

# Servlet

## 프로젝트 생성 및 설정

### 1. Maven Project 생성

File > New > Maven Project

Create a simple project 체크

Next

Group Id: com.douzone

Artifact Id: servlet-parctices

Packaging: pom

Finish

프로젝트 내 src 폴더 삭제

### 2. Maven Module 생성

servlet-parctices 우클릭 - New - Project > Maven Module > Next

Create a simple project 체크

Module Name: helloweb

Next

Packaging: war

Finish

### 3. web.xml 생성 및 설정

- 생성

helloweb 우클릭 - Java EE Tools - Generate Deployment Descriptor Stub

/src/webapp/WEB-INF/ 기본적인 web.xml이 생성됨

- 설정

/src/webapp/WEB-INF/web.xml 수정

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://JAVA.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" version="2.5">
  <display-name>helloweb22</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

```
xmlns="http://JAVA.sun.com/xml/ns/javaee"
```

소문자로 되어있는 java 를 대문자로 변경

- **Maven Update**

helloweb 우클릭 - Mave - Update Project... > helloweb 체크 > OK

#### 4. Targeted Runtimes 설정

helloweb 우클릭 - Properties > Targeted Runtimes > Apache Tomcat v8.5 체크 > Apply

계속 에러 뜨는 경우 Project > clean 한 번씩 해주기

#### 5. pom.xml 설정

원래 단계별로 하는데 귀찮아서..☹

- **pom.xml 내용 추가**

```
<project>
  ...
</packaging>

<!-- 아래 내용 추가 -->
<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <project.reporting.outputEncoding>UTF-
8</project.reporting.outputEncoding>
</properties>

<dependencies>
  <!-- Servlet -->
  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>3.0.1</version>
    <scope>provided</scope>
  </dependency>
</dependencies>
```

```

        <groupId>javax.servlet.jsp</groupId>
        <artifactId>jsp-api</artifactId>
        <version>2.0</version>
        <scope>provided</scope>
    </dependency>
</dependencies>

<build>
    <sourceDirectory>src/main/java</sourceDirectory>
    <resources>
        <resource>
            <directory>${project.basedir}/src/main/resources</directory>
            <excludes>
                <exclude>**/*.java</exclude>
            </excludes>
        </resource>
    </resources>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.8.0</version>
            <configuration>
                <source>1.8</source>
                <target>1.8</target>
            </configuration>
        </plugin>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-resources-plugin</artifactId>
            <configuration>
                <encoding>UTF-8</encoding>
            </configuration>
        </plugin>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-war-plugin</artifactId>
            <version>3.2.1</version>
            <configuration>
                <warSourceDirectory>src/main/webapp</warSourceDirectory>
                <failOnMissingWebXml>true</failOnMissingWebXml>
            </configuration>
        </plugin>
    </plugins>
</build>
<!-- 여기까지 -->

<modules>
    ...
</modules>
</project>

```

자식의 pom.xml 내용 중 겹치는 부분은 부모에게 써줌

- helloweb/pom.xml 내용 추가

```
<project>
```

```

...
</packaging>

<!-- 아래 코드 추가-->
<build>
    <finalName>helloworld</finalName>

    <plugins>
        <plugin>
            <groupId>org.codehaus.mojo</groupId>
            <artifactId>tomcat-maven-plugin</artifactId>
            <configuration>
                <url>http://127.0.0.1:8088/manager/text</url>
                <path>/helloworld</path>
                <username>admin</username>
                <password>manager</password>
            </configuration>
        </plugin>
    </plugins>
</build>
<!-- 여기까지 -->
</project>

```

## Web Programming #1

### 1. JSP 생성

src/main/webapp 우클릭 - New - JSP File > hello.jsp 생성

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <h1>Hello world!</h1>
</body>
</html>

```

#### 테스트

http://127.0.0.1:8080/helloworld/hello.jsp 접속

Hello World!가 출력되면 잘 된 것!!!!!!!!!!!!!!

## 2. 패키지 생성

src/main/java 우클릭 - New - Package

Name: helloweb

Finish

## 3. servlet 생성

helloweb 패키지 우클릭 - New - Servlet

**Class name:** HelloServlet

Next

**URL mappings:** /HelloServlet -> /hello 변경

Next

Constructors from superclass 체크 해제  
doGet 체크

Finish

```
package helloweb;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class HelloServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");

        response.setContentType("text/html; charset=utf-8");

        PrintWriter pw = response.getWriter();
        pw.print("<h1>Hello world!</h1>");
    }
}
```

<http://127.0.0.1:8080/helloweb/hello> 접속  
Hello World!가 출력되면 잘 된 것!!!!!!!!!!!!!!

#### 4. 테스트

`http://127.0.0.1:8080/helloweb/hello` 접속

Hello World!가 출력되면 잘 된 것!!!!!!!!!!!!!!

`http://127.0.0.1:8080/helloweb/hello` 접속

Hello World!가 출력되면 잘 된 것!!!!!!!!!!!!!!

## Jenkins

Git에 Push 하고.. 하세요 .....!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

### 1. Jenkins 접속

`http://127.0.0.1:8088/jenkins` 접속

**ID:** jenkins

**PW:** manager

### 2. 프로젝트 생성

새로운 Item > Freestyle project

Enter an item name : servlet-practices

소스 코드 관리

**Git**

Repository URL : <https://github.com/jooeui/servlet-practices.git>

빌드 환경

**Build**

Add build step > Invoke top-level Maven targets

Maven Version : maven 선택 / Goals : clean package tomcat:redeploy

### 3. Build

Build Now 선택