

2025년 상반기 K-디지털 트레이닝

웹 어플리케이션 프로젝트

[KB] IT's Your Life

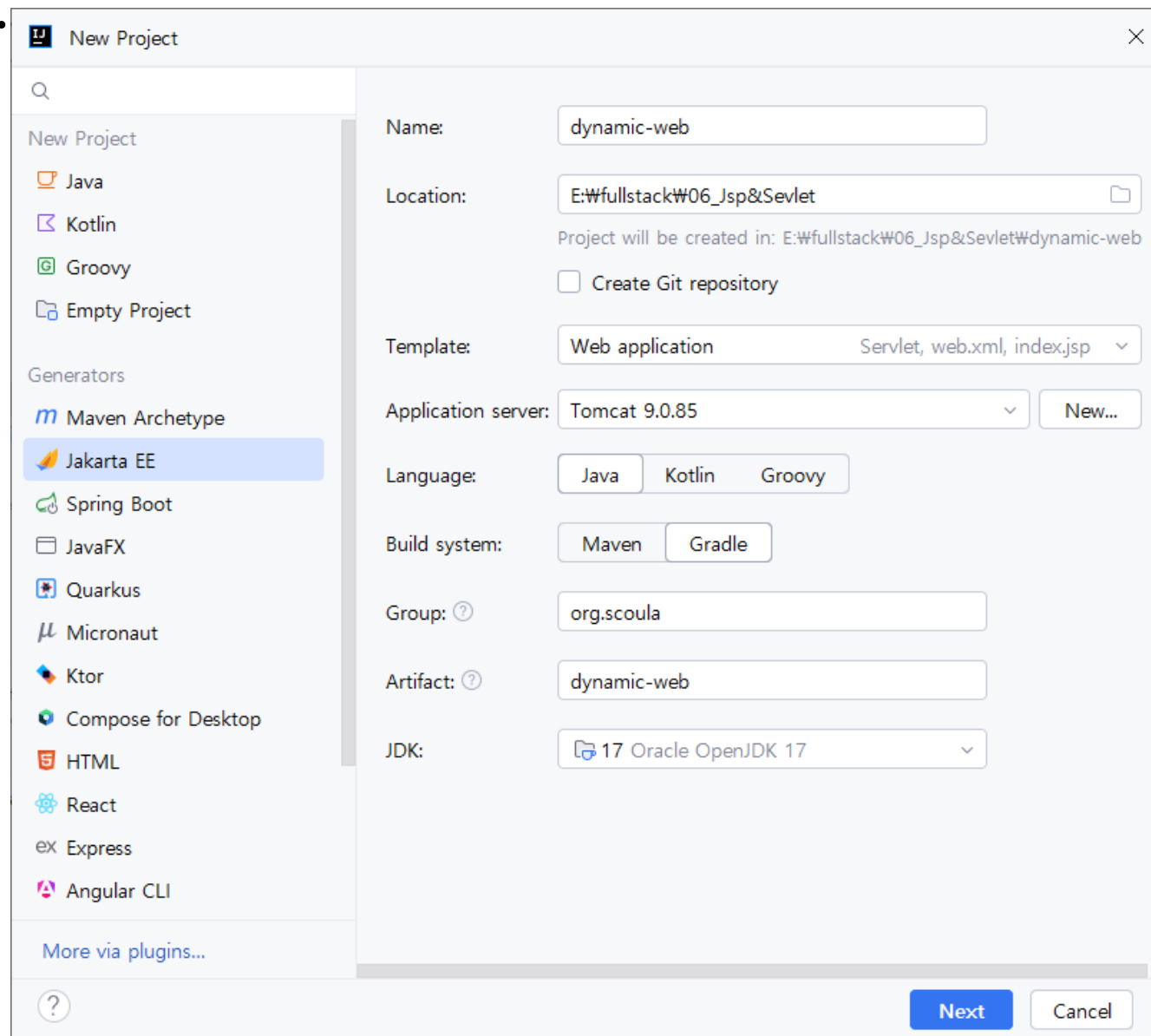
✓ 다음과 같은 설정으로 프로젝트를 생성하세요.

- 프로젝트 유형: Jakarta EE
- 프로젝트 상세 설정
 - Name: dynamic-web
 - Location:
C:\Wfullstack\W06_Jsp&Servlet
 - Template: Web application
 - Build system: Gradle
 - GroupId: org.scoula
 - ArtifactId: dynamic-web

 - Version: JavaEE 8
 - Specifications: Servlet 4.0.1

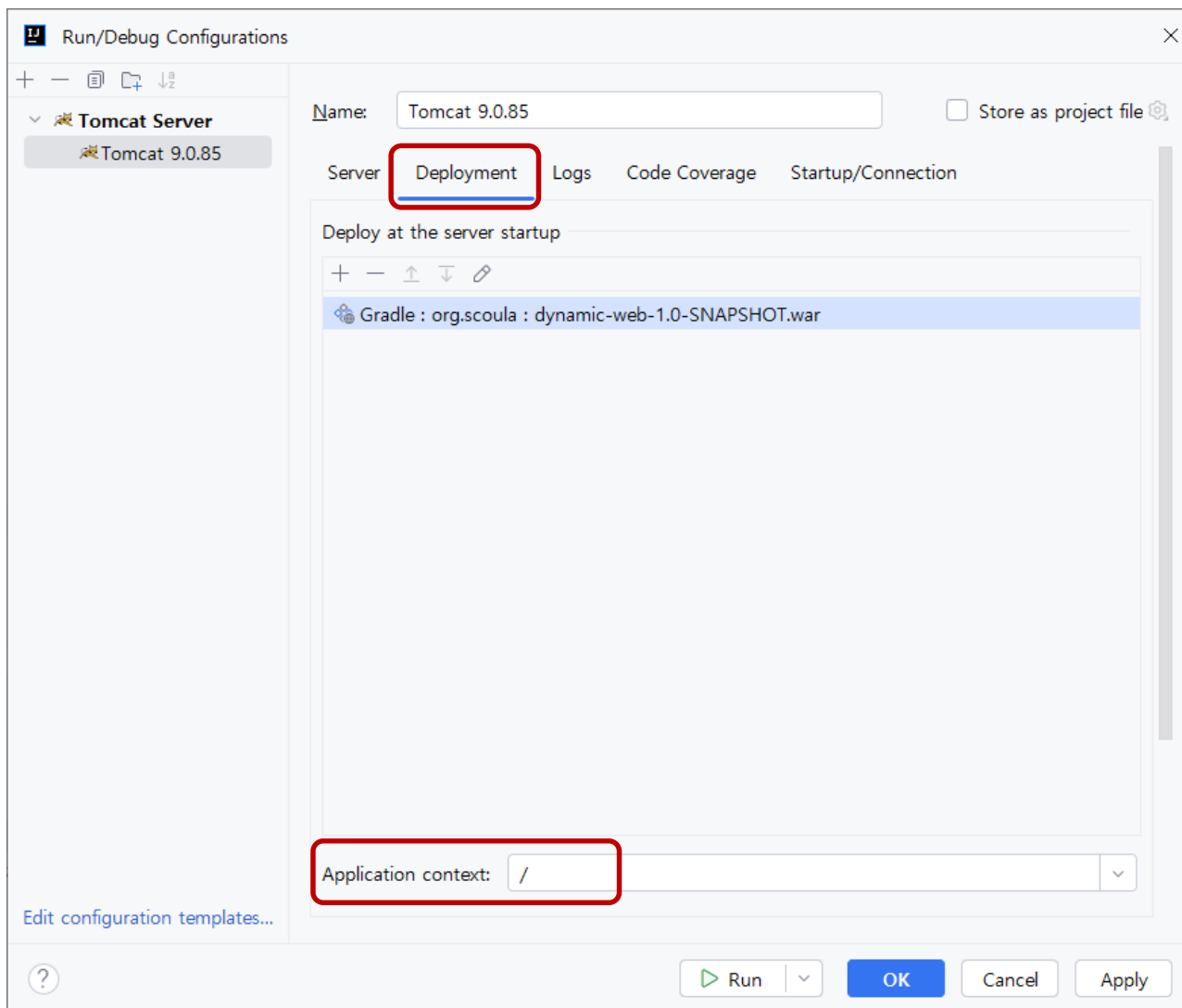
✓ 다음과 같은 설정으로 프로젝트를 생성하세요.

- 프로젝트 유형: Jakarta EE
- 프로젝트 상세 설정
 - Name: dynamic-web
 - Location: C:\Wfullstack\W06_Jsp&Servlet
 - Template: Web application
 - Build system: Gradle
 - GroupId: org.scoula
 - ArtifactId: dynamic-web
 - Version: JavaEE 8
 - Specifications: Servlet 4.0.1



- ✓ 생성된 프로젝트의 tomcat 설정에서 Context Path를 '/'로 변경하세요.

- ✓ 생성된 프로젝트의 tomcat 설정에서 Context Path를 '/'로 변경하세요.



- ✓ **/hello-servlet 요청이 왔을 때 다음과 같이 출력하세요.**
 - "홈으로" 링크 설정
 - 링크 클릭 시 홈으로 이동

Hello Servlet!

[Home](#)

✓ /hello-servlet 요청이 왔을 때 다음과 같이 출력하세요.

```
package org.scoula.dynamicweb;

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 Servlet!";
    }

    public void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException, ServletException {
        resp.setContentType("text/html");

        PrintWriter out = resp.getWriter();
        out.println("<html><body>");
        out.println("<h1>" + message + "</h1>");
        out.println("<a href=W\"/W\">Home</a>");
        out.println("</body></html>");
    }

    public void destroy() {
    }
}
```

2025년 상반기 K-디지털 트레이닝

서블릿의 이해

[KB] IT's Your Life

✓ 다음 설정대로 프로젝트를 생성하세요.

○ 프로젝트 상세 설정

- Name: ex02
- Location: C:\Wfullstack\W06_Jsp&Servlet
- Template: Web application
- Build system: Gradle
- GroupId: org.scoula
- ArtifactId: ex02
- Version: JavaEE 8
 - Specifications: Servlet 4.0.1
- context path: /

New Project

Java
Kotlin
Groovy
Empty Project

Generators

Maven Archetype
Jakarta EE
Spring Boot
JavaFX
Quarkus
Micronaut
Ktor
Compose for Desktop
HTML
React

More via plugins...

Name: ex02

Location: C:\WKB_Fullstack\W07_Jsp,Servlet
Project will be created in: C:\WKB_Fullstack\W07_Jsp,Servlet\Wex02
☐ Create Git repository

Template: Web application Servlet, web.xml, index.jsp

Application server: Tomcat 9.0.89 New...

Language: Java Kotlin Groovy

Build system: Maven Gradle

Group: ? org.scoula

Artifact: ? ex02

JDK: 17 java version "17"

Next Cancel

- ✓ **HelloServlet 클래스에서 @WebServlet을 주석 처리한 후, HelloServlet이 동작하도록 web.xml을 수정하세요.**

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 req, HttpServletResponse resp) throws IOException, ServletException {
        resp.setContentType("text/html");

        PrintWriter out = resp.getWriter();
        out.println("<html><body>");
        out.println("<h1>" + message + "</h1>");
        out.println("</body></html>");
    }

    public void destroy() {
    }
}
```

web.xml

- web.xml 수정본을 반영하려면 웹서버 재기동

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  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>
  <servlet-mapping>
    <servlet-name>helloServlet</servlet-name>
    <url-pattern>/hello-servlet</url-pattern>
  </servlet-mapping>

</web-app>
```

- ✓ 다음 두 url에 HelloServlet이 맵핑되도록 @WebServlet을 설정하세요.
 - `http://localhost/xxx`
 - `http://localhost/yyy`

HelloServlet.java

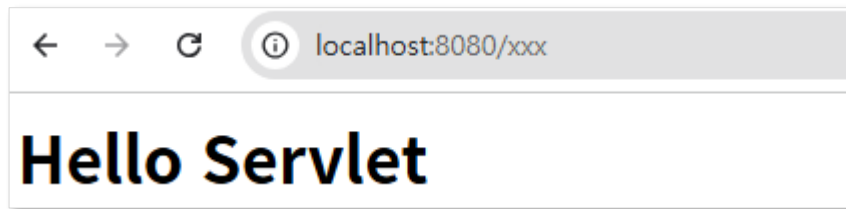
```
@WebServlet(name="MyServlet", urlPatterns={"/xxx", "/yyy" })
public class HelloServlet extends HttpServlet {

    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException, ServletException {

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

    }
}
```

- <http://localhost/xxx>
- <http://localhost/yyy>



- ✓ **HelloServlet에 생명주기 메서드 `init()`, `destroy()`를 재정의하고, 해당 메서드가 호출되는지 확인하세요.**

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 IOException, ServletException {
        ...
    }

    @Override
    public void destroy() {
        System.out.println("destroy 호출");
    }

    @Override
    public void init() throws ServletException {
        ...
        System.out.println("init호출");
    }
}
```


- ✓ **ResponseServlet** 클래스를 추가하고, `/response`요청에 대해 다음 처럼 한글 응답을 내보내세요.
 - 한글이 깨지지 않도록 문자셋을 설정하세요.

ResponseServlet 요청 성공

ResponseServlet.java

```
@WebServlet("/response")
public class ResponseServlet extends HttpServlet {

    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException, ServletException {

        // MIME 타입 설정
        resp.setContentType("text/html;charset=UTF-8");

        // 자바 I/O
        PrintWriter out = resp.getWriter();

        // html 작성 및 출력
        out.print("<html><body>");
        out.print("ResponseServlet 요청 성공");
        out.print("</body></html>");

    }
}
```

http://localhost:8080/response

ResponseServlet 요청 성공