Spring Boot 프로젝트에 JSP 설정 추가

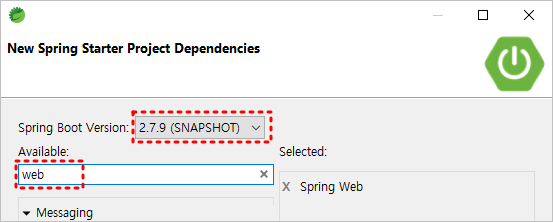
# 프로젝트명 : B00SpringBootProject

서버설정부터 할 것.

앞서 만든 B00SpringBootProject에 JSP를 위한 설정을 추가한다.

프로젝트를 새롭게 만들지 않는다.

의존설정은 Spring Web 만 되어있는 상태이다.

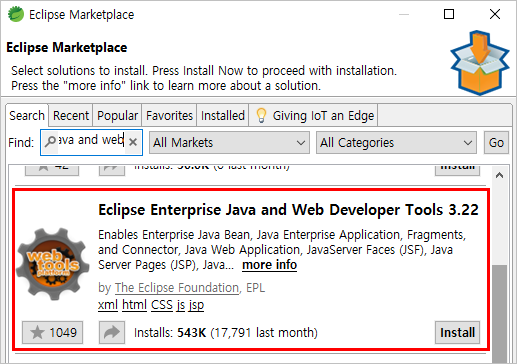




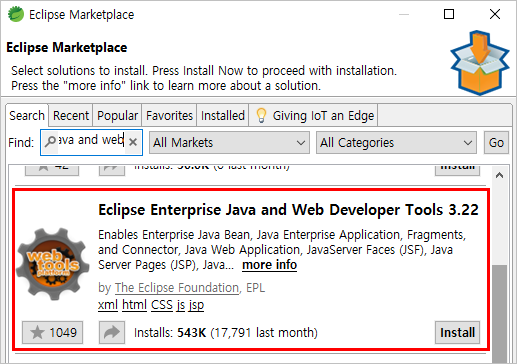
## JSP 혹은 HTML파일을 만들기 위한 설치

#STS4에서는 현시점(2024년 03월) WEB관련 HTML, CSS, JSP 파일등을 생성 하려면 Tools를 따로 설치해줘야 합니다.

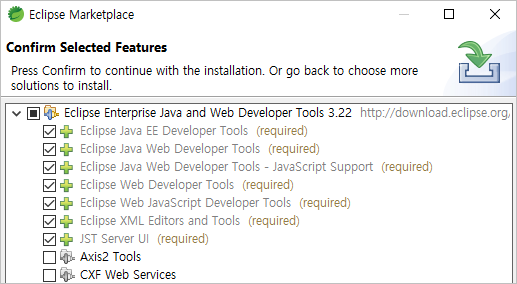
#상단 메뉴바에서 Help -> Eclipse MarketPlace... 클릭합니다.



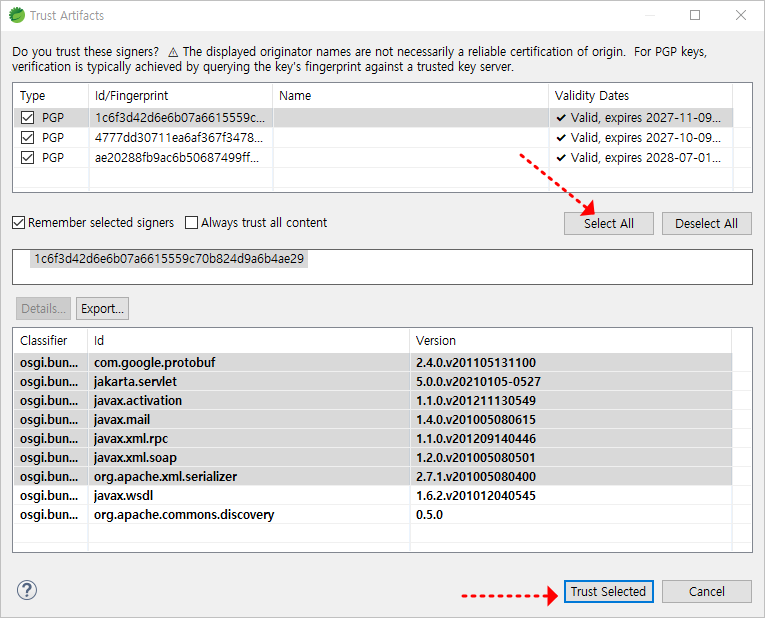
#MarketPlace에서 **Java and web** 검색 후 “Eclipse Enterprise Java and Web Developer Tools 3.xx” 항목의 **install**을 클릭해서 다운로드 받습니다.



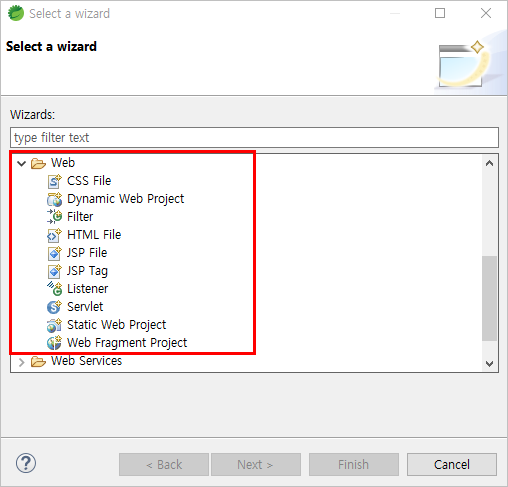
다음 창이 뜨면 **confirm**을 클릭한다.



설치 중 다음과 같은 창이 뜨면 모두 선택한 후 **Trust Selected**를 클릭한다.

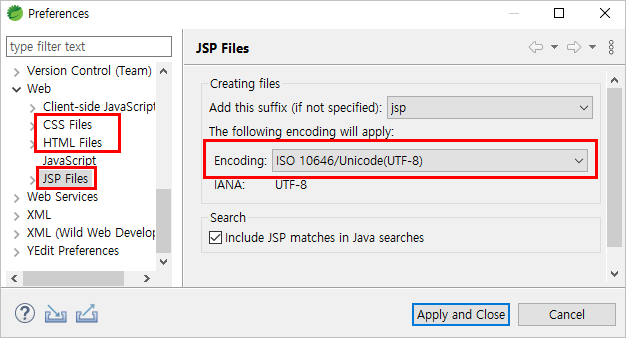


설치가 완료되고 재시작 한 후 File > New > Other > Select a wizard 에서 보면 아래와 같이 Web 항목이 추가된것을 볼 수 있다.



Web 항목이 추가 되었으므로 Encoding 설정을 해야 한다.

Window > Preferences > Web 항목에서 다음과 같이 File Encoding 부분을 UTF-8로 변경한다.



**Spring Boot가 지원하는 View(뷰)**

## ☕FreeMarker

## ☕Groovy

## 🍘Thymeleaf

* 프로젝트 생성시 의존설정을 하면 별도의 추가적인 설정없이 template 폴더 아래에 html 파일을 만들어서 사용할 수 있다.
* HTML은 프론트앤드에서 사용하는 정적(static)인 파일이므로, 프로그래밍을 위한 Java 코드를 작성할 수 없다.
* 하지만 타임리프는 이런 html 파일에 JSP와 같은 동적(dynamic)인 기능을 추가할 수 있다.

## ☕Velocity

## 🍟JSP

* Spring boot에서는 기본으로 지원되지 않는다.
* 프로젝트 생성 후 별도의 설정을 해야 사용할 수 있다.
* 우리는 JSP를 기반으로 뷰(View)를 구성하므로 주로 사용한다.

# JSP 사용을 위한 프로젝트 설정

### build.gradle

* + JSP 사용을 위한 디펜던시 추가
  + Project facet 추가 ()
  + 그레이들 업데이트

### application.properties 설정 추가

* + 포트번호 변경(default는 8080)
  + JSP 폴더 설정(View Resolver)

### 폴더 구성

* + src ⇒ main 하위에 **webapp 폴더** 추가
  + home.jsp 파일 생성

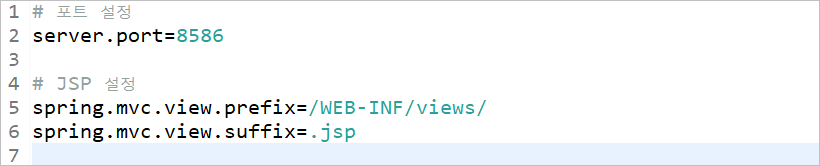
### 기본 컨트롤러 추가

* + src/main/java 하위의 기본 패키지에 MainController.java 생성
  + 기본 매핑 (@RequestMapping 어노테이션)

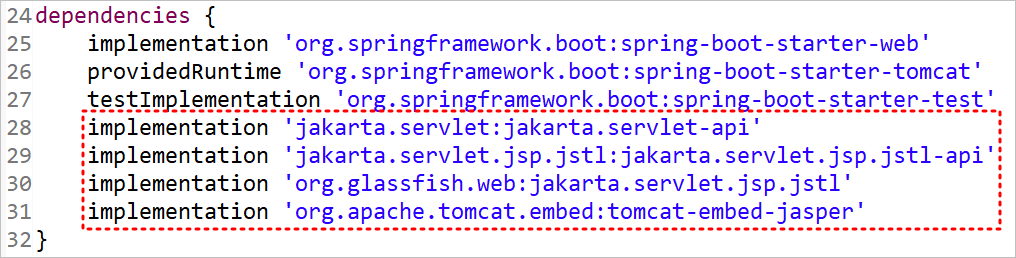
# 

# JSP 사용을 위한 설정

### 설정파일 : /B02FormSubmit/src/main/resources/application.properties

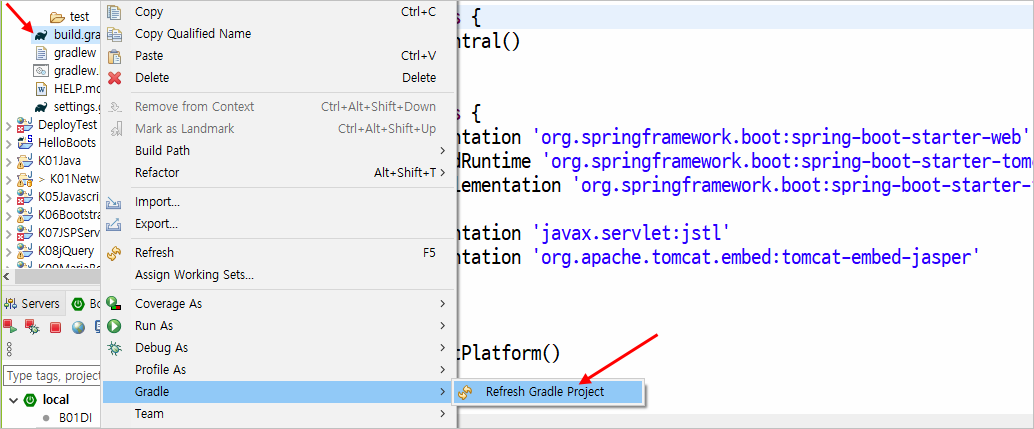


### gradle 파일 : /B02FormSubmit/build.gradle

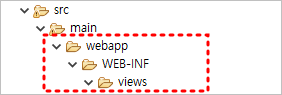


| implementation 'jakarta.servlet:jakarta.servlet-api'  implementation 'jakarta.servlet.jsp.jstl:jakarta.servlet.jsp.jstl-api'  implementation 'org.glassfish.web:jakarta.servlet.jsp.jstl'  implementation 'org.apache.tomcat.embed:tomcat-embed-jasper' |
| --- |

저장한 후 반드시 [프로젝트명 or build.gradle 선택 > 우클릭 > 팝업 메뉴 > Gradle > Refresh Gradle Project] 를 통해 적용해야 한다. 안드로이드 스튜디오에서 변경한 후 Sync now를 누르는 것과 동일하다.

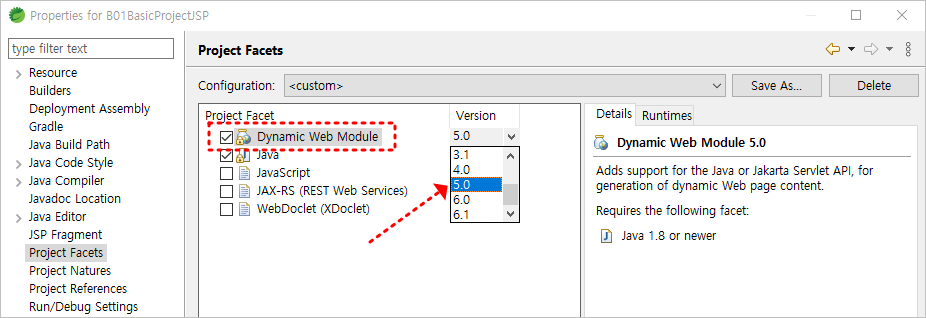


뷰경로에 맞는 폴더를 생성한다.



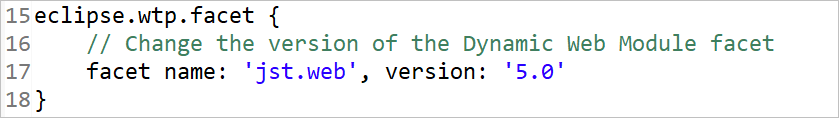
설정 후 views 폴더에 JSP를 생성하면 에러가 표시되는데, 이는 Dynamic Web Module이 맞지 않아 발생되는 부분이다.

프로젝트 ⇒ 우클릭 ⇒ Properties ⇒ Project Facets 에서 5.0을 선택한다.



이렇게 설정한 후 다시 Refresh Gradle Project를 실행하면 설정값이 기존상태로 리셋된다.

따라서 build.gradle에 다음과 같은 설정을 해야한다.



| eclipse.wtp.facet {  // Change the version of the Dynamic Web Module facet  facet name: 'jst.web', version: '5.0'  } |
| --- |

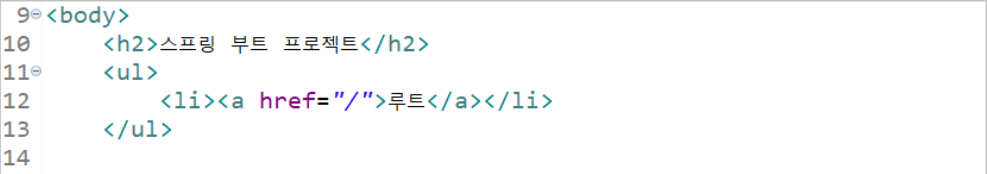
이로써 JSP를 사용하기 위한 설정이 완료되었다.

# 컨트롤러 및 뷰 생성

### 컨트롤러 : com.edu.springboot.MainController.java

### 

### home파일 : webapp/WEB-INF/views/home.jsp

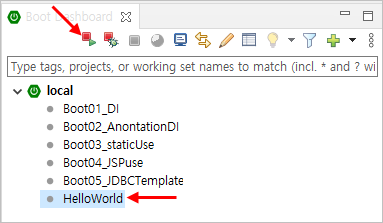


### Boot Dashboard

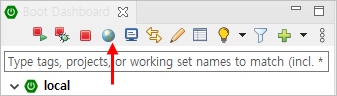
스프링 부트는 실행을 위해 Boot Dashboard 가 필요하다.



프로젝트를 선택하면 Start 버튼이 활성화된다.



프로젝트가 시작되면 아래와 같이 포트번호가 표시되면서 지구본 모양의 아이콘도 활성화된다. 아이콘을 눌러서 웹 브라우저를 실행한다.



실행화면은 다음과 같다.

