

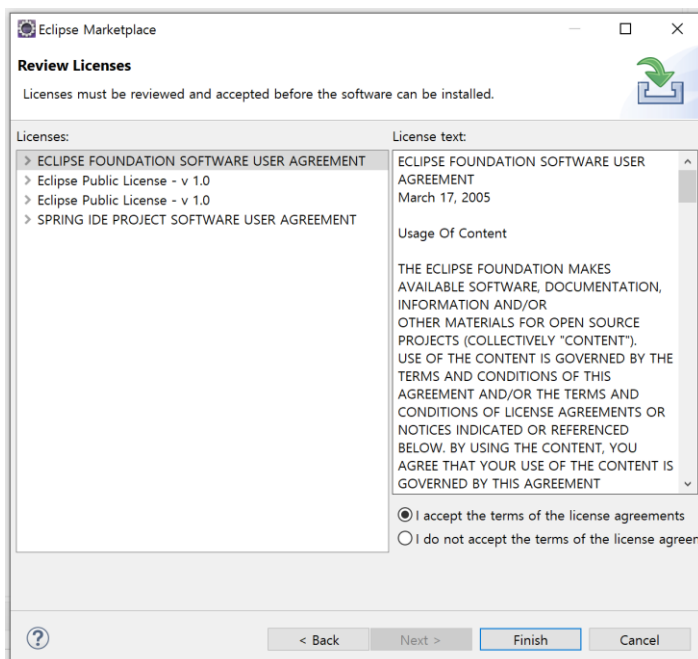
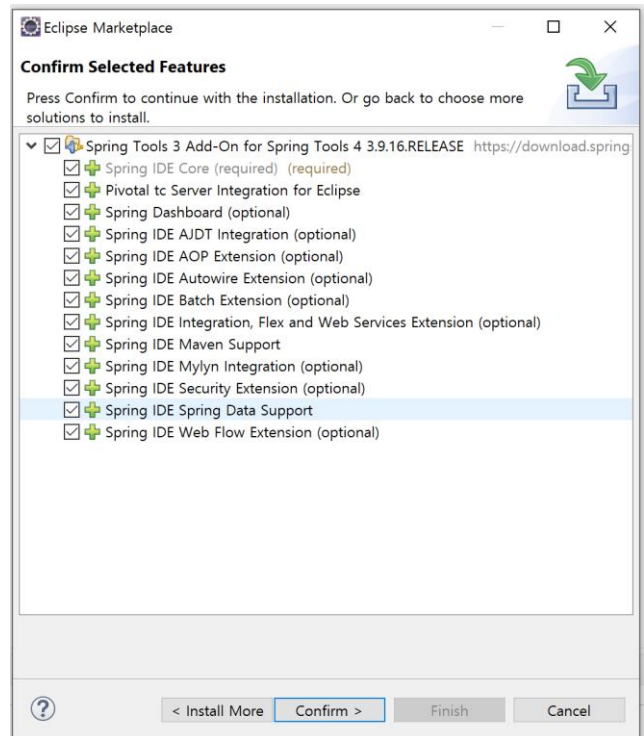
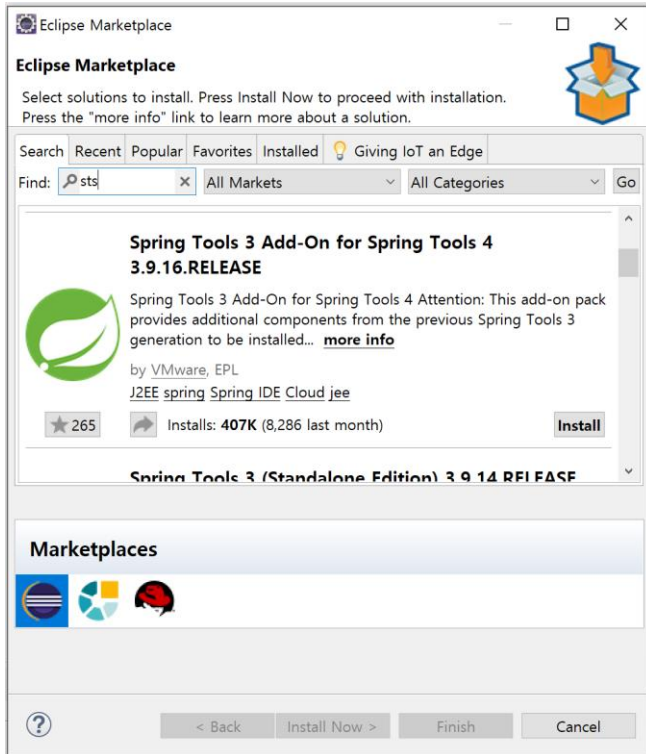
# Spring 웹 개발

## 1 장 Spring 웹 개발환경

### 1. 이클립스에 Spring Tools 4 프로그램 설치

Help -> Eclipse Marketplace 에서 sts 검색

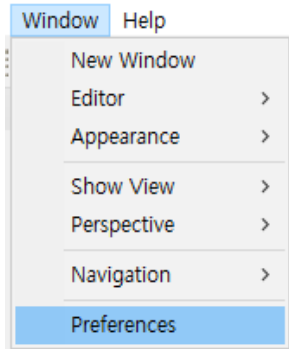
Spring Tools 3 Add-On for Spring Tools 4 3.9.16.RELEASE 설치



## 2. Spring 환경 설정

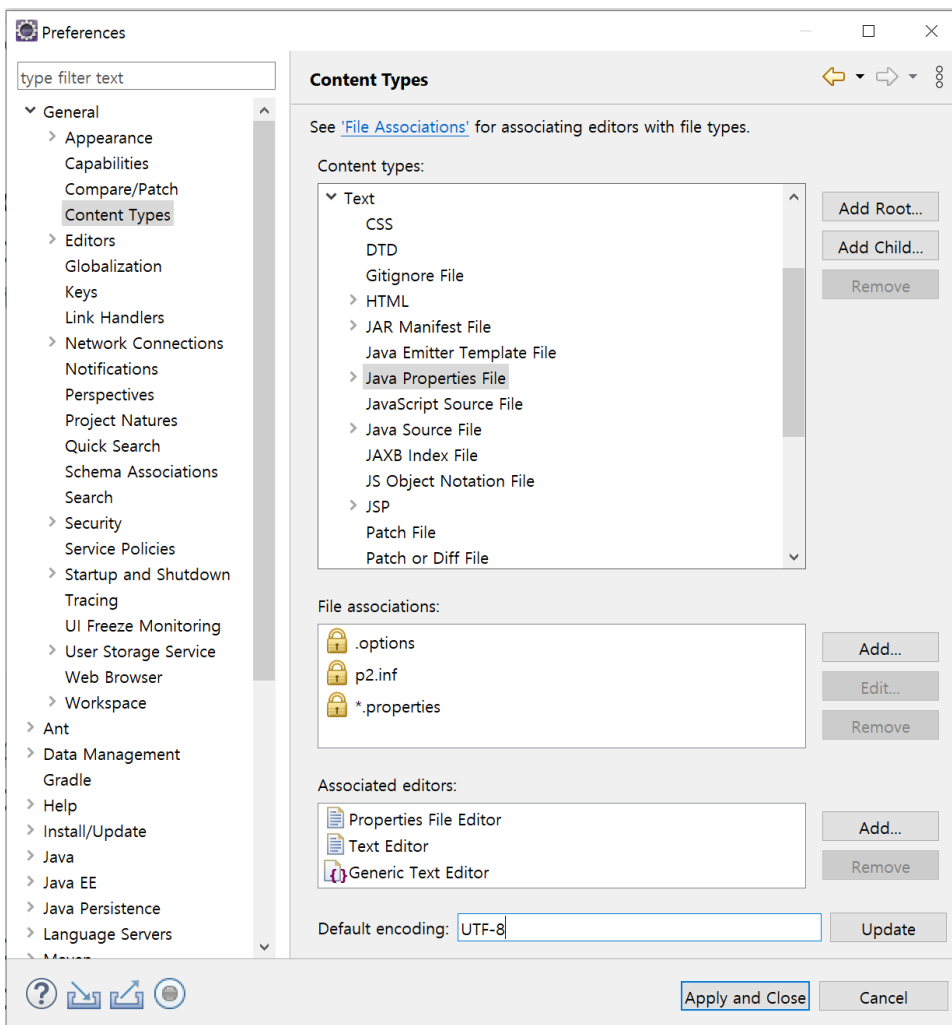
### 2.1 UTF-8 설정

툴 상단 메뉴의 Window -> Preferences 선택



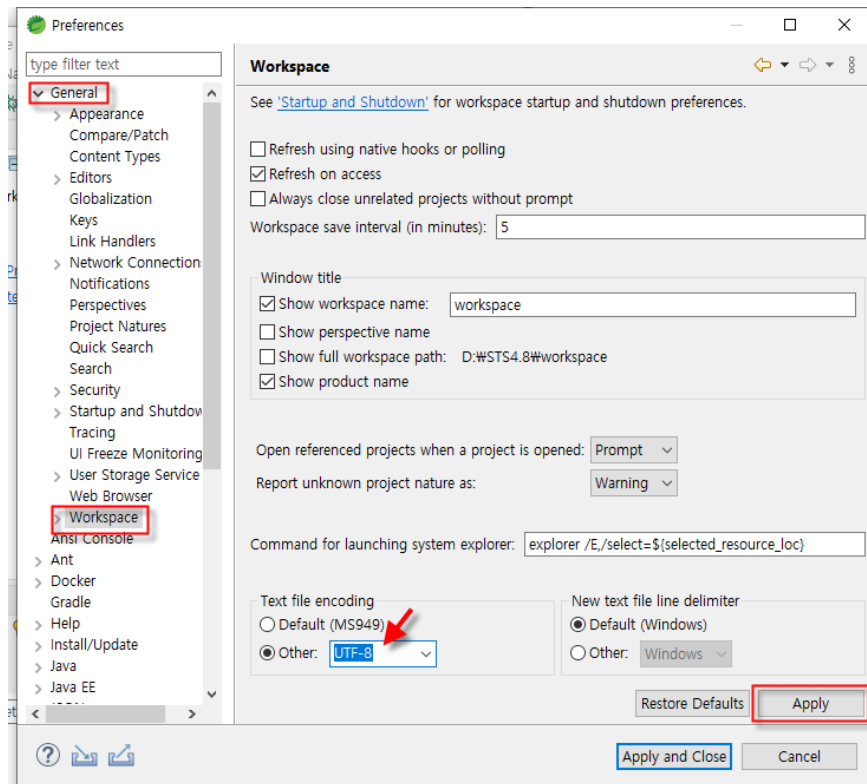
General -> Content Types -> Text -> Java Properties File 선택

Default encoding : UTF-8 로 변경 후 Update 버튼 클릭



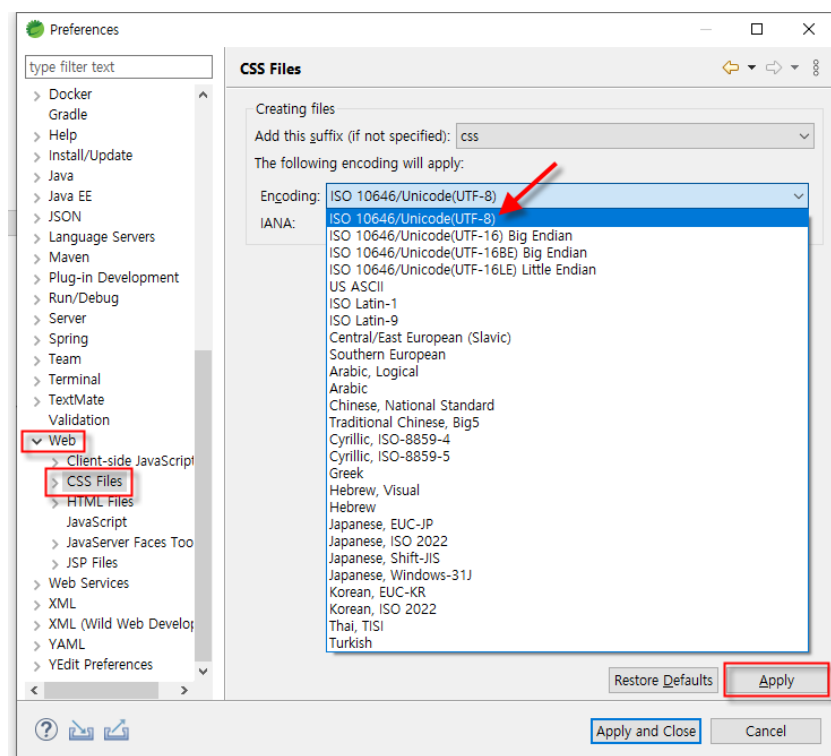
General -> Workspace 선택

Text file encoding 에서 Other 을 선택하고 UTF-8 을 선택한다. 그리고 Apply 버튼 클릭한다.



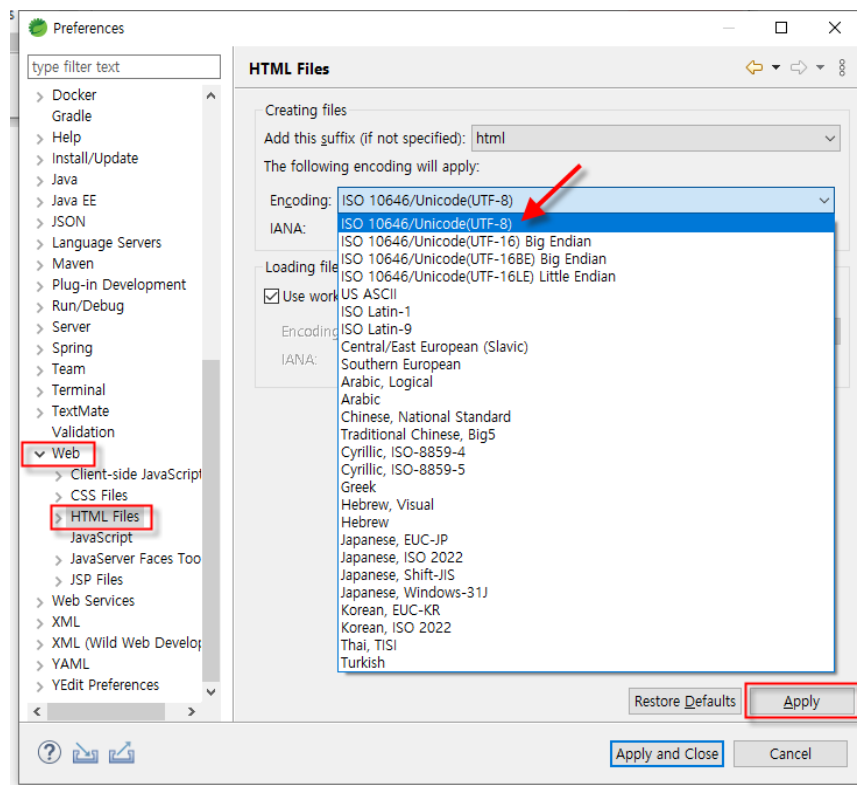
좌측 분류에서 Web -> CSS Files 선택

Encoding 에서 ISO 10646/Unicode(UTF-8) 선택하고 Apply 버튼 클릭



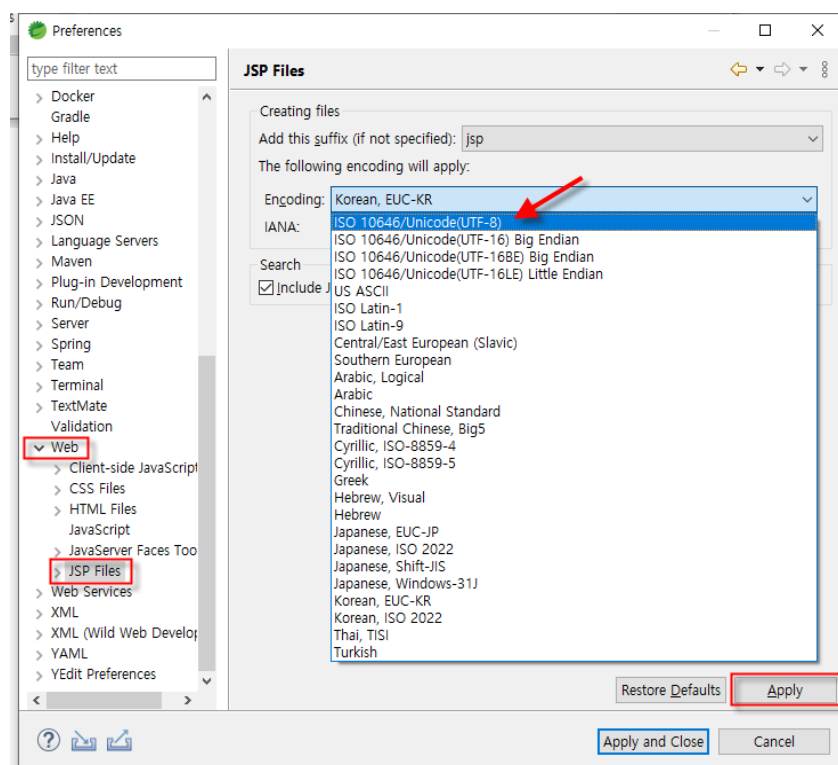
좌측 분류에서 Web -> HTML Files 선택

Encoding 에서 ISO 10646/Unicode(UTF-8) 선택하고 Apply 버튼 클릭



좌측 분류에서 Web -> JSP Files 선택

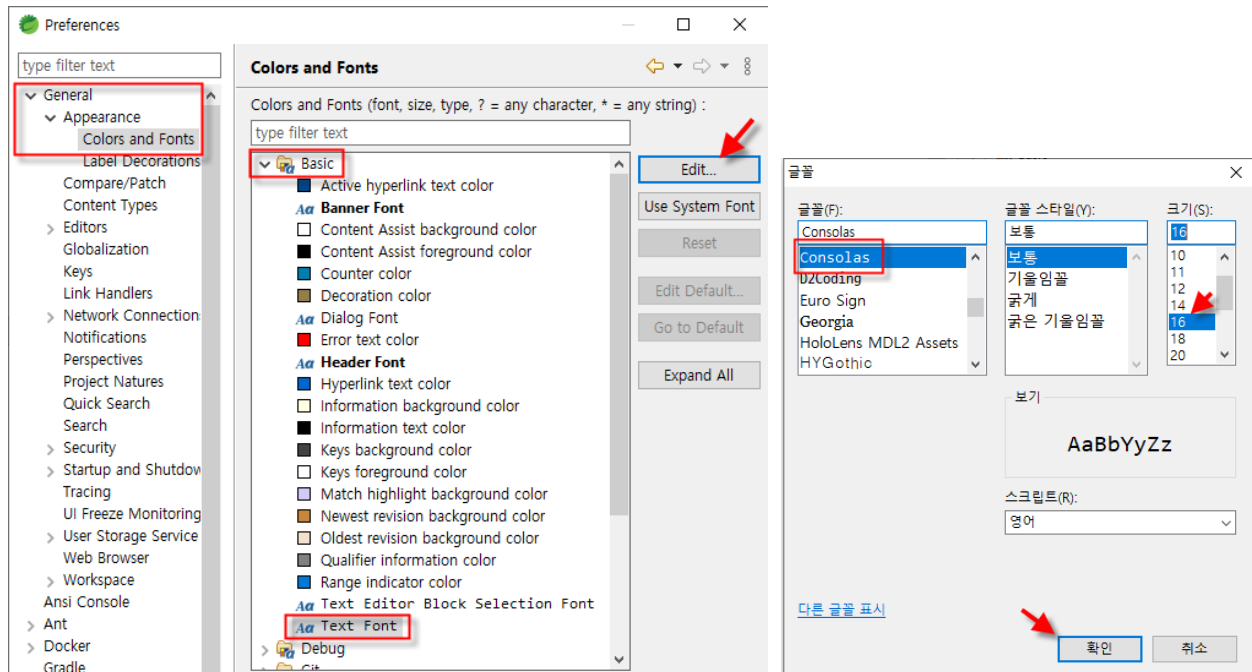
Encoding 에서 ISO 10646/Unicode(UTF-8) 선택하고 Apply 버튼 클릭



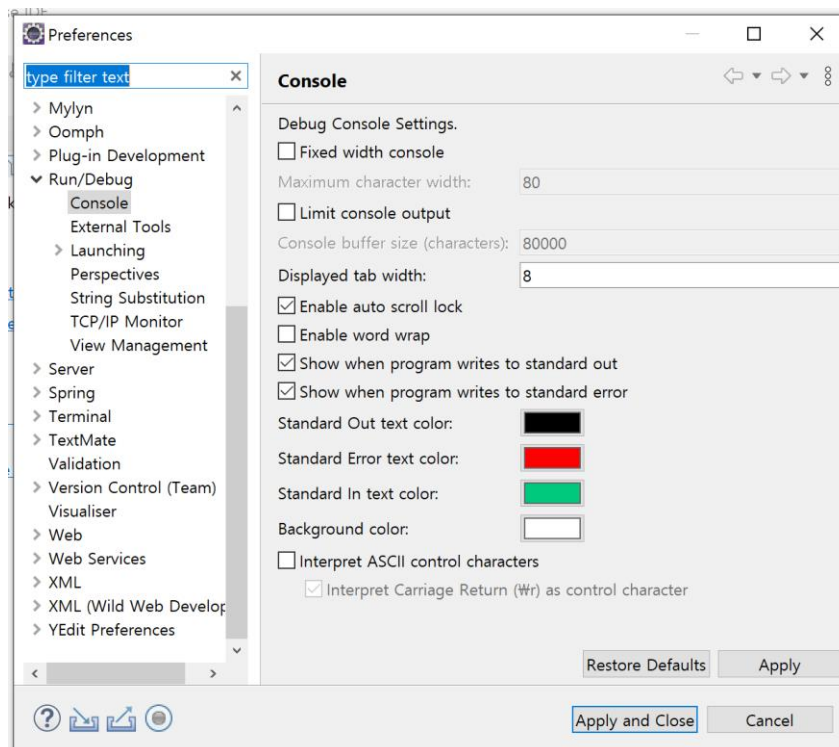
## 2.2 글꼴 Size 변경 및 console 출력 라인 크기 해제

General -> Appearance -> Colors and Fonts 선택

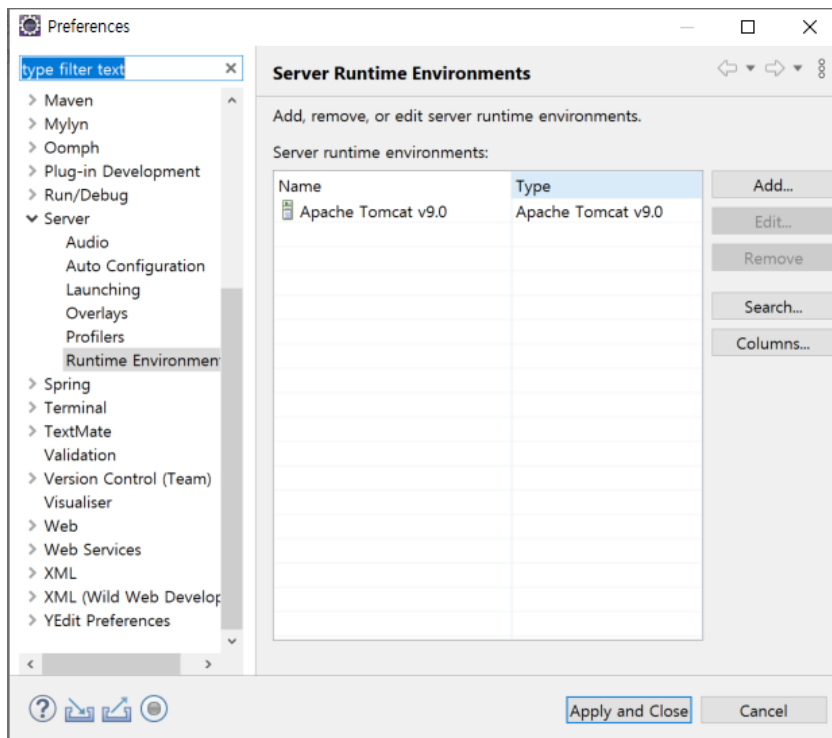
Basic -> Text Fonts 선택 후 Edit... 버튼 클릭하고 크기를 14 또는 16 으로 변경



Run/Debug -> Console : Limit console output 체크 해제



## 2.3 Apache Tomcat 웹 서버 실행 환경 설정

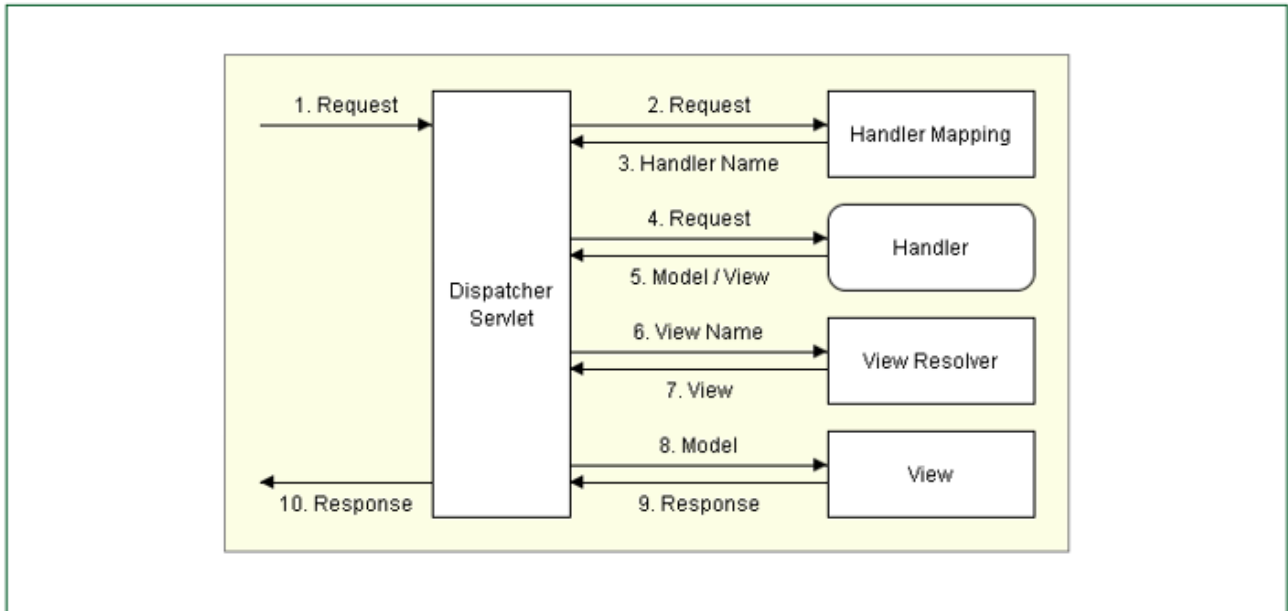


## 2 장 프로젝트 생성

### 1. Spring MVC 구조

스프링 MVC 는 DispatcherServlet, View Resolver, Handler, View 등으로 구성되어 있다.

스프링 MVC 요청 처리 흐름도



단계별 설명

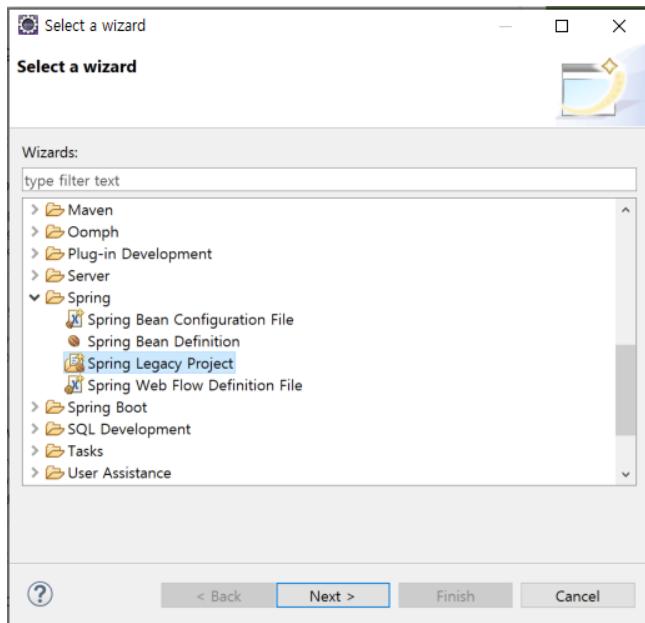
- ① 클라이언트(웹 브라우저)의 요청(Request)을 DispatcherServlet 이 받는다.
- ② 받은 요청을 처리할 수 있는 Handler 의 이름을 HandlerMapping 에게 물어본다.
- ③ HandlerMapping 은 요청 URL 을 비롯한 여러 요청 정보를 가지고 Handler 를 판단한다.
- ④ DispatcherServlet 은 HandlerMapping 으로부터 선택받은 Handler 에게 요청을 보낸다.
- ⑤ Handler 는 요청을 적절히 처리하고 응답에서 렌더링해야 하는 View Name 을 판단해서 DispatcherServlet 에 전송한다.
- ⑥ DispatcherServlet 은 논리적인 View Name 을 View Resolver 에 전달한다.
- ⑦ View Resolver 는 응답이 필요한 View 를 생성하여 DispatcherServlet 에 보낸다.
- ⑧ DispatcherServlet 은 해당하는 View 에 Model 을 전달한다.
- ⑨ View 는 Model 을 참조해서 응답을 생성한다.
- ⑩ DispatcherServlet 은 생성된 응답(Response)을 클라이언트(웹 브라우저)에 반환한다.

dispatcher : 운영관리자, resolver : 해결자

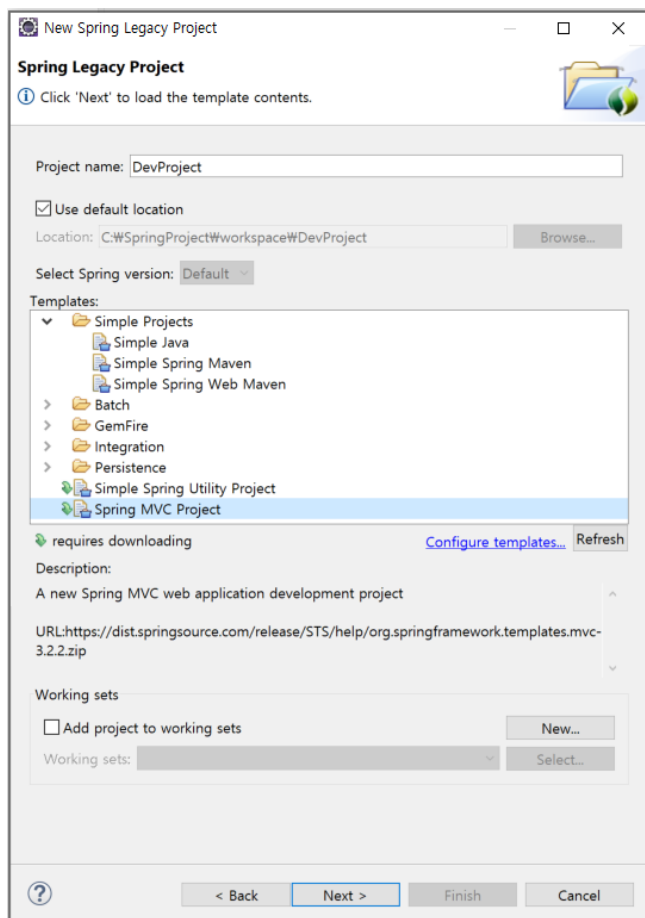
## 2. Spring MVC 프로젝트 생성

### 2.1 File -> New -> Other 선택

Select a wizard 대화창에서 Spring -> Spring Legacy Project 선택 -> Next



### 2.2 Spring Legacy Project 대화창이 열린다. Spring MVC Project 선택 -> Next

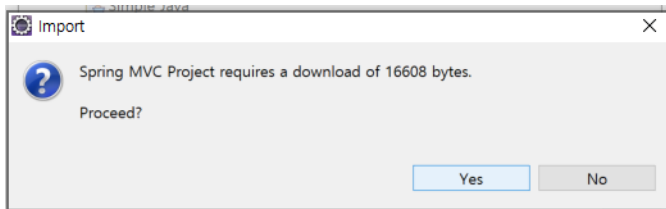




### 2.3 프로젝트 생성에 필요한 정보를 입력

항목	설명	설정값
Project Name	생성할 프로젝트의 이름	DevProject
Templates	프로젝트 템플릿 선택	Spring MVC Project

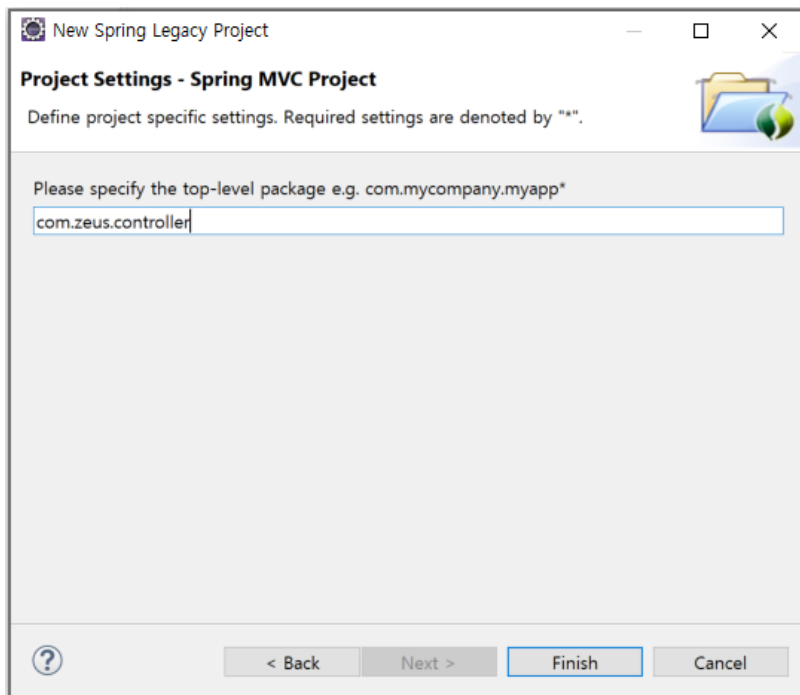
Next 을 선택한다.



Yes 선택

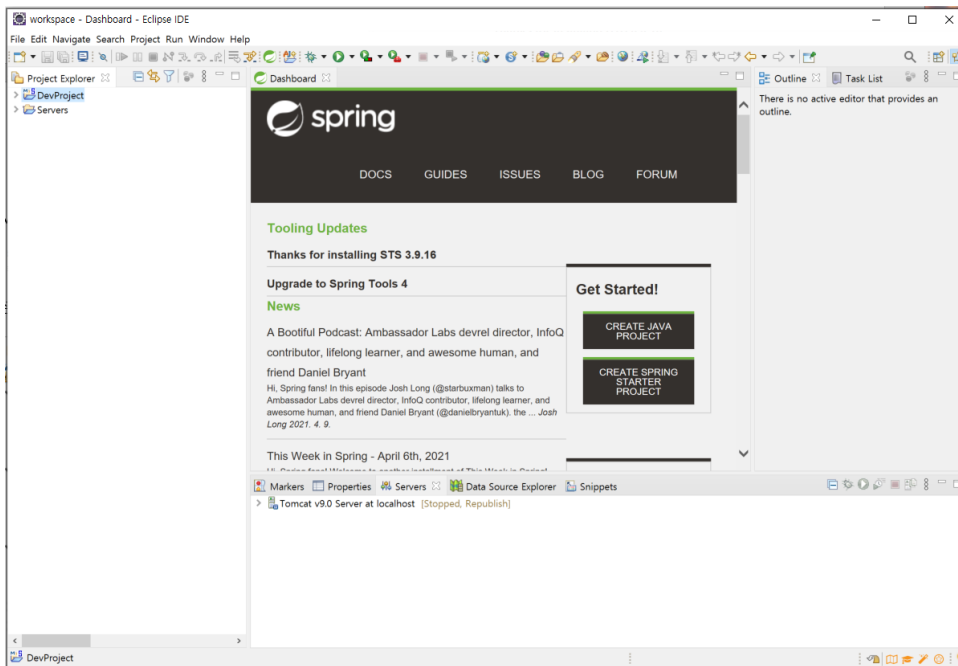
### 2.4 패키지 입력

com.zeus.controller

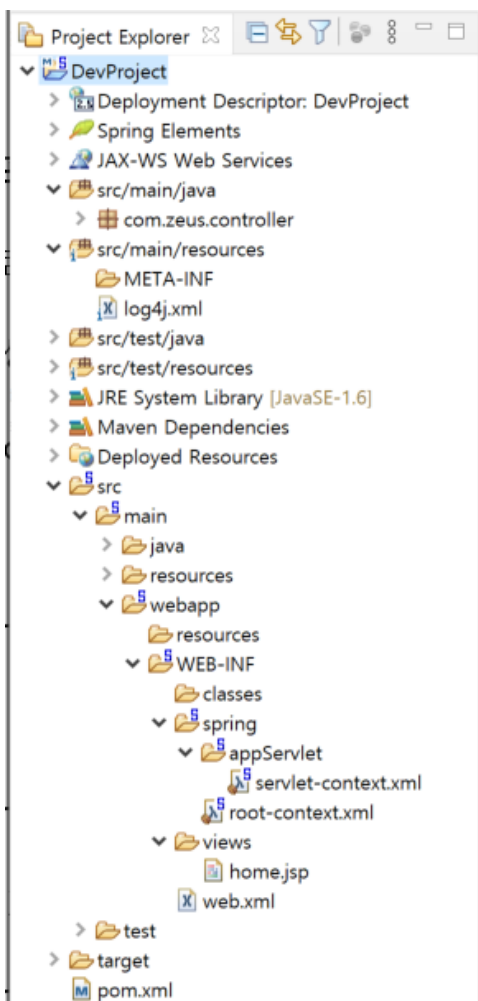


Finish 선택하면 maven 이 필요한 라이브러리를 다운로드 한다(시간이 걸린다.).

### 3. MVC 프로젝트



### 생성된 프로젝트 파일 구성



#### 4. 프로젝트 파일 구성 요소 설명

프로젝트를 새로 생성하면 관련 파일들과 폴더들이 자동으로 생성된다.

[구성 요소]

- pom.xml
  - : 메이븐 프로젝트 빌드 파일
- src/main/java
  - : 자바 소스 경로
- src/main/resources
  - : XML 파일, 프로퍼티(properties) 파일, 리소스 파일 경로
- src/.../WEB-INF/web.xml
  - : 웹 컨테이너 설정 파일
- src/.../WEB-INF/spring/root-context.xml
  - : 스프링 설정 파일
- src/.../WEB-INF/spring/appServlet/servlet-context.xml
  - : 스프링 웹 설정 파일
- src/.../WEB-INF/views
  - : 뷰 파일 경로

[구현 소스]

프로젝트 빌드 파일

pom.xml
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;project xmlns="http://maven.apache.org/POM/4.0.0"   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0   http://maven.apache.org/maven-v4_0_0.xsd"&gt;   &lt;modelVersion&gt;4.0.0&lt;/modelVersion&gt;   &lt;groupId&gt;com.zeus&lt;/groupId&gt;   &lt;artifactId&gt;controller&lt;/artifactId&gt;   &lt;name&gt;DevProject&lt;/name&gt;   &lt;packaging&gt;war&lt;/packaging&gt;   &lt;version&gt;1.0.0-BUILD-SNAPSHOT&lt;/version&gt;   &lt;properties&gt;</pre>

```

<java-version>1.6</java-version>
<org.springframework-version>3.1.1.RELEASE</org.springframework-version>
<org.aspectj-version>1.6.10</org.aspectj-version>
<org.slf4j-version>1.6.6</org.slf4j-version>
</properties>
<dependencies>
  <!-- Spring -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>${org.springframework-version}</version>
    <exclusions>
      <!-- Exclude Commons Logging in favor of SLF4j -->
      <exclusion>
        <groupId>commons-logging</groupId>
        <artifactId>commons-logging</artifactId>
      </exclusion>
    </exclusions>
  </dependency>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>${org.springframework-version}</version>
  </dependency>

  <!-- AspectJ -->
  <dependency>
    <groupId>org.aspectj</groupId>
    <artifactId>aspectjrt</artifactId>
    <version>${org.aspectj-version}</version>
  </dependency>

  <!-- Logging -->
  <dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-api</artifactId>
    <version>${org.slf4j-version}</version>
  </dependency>
  <dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>jcl-over-slf4j</artifactId>
    <version>${org.slf4j-version}</version>
    <scope>runtime</scope>
  </dependency>
  <dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-log4j12</artifactId>
    <version>${org.slf4j-version}</version>
    <scope>runtime</scope>
  </dependency>
  <dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.15</version>
    <exclusions>
      <exclusion>
        <groupId>javax.mail</groupId>
        <artifactId>mail</artifactId>
      </exclusion>
    </exclusions>
  </dependency>
</dependencies>

```

```

        <exclusion>
            <groupId>javax.jms</groupId>
            <artifactId>jms</artifactId>
        </exclusion>
        <exclusion>
            <groupId>com.sun.jdmk</groupId>
            <artifactId>jmxtools</artifactId>
        </exclusion>
        <exclusion>
            <groupId>com.sun.jmx</groupId>
            <artifactId>jmxri</artifactId>
        </exclusion>
    </exclusions>
    <scope>runtime</scope>
</dependency>

<!-- @Inject -->
<dependency>
    <groupId>javax.inject</groupId>
    <artifactId>javax.inject</artifactId>
    <version>1</version>
</dependency>

<!-- Servlet -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>servlet-api</artifactId>
    <version>2.5</version>
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>jsp-api</artifactId>
    <version>2.1</version>
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
</dependency>

<!-- Test -->
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.7</version>
    <scope>test</scope>
</dependency>
</dependencies>
<build>
    <plugins>
        <plugin>
            <artifactId>maven-eclipse-plugin</artifactId>
            <version>2.9</version>
            <configuration>
                <additionalProjectnatures>
<projectnature>org.springframework.ide.eclipse.core.springnature</projectnature>

```

```

        </additionalProjectnatures>
        <additionalBuildcommands>

<buildcommand>org.springframework.ide.eclipse.core.springbuilder</buildcommand>
        </additionalBuildcommands>
        <downloadSources>true</downloadSources>
        <downloadJavadocs>true</downloadJavadocs>
    </configuration>
</plugin>
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>2.5.1</version>
    <configuration>
        <source>1.6</source>
        <target>1.6</target>
        <compilerArgument>-Xlint:all</compilerArgument>
        <showWarnings>true</showWarnings>
        <showDeprecation>true</showDeprecation>
    </configuration>
</plugin>
<plugin>
    <groupId>org.codehaus.mojo</groupId>
    <artifactId>exec-maven-plugin</artifactId>
    <version>1.2.1</version>
    <configuration>
        <mainClass>org.test.int1.Main</mainClass>
    </configuration>
</plugin>
</plugins>
</build>
</project>

```

## 설정 파일

- 웹 컨테이너 설정

❗19      <servlet-name>appServlet</servlet-name>

발생하면 다음과 같이 수정한다.(java 를 대문자로)

xsi:schemaLocation="http://JAVA.sun.com/xml/ns/javaee

### web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://JAVA.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">

    <!-- The definition of the Root Spring Container shared by all Servlets and
Filters -->
    <context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>/WEB-INF/spring/root-context.xml</param-value>
    </context-param>

```

```

        <!-- Creates the Spring Container shared by all Servlets and Filters -->
        <listener>
            <listener-
class>org.springframework.web.context.ContextLoaderListener</listener-class>
            </listener>

        <!-- Processes application requests -->
        <servlet>
            <servlet-name>appServlet</servlet-name>
            <servlet-
class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
            <init-param>
                <param-name>contextConfigLocation</param-name>
                <param-value>/WEB-INF/spring/appServlet/servlet-
context.xml</param-value>
            </init-param>
            <load-on-startup>1</load-on-startup>
        </servlet>

        <servlet-mapping>
            <servlet-name>appServlet</servlet-name>
            <url-pattern>/</url-pattern>
        </servlet-mapping>

    </web-app>

```

#### · 스프링 설정

##### root-context.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.springframework.org/schema/beans
https://www.springframework.org/schema/beans/spring-beans.xsd">

    <!-- Root Context: defines shared resources visible to all other web components
-->

</beans>

```

#### · 스프링 웹 설정

##### servlet-context.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns="http://www.springframework.org/schema/mvc"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:beans="http://www.springframework.org/schema/beans"
        xmlns:context="http://www.springframework.org/schema/context"
        xsi:schemaLocation="http://www.springframework.org/schema/mvc
https://www.springframework.org/schema/mvc/spring-mvc.xsd
http://www.springframework.org/schema/beans
https://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
https://www.springframework.org/schema/context/spring-context.xsd">

```

```

    <!-- DispatcherServlet Context: defines this servlet's request-processing
    infrastructure -->

    <!-- Enables the Spring MVC @Controller programming model -->
    <annotation-driven />

    <!-- Handles HTTP GET requests for /resources/** by efficiently serving up
    static resources in the ${webappRoot}/resources directory -->
    <resources mapping="/resources/**" location="/resources/" />

    <!-- Resolves views selected for rendering by @Controllers to .jsp resources in
    the /WEB-INF/views directory -->
    <beans:bean
    class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <beans:property name="prefix" value="/WEB-INF/views/" />
        <beans:property name="suffix" value=".jsp" />
    </beans:bean>

    <context:component-scan base-package="com.zeus.controller" />

</beans:beans>

```

자바 소스 파일

· com.zeus.controller 패키지

#### HomeController.java

```

package com.zeus.controller;

import java.text.DateFormat;
import java.util.Date;
import java.util.Locale;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

/**
 * Handles requests for the application home page.
 */
@Controller
public class HomeController {

    private static final Logger logger =
    LoggerFactory.getLogger(HomeController.class);

    /**
     * Simply selects the home view to render by returning its name.
     */
    @RequestMapping(value = "/", method = RequestMethod.GET)
    public String home(Locale locale, Model model) {
        logger.info("Welcome home! The client locale is {}", locale);
    }
}

```



```

        Date date = new Date();
        DateFormat dateFormat = DateFormat.getDateInstance(DateFormat.LONG,
DateFormat.LONG, locale);

        String formattedDate = dateFormat.format(date);

        model.addAttribute("serverTime", formattedDate );

        return "home";
    }
}

```

뷰 파일

· WEB-INF/views

#### home.jsp

```

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ page session="false" %>
<html>
<head>
    <title>Home</title>
</head>
<body>
<h1>
    Hello world!
</h1>

<P> The time on the server is ${serverTime}. </P>
</body>
</html>

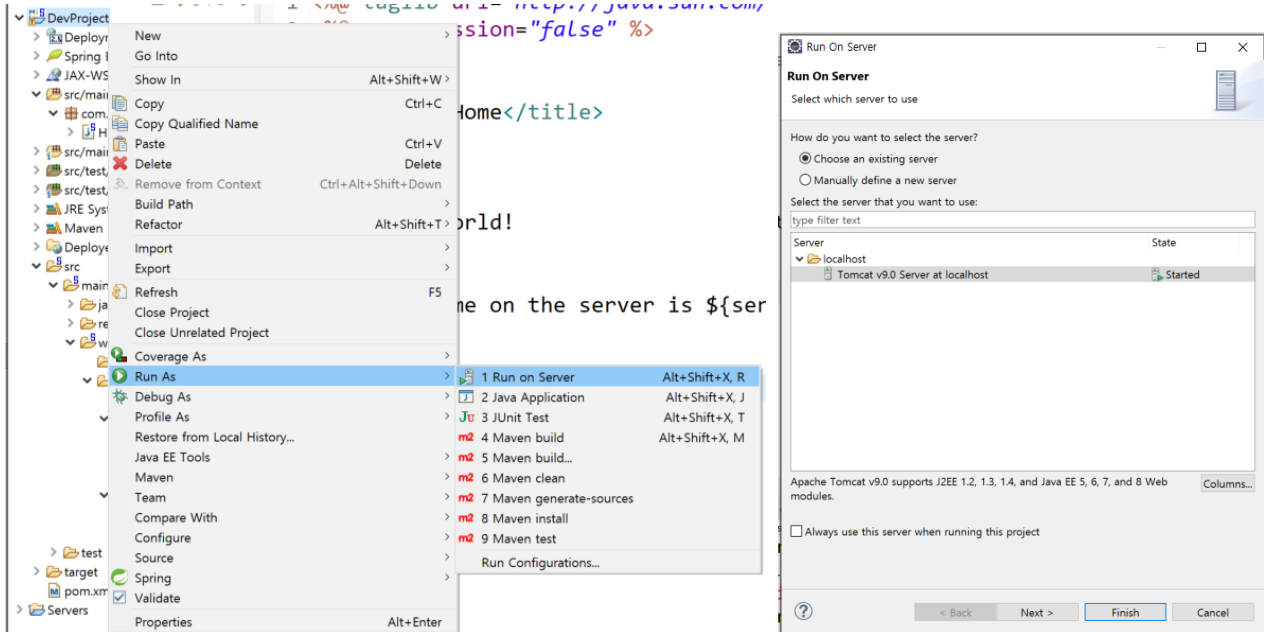
```

## 4. 프로젝트 실행

4.1 아래와 같이 프로젝트를 선택한 후 마우스 오른쪽 버튼을 클릭

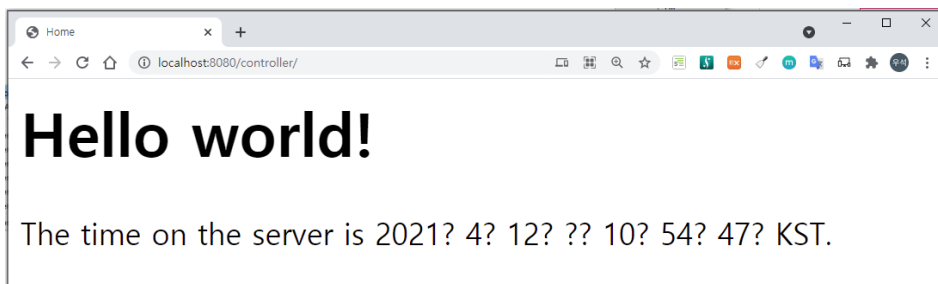
Run As -> Run on Server 선택

Finish 클릭



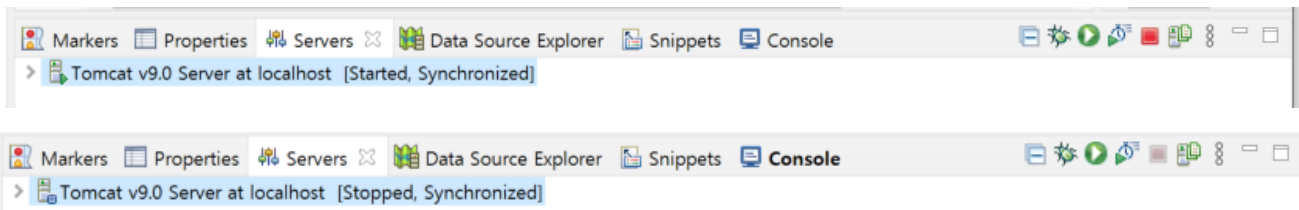
4.2 크롬 브라우저에서 실행 결과

http://localhost:8080/controller/

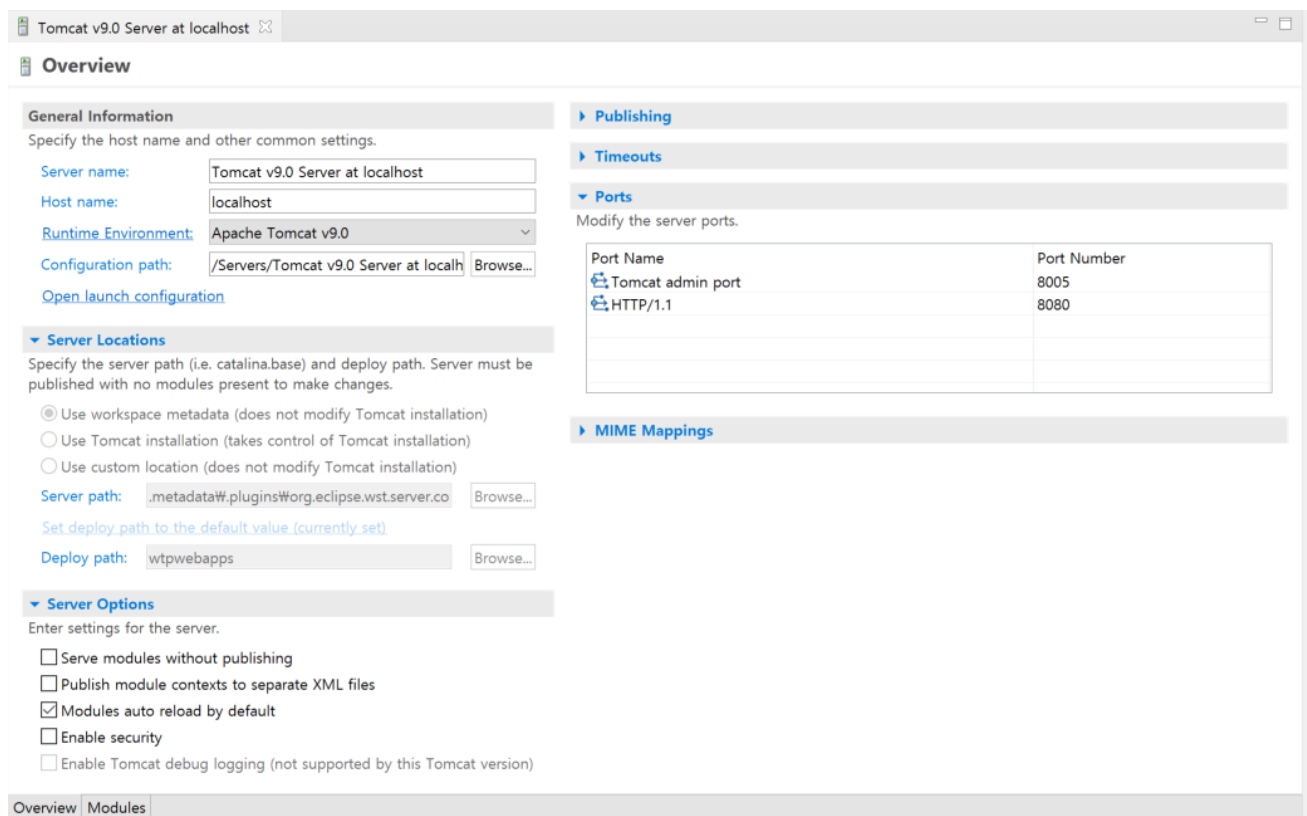


## 5. 웹 컨텍스트 변경

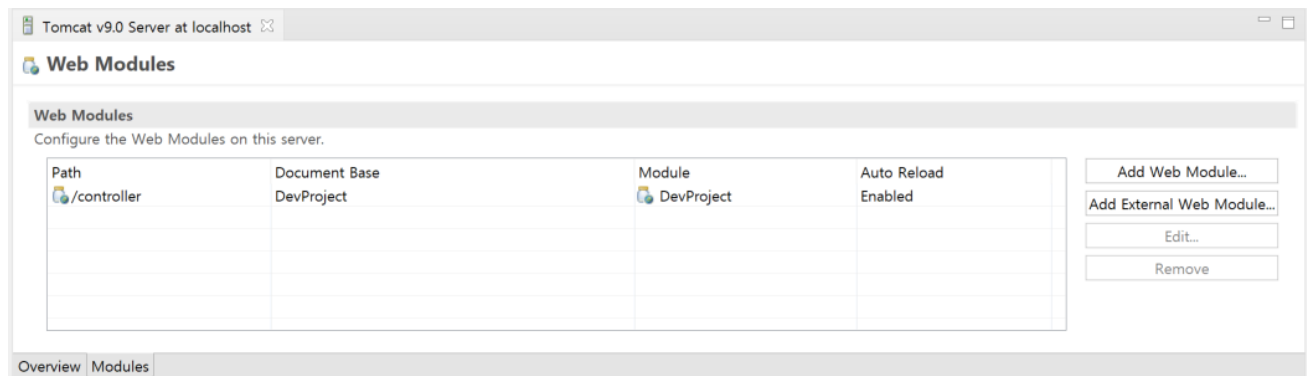
### 5.1 톰캣 서버를 중지한다.



### 5.2 Servers 탭에서 톰캣 서버를 더블 클릭하면 톰캣 서버 환경 설정 화면이 나온다.

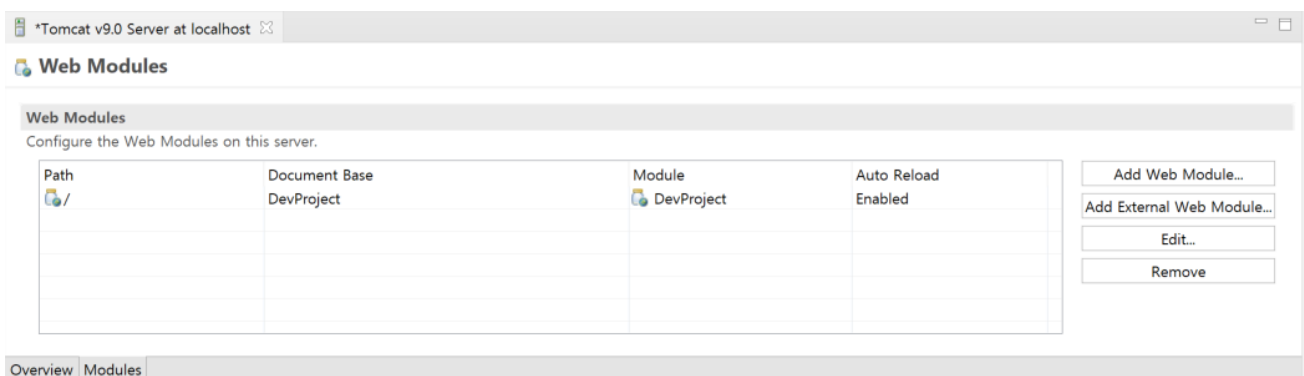
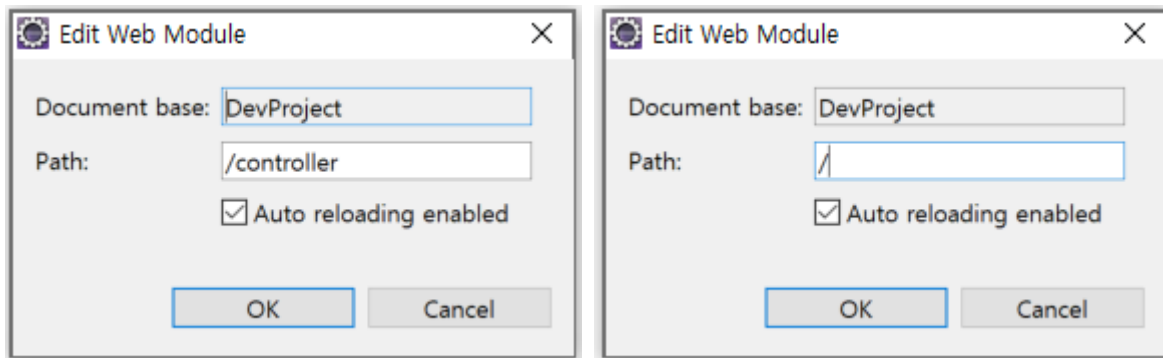


### 5.3 Modules 메뉴를 선택한다.



5.4 설정을 변경할 웹 모듈을 선택하고 Edit... 버튼을 클릭한다.

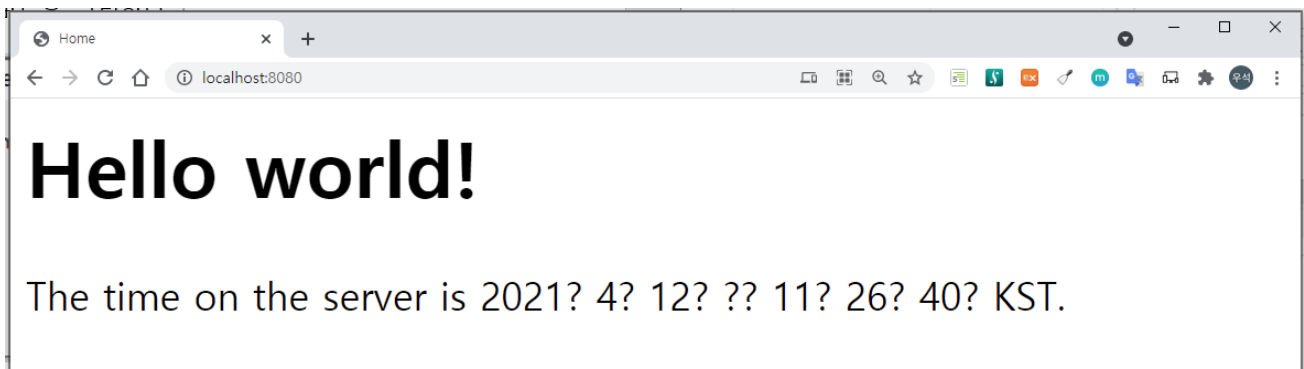
웹 컨텍스트 경로를 `"/controller"`에서 `"/"`로 변경하고 OK 버튼을 클릭한다.



5.5 Ctrl+S 를 눌러 저장한다.

톰캣 서버를 재시작 한다. 아래의 URL 주소로 요청한다.

`http://localhost:8080/`



## 6. 버전 변경

JDK 1.8 버전, 서블릿 3.1 버전, 스프링 프레임워크 5.0.7.RELEASE 버전을 사용한다.

### 6.1 환경설정

#### • 버전 변경

##### pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  https://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.zeus</groupId>
  <artifactId>controller</artifactId>
  <name>DevProject</name>
  <packaging>war</packaging>
  <version>1.0.0-BUILD-SNAPSHOT</version>
  <properties>
    <!-- 자바 버전을 1.6에서 1.8로 변경 -->
    <java-version>1.8</java-version>
    <!-- 스프링 버전을 3.1.1.RELEASE에서 5.0.7.RELEASE로 변경 -->
    <org.springframework-version>5.0.7.RELEASE</org.springframework-version>
    <org.aspectj-version>1.6.10</org.aspectj-version>
    <org.slf4j-version>1.6.6</org.slf4j-version>
  </properties>
  <dependencies>
    <!-- Spring -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>${org.springframework-version}</version>
      <exclusions>
        <!-- Exclude Commons Logging in favor of SLF4j -->
        <exclusion>
          <groupId>commons-logging</groupId>
          <artifactId>commons-logging</artifactId>
        </exclusion>
      </exclusions>
    </dependency>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>${org.springframework-version}</version>
    </dependency>

    <!-- AspectJ -->
    <dependency>
      <groupId>org.aspectj</groupId>
      <artifactId>aspectjrt</artifactId>
      <version>${org.aspectj-version}</version>
    </dependency>
```

```

<!-- Logging -->
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-api</artifactId>
    <version>${org.slf4j-version}</version>
</dependency>
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>jcl-over-slf4j</artifactId>
    <version>${org.slf4j-version}</version>
    <scope>runtime</scope>
</dependency>
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-log4j12</artifactId>
    <version>${org.slf4j-version}</version>
    <scope>runtime</scope>
</dependency>
<dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.15</version>
    <exclusions>
        <exclusion>
            <groupId>javax.mail</groupId>
            <artifactId>mail</artifactId>
        </exclusion>
        <exclusion>
            <groupId>javax.jms</groupId>
            <artifactId>jms</artifactId>
        </exclusion>
        <exclusion>
            <groupId>com.sun.jdmk</groupId>
            <artifactId>jmxtools</artifactId>
        </exclusion>
        <exclusion>
            <groupId>com.sun.jmx</groupId>
            <artifactId>jmxri</artifactId>
        </exclusion>
    </exclusions>
    <scope>runtime</scope>
</dependency>

<!-- @Inject -->
<dependency>
    <groupId>javax.inject</groupId>
    <artifactId>javax.inject</artifactId>
    <version>1</version>
</dependency>

<!-- Servlet -->
<!-- 서블릿 버전을 2.5에서 3.1.0로 변경 -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>3.1.0</version>
    <scope>provided</scope>
</dependency>

```

```

        <dependency>
            <groupId>javax.servlet.jsp</groupId>
            <artifactId>jsp-api</artifactId>
            <version>2.1</version>
            <scope>provided</scope>
        </dependency>
        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>jstl</artifactId>
            <version>1.2</version>
        </dependency>

        <!-- Test -->
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.7</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
    <build>
        <plugins>
            <plugin>
                <artifactId>maven-eclipse-plugin</artifactId>
                <version>2.9</version>
                <configuration>
                    <additionalProjectnatures>

<projectnature>org.springframework.ide.eclipse.core.springnature</projectnature>
                    </additionalProjectnatures>
                    <additionalBuildcommands>

<buildcommand>org.springframework.ide.eclipse.core.springbuilder</buildcommand>
                    </additionalBuildcommands>
                    <downloadSources>true</downloadSources>
                    <downloadJavadocs>true</downloadJavadocs>
                </configuration>
            </plugin>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>2.5.1</version>
                <configuration>
                    <!-- 자바 버전을 1.6에서 1.8로 변경 -->
                    <source>1.8</source>
                    <target>1.8</target>
                    <compilerArgument>-Xlint:all</compilerArgument>
                    <showWarnings>true</showWarnings>
                    <showDeprecation>true</showDeprecation>
                </configuration>
            </plugin>
            <plugin>
                <groupId>org.codehaus.mojo</groupId>
                <artifactId>exec-maven-plugin</artifactId>
                <version>1.2.1</version>
                <configuration>
                    <mainClass>org.test.int1.Main</mainClass>
                </configuration>
            </plugin>
        </plugins>
    </build>

```

```

    </plugins>
  </build>
</project>

```

## • 웹 컨테이너 설정

서블릿 버전을 2.5 에서 3.1 로 변경됨에 따라 <web-app> 속성 수정

### /WEB-INF/web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
https://java.sun.com/xml/ns/javaee/web-app_3_1.xsd">

  <!-- The definition of the Root Spring Container shared by all Servlets and
Filters -->
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/spring/root-context.xml</param-value>
  </context-param>

  <!-- Creates the Spring Container shared by all Servlets and Filters -->
  <listener>
    <listener-
class>org.springframework.web.context.ContextLoaderListener</listener-class>
  </listener>

  <!-- Processes application requests -->
  <servlet>
    <servlet-name>appServlet</servlet-name>
    <servlet-
class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <init-param>
      <param-name>contextConfigLocation</param-name>
      <param-value>/WEB-INF/spring/appServlet/servlet-
context.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <servlet-mapping>
    <servlet-name>appServlet</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>

</web-app>

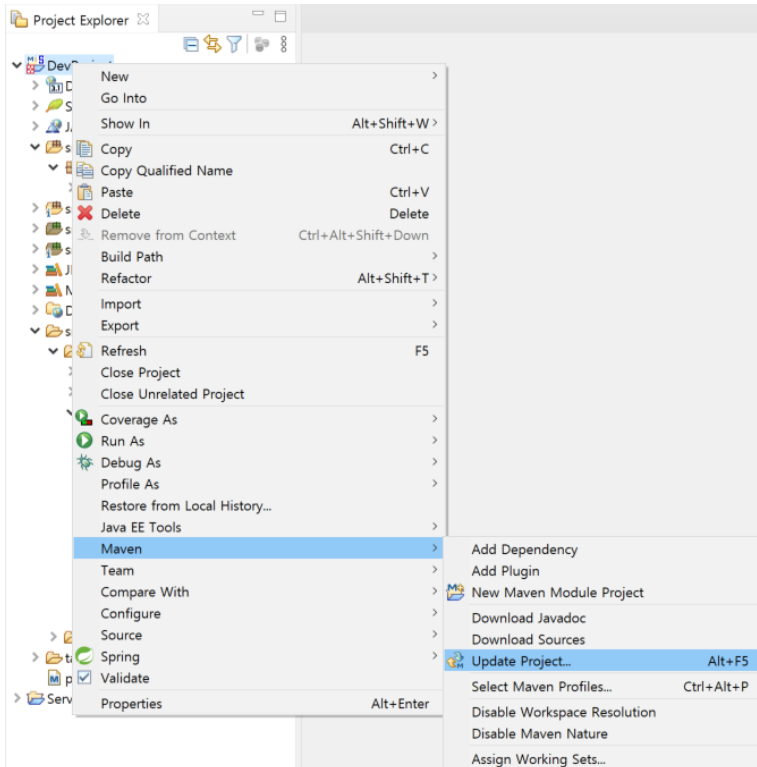
```



## 7. 프로젝트 업데이트

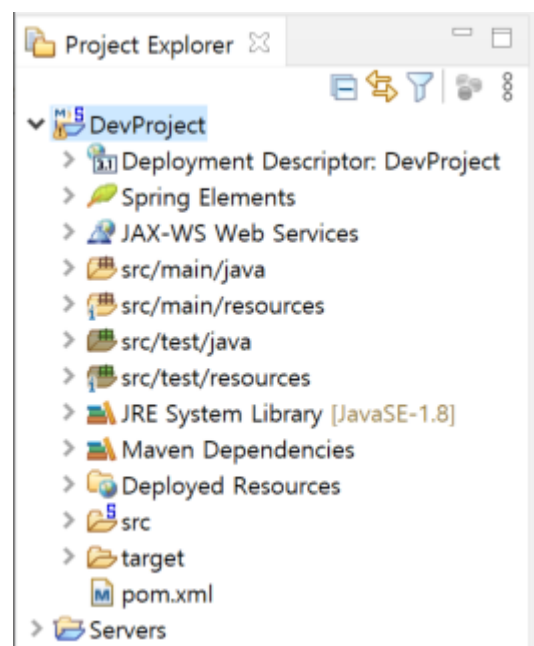
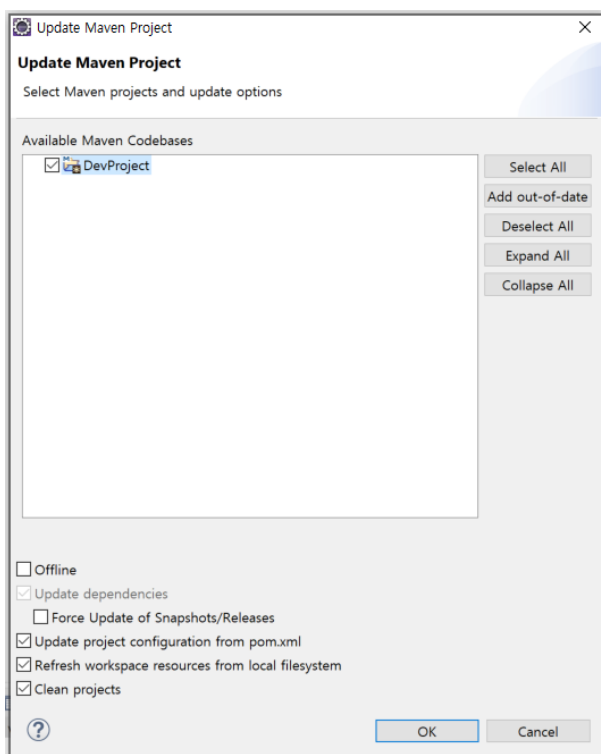
프로젝트 실행 환경이 JDK 1.8 버전으로 변경되었으므로 프로젝트를 업데이트해야 한다.

프로젝트 명 마우스 오른쪽 버튼 클릭 Maven -> Update Project... 를 선택한다.



OK 버튼을 클릭한다.

프로젝트가 업데이트 되었는지 확인한다.



## 8. UTF-8 처리 필터 등록

한글을 처리하는 필터를 등록한다.

### 8.1 환경 설정

• 웹 컨테이너 설정

#### /WEB-INF/web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://JAVA.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_1.xsd">

  <!-- The definition of the Root Spring Container shared by all Servlets and
Filters -->
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/spring/root-context.xml</param-value>
  </context-param>

  <!-- Creates the Spring Container shared by all Servlets and Filters -->
  <listener>
    <listener-
class>org.springframework.web.context.ContextLoaderListener</listener-class>
  </listener>

  <!-- Processes application requests -->
  <servlet>
    <servlet-name>appServlet</servlet-name>
    <servlet-
class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <init-param>
      <param-name>contextConfigLocation</param-name>
      <param-value>/WEB-INF/spring/appServlet/servlet-
context.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <servlet-mapping>
    <servlet-name>appServlet</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>

  <!-- 한글 처리를 위한 UTF-8 필터 등록 -->
  <filter>
    <filter-name>encodingFilter</filter-name>
    <filter-
class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
    <init-param>
      <param-name>encoding</param-name>
      <param-value>UTF-8</param-value>
    </init-param>
  </filter>
```

```

    <filter-mapping>
        <filter-name>encodingFilter</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>

</web-app>

```

## 9. 뷰 콘텐츠 유형 지정

뷰(view)의 MIME 타입과 문자 코드를 지정한다.

### 9.1 콘텐츠 유형 지정

• MIME 타입과 문자 코드를 지정

MIME 타입은 "text/html"로 지정하고 문자 코드는 "UTF-8"로 지정한다.

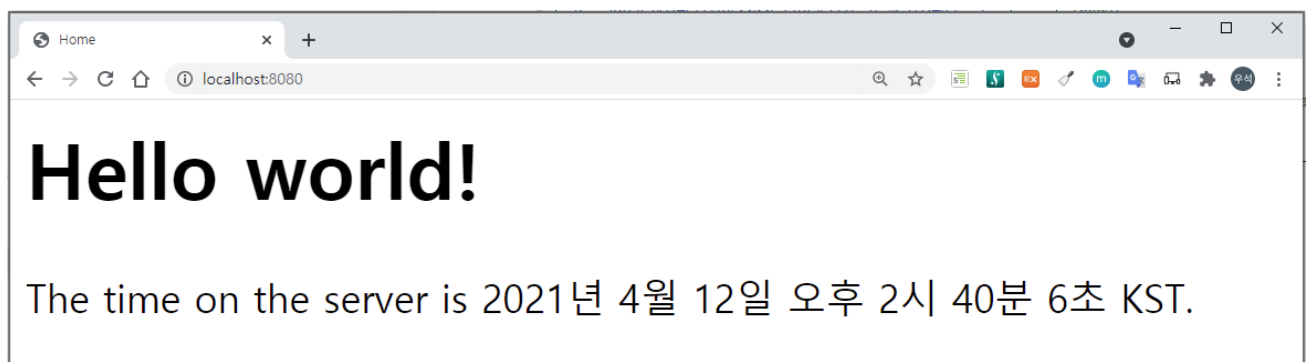
**/WEB-INF/views/home.jsp**

```

<%@ page contentType="text/html; charset=UTF-8" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ page session="false" %>
<html>
<head>
    <title>Home</title>
</head>
<body>
<h1>
    Hello world!
</h1>

<P> The time on the server is ${serverTime}. </P>
</body>
</html>

```



## 10. 로그 레벨 설정

### ◦ Log4j 구성 요소

요소	설명
Logger	출력할 메시지를 Appender에 전달한다.
Appender	전달된 로그를 어디에 출력할 지 결정한다.(콘솔 출력, 파일 기록, DB 저장 등)
Layout	로그를 어떤 형식으로 출력할 지 결정한다.

### ◦ Log4j 레벨

위의 순서대로 높은 레벨을 가지며, 출력 레벨 설정에 따라 설정 레벨 이상의 로그가 출력된다.

로그 레벨	설명
FATAL	아주 심각한 에러가 발생한 상태를 나타낸다.
ERROR	어떠한 요청을 처리하는 중 문제가 발생한 상태를 나타낸다.
WARN	프로그램의 실행에는 문제가 없지만, 향후 시스템 에러의 원인이 될 수 있는 경고성 메시지를 나타낸다.
INFO	어떠한 상태 변경과 같은 정보성 메시지를 나타낸다.
DEBUG	개발시 디버그 용도로 사용하는 메시지를 나타낸다.
TRACE	디버그 레벨이 너무 광범위한 것을 해결하기 위해서 좀 더 상세한 이벤트를 나타낸다.

### ◦ 로그 레벨 설정

log4.xml 파일이 에러 발생하면 상단 DTD 선언문을 아래 소스코드의 2 라인과 같이 수정합니다.

DTD 파일의 경로를 웹의 절대 경로로 지정한 형태입니다.

/main/resources/log4j.xml
<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;!DOCTYPE log4j:configuration SYSTEM "http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/xml/doc- files/log4j.dtd"&gt; &lt;log4j:configuration xmlns:log4j="http://jakarta.apache.org/Log4j/"      &lt;!-- Appenders --&gt;     &lt;appender name="console" class="org.apache.Log4j.ConsoleAppender"&gt;         &lt;param name="Target" value="System.out" /&gt;         &lt;layout class="org.apache.Log4j.PatternLayout"&gt;             &lt;param name="ConversionPattern" value="%-5p: %c - %m%n" /&gt;         &lt;/layout&gt;     &lt;/appender&gt;      &lt;!-- 특정 패키지 출력 설정 --&gt;     &lt;!-- Application Loggers --&gt;     &lt;logger name="com.zeus.controller"&gt; </pre>

```
        <level value="info" />
    </logger>

    <!-- 3rdparty Loggers -->
    <logger name="org.springframework.core">
        <level value="info" />
    </logger>

    <logger name="org.springframework.beans">
        <level value="info" />
    </logger>

    <logger name="org.springframework.context">
        <level value="info" />
    </logger>

    <logger name="org.springframework.web">
        <level value="info" />
    </logger>

    <!-- 기본 출력 설정 -->
    <!-- Root Logger -->
    <root>
        <priority value="warn" />
        <appender-ref ref="console" />
    </root>
</log4j:configuration>
```