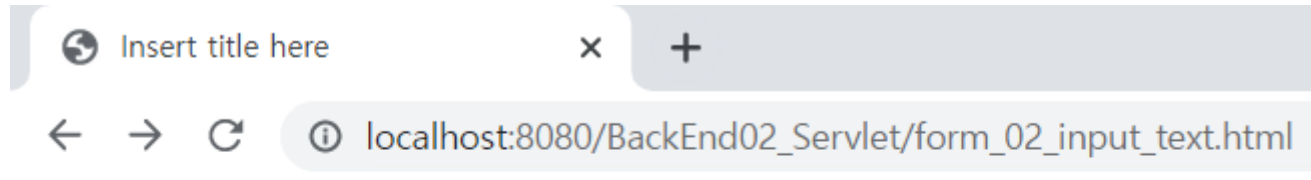


GET Form

SSAFY 제출

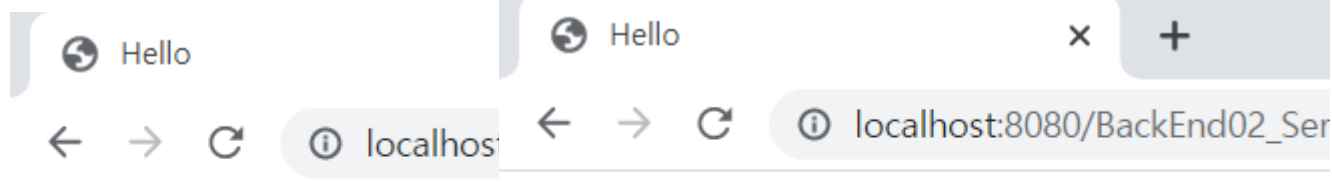


SSAFY



GET Form

즐거운 백엔드 제출



??? ???

즐거운 백엔드

✓ Front-Controller

- 웹에서 발생하는 모든 요청에 대해 호출되는 Servlet을 만들어 처리함.

다음 방송에서 만나요!

삼성 청년 SW 아카데미