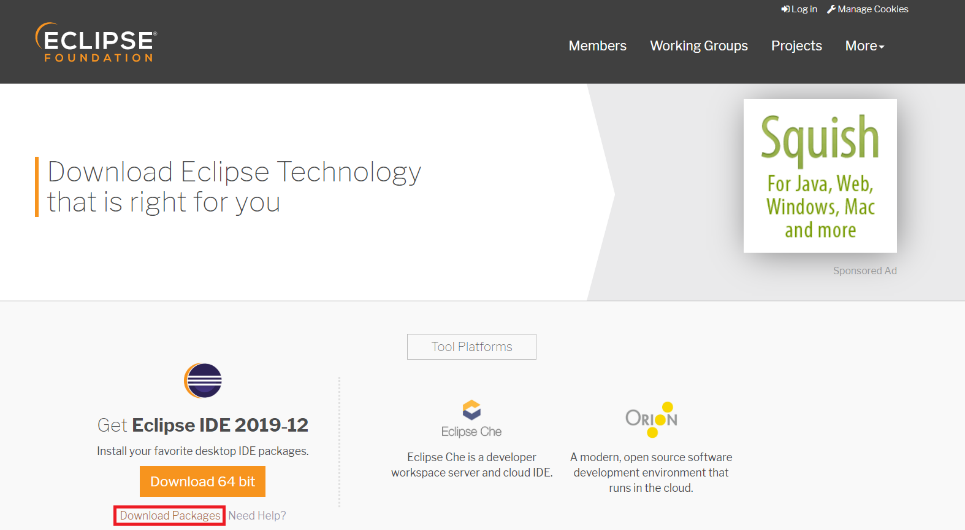
# 環境安裝

## 開發Spring MVC Project所需軟體

1. Tomcat 9.0.30
2. Eclipse IDE for Java EE Developers (2019-12 (4.14.0))
3. Spring 5.2.3 (No download required)
4. JDK 8

## 下載eclipse IDE及配置

1. **下載eclipse**
2. 至<https://www.eclipse.org/downloads/>下載eclipse開發環境，建議安裝免安裝版(下圖紅框地方)。



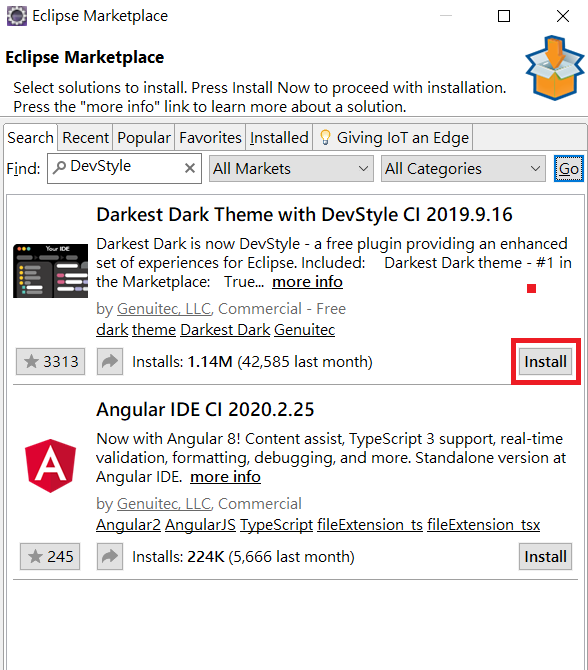
1. 從 eclipse.ini 可設定開啟Eclipse使用的參數：

|  |  |
| --- | --- |
|  | -Xms1024M  -Xmx2048M  -Xmn512M  -XX:PermSize=256M  -XX:MaxPermSize=256M |

1. **主題修改**

Ref: <https://blog.csdn.net/StillOnMyWay/article/details/79109741>

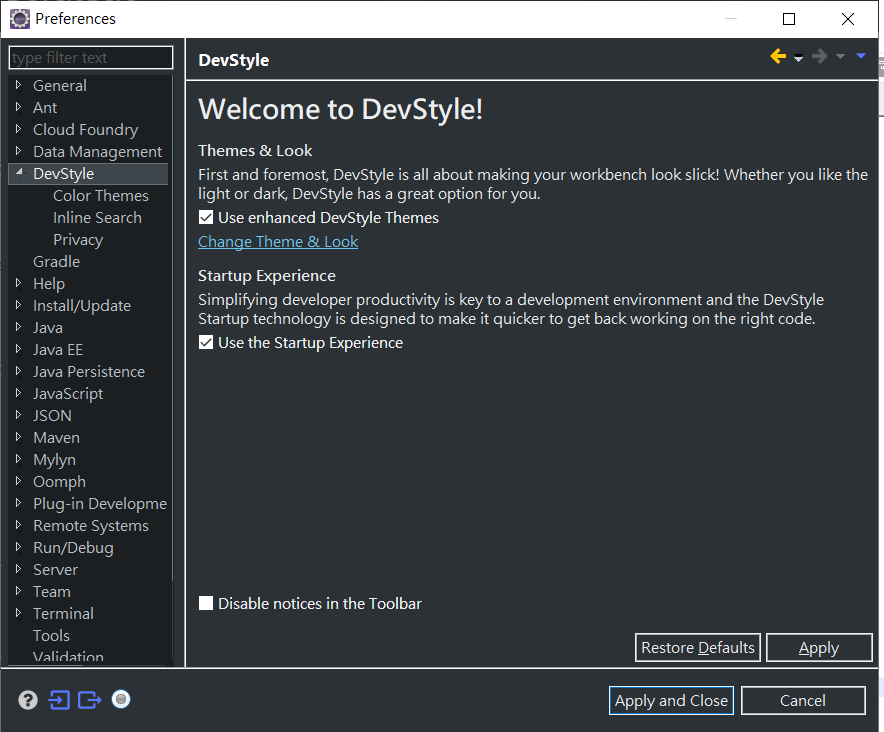
1. 打開eclipse→Help→marketplace。
2. 搜尋DevStyle點選Install。



1. 安裝後，重啟eclipse。
2. 重啟後會選擇style，挑選想要的主題，後續可以修改。
3. DevStyle設置:

Themes&Look:啟用DevStyle配置

Startup Experience:提高啟用eclipse速度



## 安裝Java

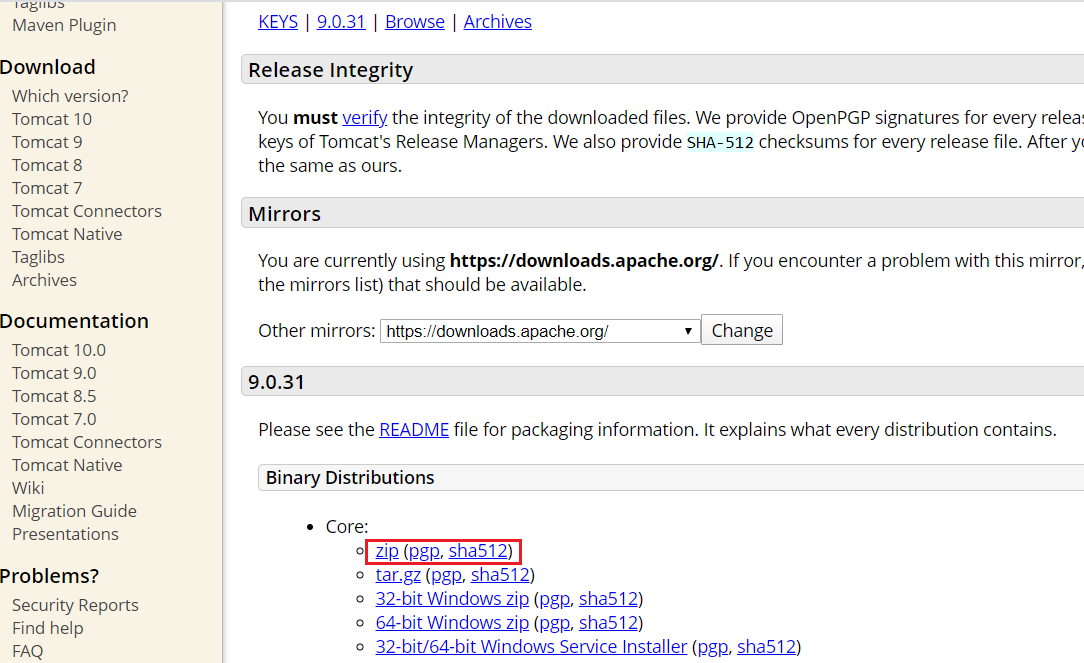
至下方下載Java並安裝。

<https://www.oracle.com/java/technologies/javase-jdk8-downloads.html>

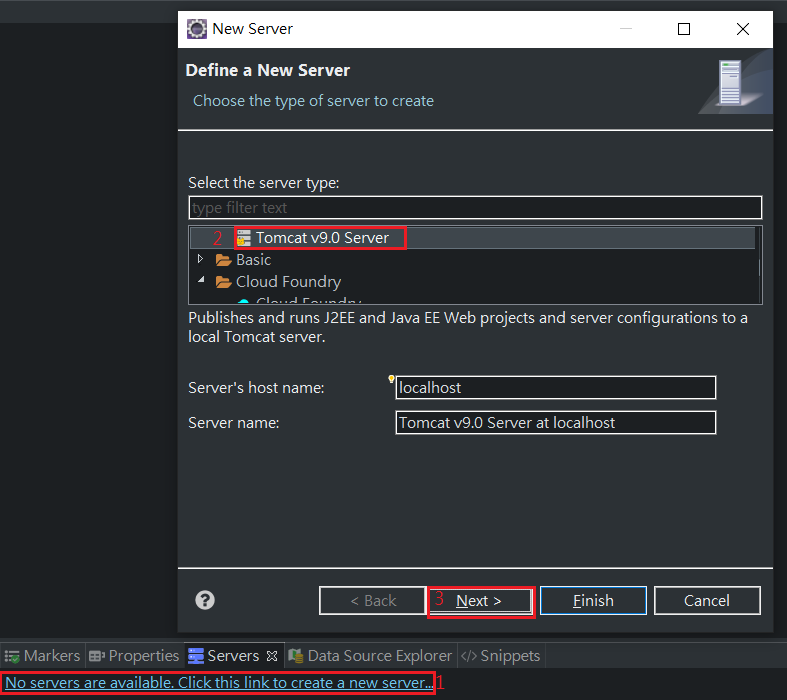
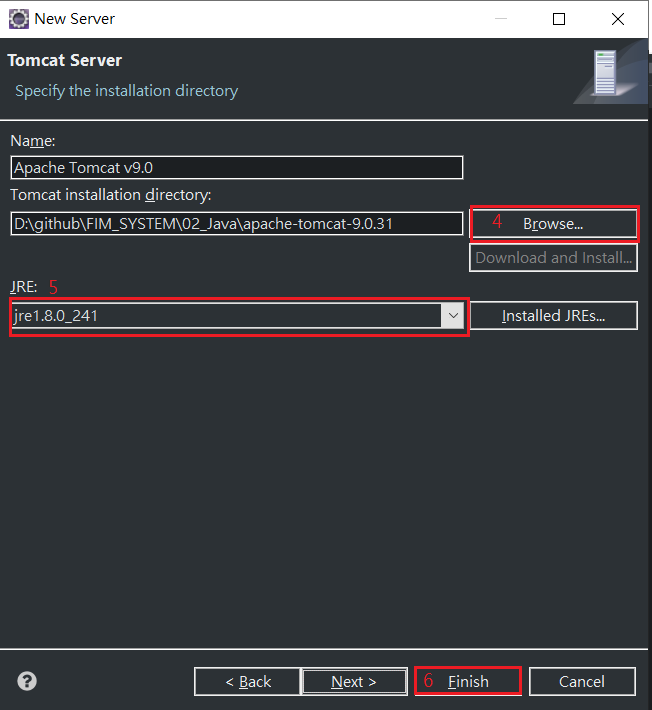
## 安裝Apache Tomcat

1. 方下載Apache Tomcat。

<https://tomcat.apache.org/download-90.cgi>



1. 解壓縮後進入eclipse進行設置。
2. 打開eclipse->打開Servers視窗

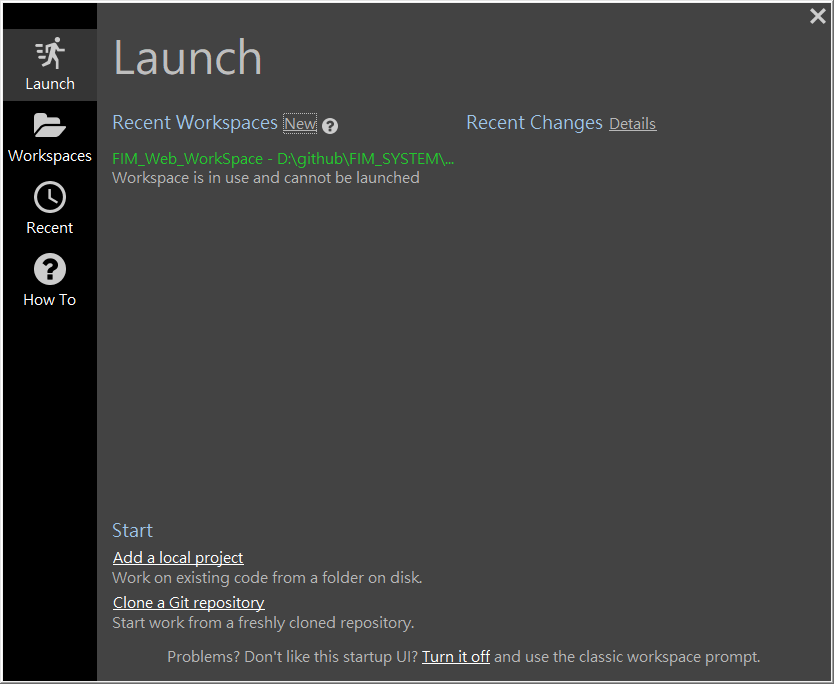
 

# 專案建置

## Create Maven Project

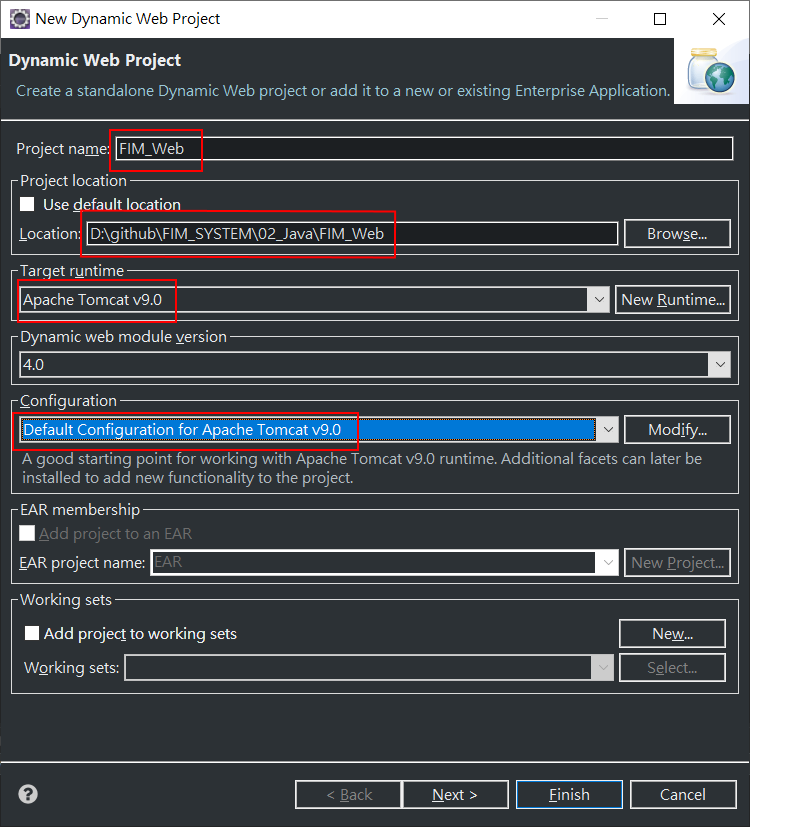
1. **Open Eclipse:**

打開eclipse後，點選適合的資料夾路徑當作Workspace。



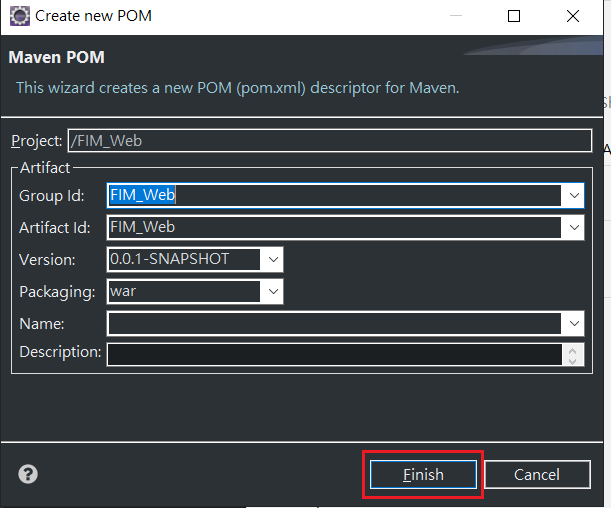
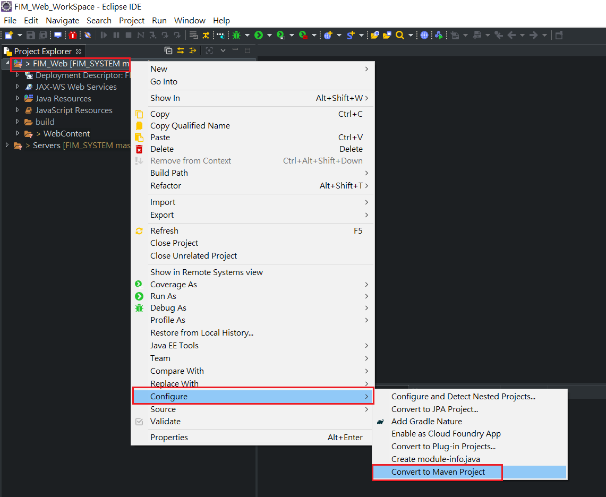
1. **Create Dynamic Web Project:**

* 點選 File
* 點選New
* 選擇 Dynamic Web Project
* Provide Project Name輸入: CrunchifySpringMVCTutorial
* 選擇程式碼路徑
* 確認 Target Runtime 為 Apache Tomcat 9.0
* 確認Configuration 為Default Configuration



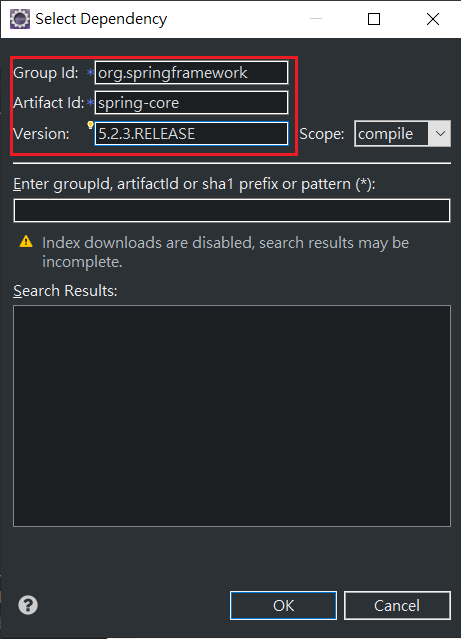
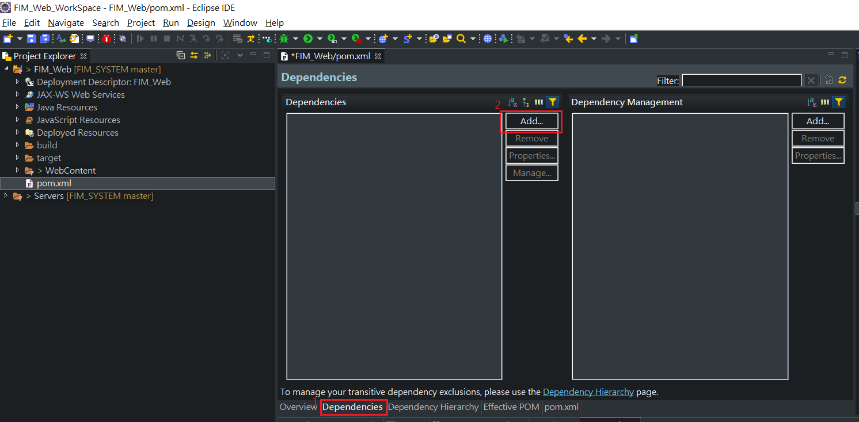
1. **Convert Project to Maven Project:**

* 專案點右鍵
* 選Configure
* 選擇Convert to Maven project
* Finish
* 左邊專案會出現pom.xml的檔案



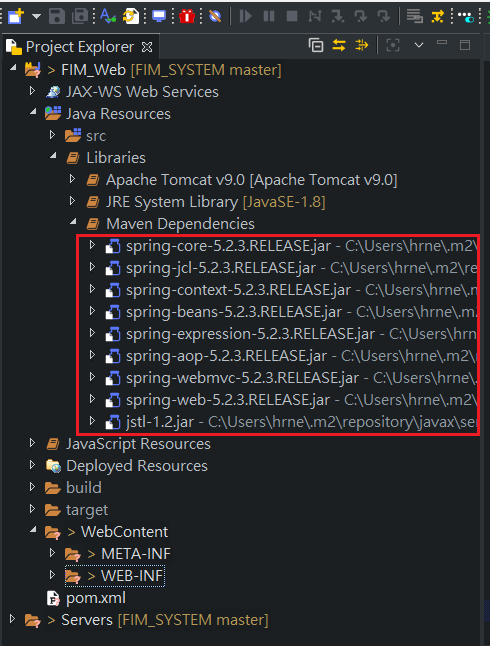
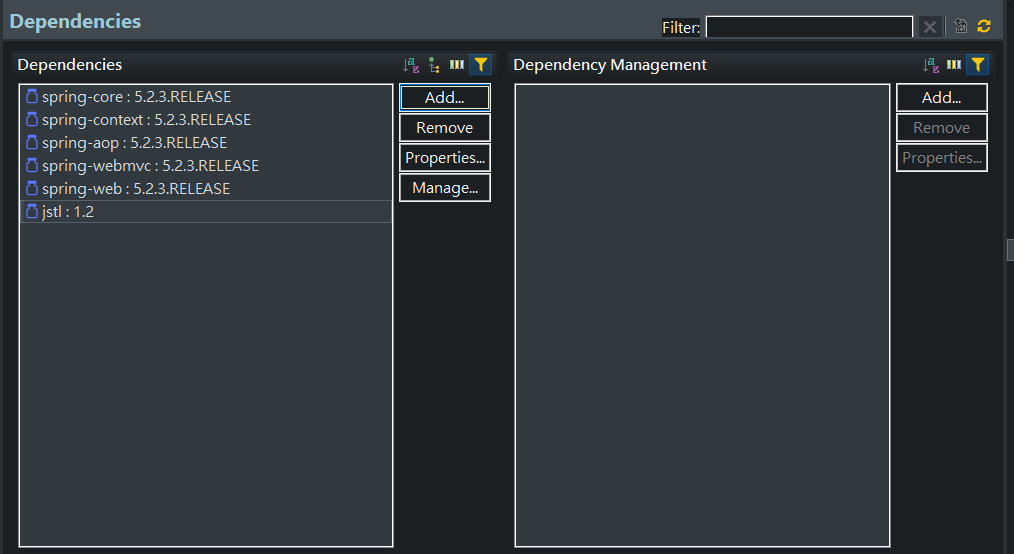
## 導入Spring Jar

1. 打開pom.xml
2. 下方選擇Dependencies
3. 按下Add，加入選用的jar



|  |  |  |  |
| --- | --- | --- | --- |
|  | groupId | artifactId | version |
|  | org.springframework | spring-core | 5.2.3.RELEASE |
|  | org.springframework | spring-context | 5.2.3.RELEASE |
|  | org.springframework | spring-aop | 5.2.3.RELEASE |
|  | org.springframework | spring-webmvc | 5.2.3.RELEASE |
|  | org.springframework | spring-web | 5.2.3.RELEASE |
|  | javax.servlet | jstl | 1.2 |

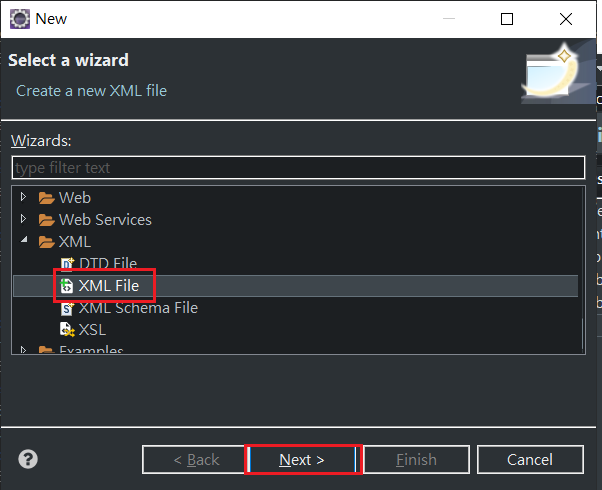
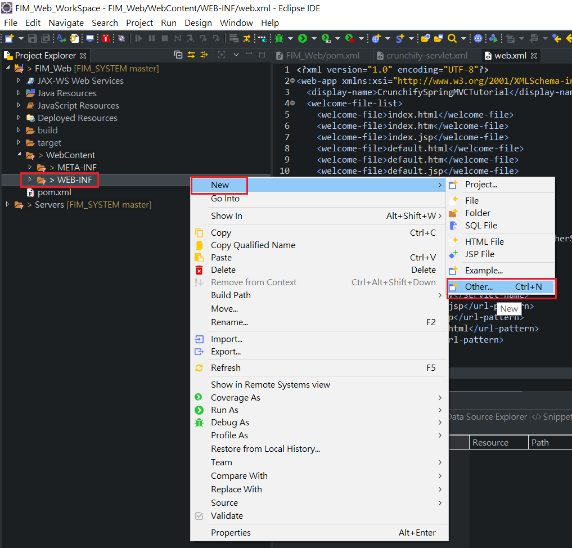
1. 產生結果如下，可以在左邊專案資料夾中看到jar檔已匯入

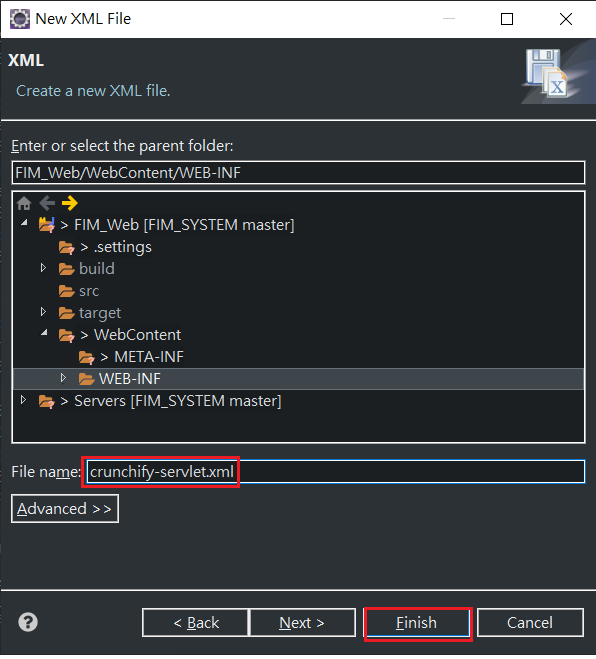


## 測試專案運行

1. **Create new Spring Configuration**

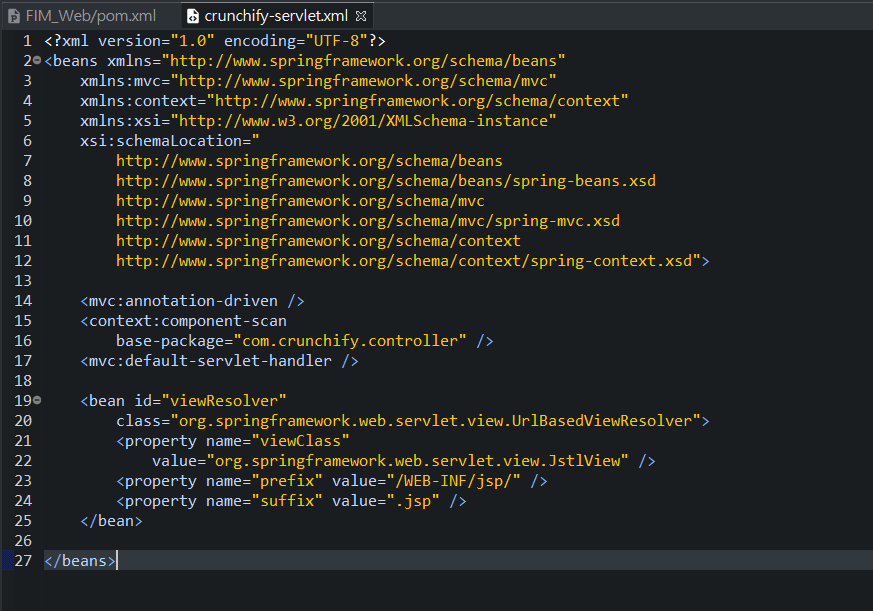
* WEB-INF右鍵
* 點New
* 點Other
* XML→XML File→Next
* File name: crunchify-servlet.xml





* crunchify-servlet.xml程式碼:

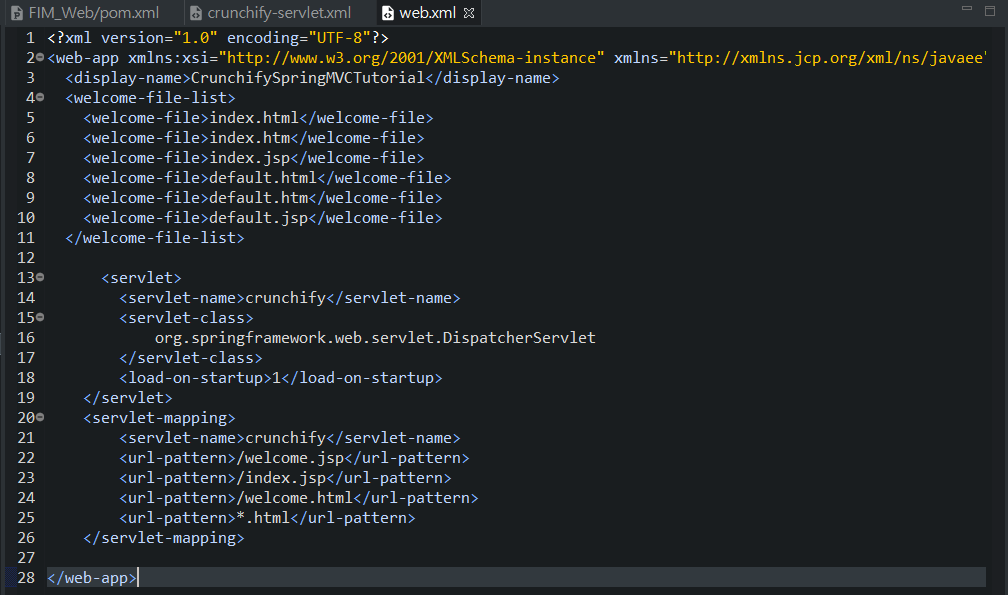
|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans"  xmlns:mvc="http://www.springframework.org/schema/mvc"  xmlns:context="http://www.springframework.org/schema/context"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="  http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://www.springframework.org/schema/mvc  http://www.springframework.org/schema/mvc/spring-mvc.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context.xsd">    <mvc:annotation-driven />  <context:component-scan  base-package="com.crunchify.controller" />  <mvc:default-servlet-handler />    <bean id="viewResolver"  class="org.springframework.web.servlet.view.UrlBasedViewResolver">  <property name="viewClass"  value="org.springframework.web.servlet.view.JstlView" />  <property name="prefix" value="/WEB-INF/jsp/" />  <property name="suffix" value=".jsp" />  </bean>    </beans> |



1. **Create web.xml**

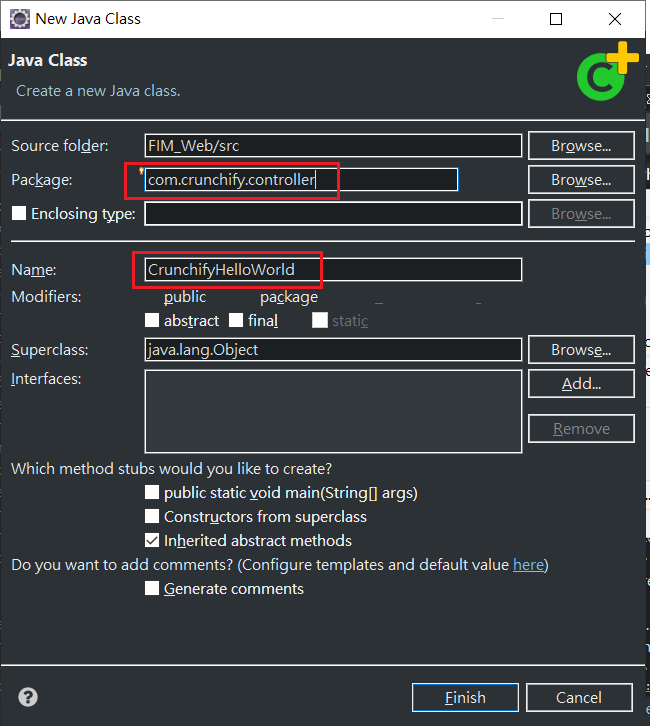
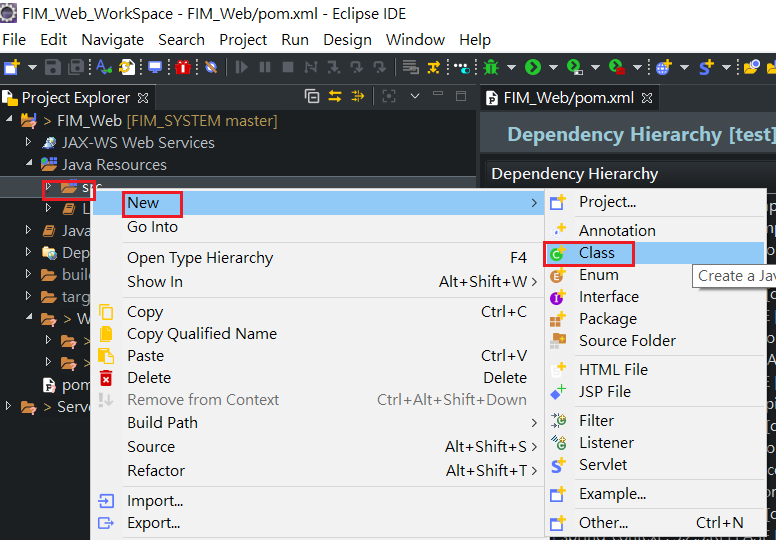
* Map Spring MVC in /WebContent/WEB-INF/web.xml file.
* web.xml程式碼:

|  |
| --- |
| <?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\_4\_0.xsd" id="WebApp\_ID" version="4.0">  <display-name>CrunchifySpringMVCTutorial</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>    <servlet>  <servlet-name>crunchify</servlet-name>  <servlet-class>  org.springframework.web.servlet.DispatcherServlet  </servlet-class>  <load-on-startup>1</load-on-startup>  </servlet>  <servlet-mapping>  <servlet-name>crunchify</servlet-name>  <url-pattern>/welcome.jsp</url-pattern>  <url-pattern>/index.jsp</url-pattern>  <url-pattern>/welcome.html</url-pattern>  <url-pattern>\*.html</url-pattern>  </servlet-mapping>    </web-app> |



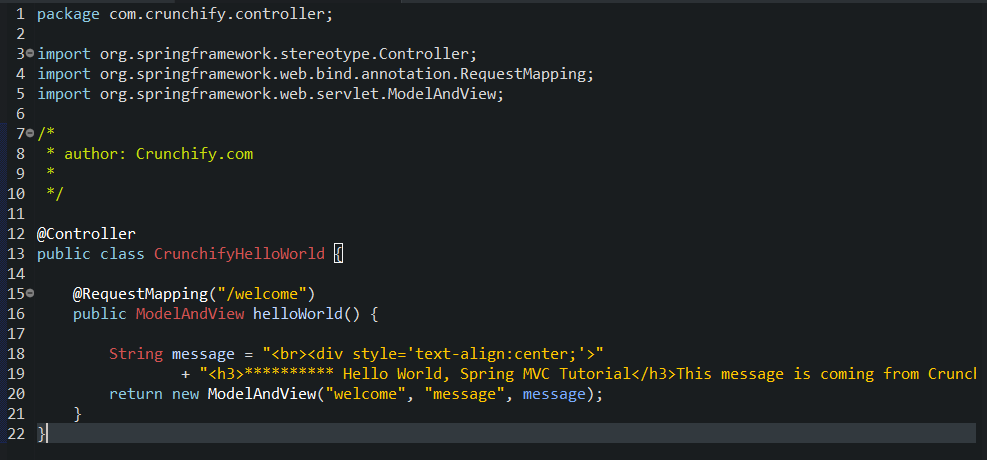
1. **Create Controller Class**

* Right click on Java Resources -> src
* Click New -> Class
* Package: com.crunchify.controller
* Filename: CrunchifyHelloWorld.java



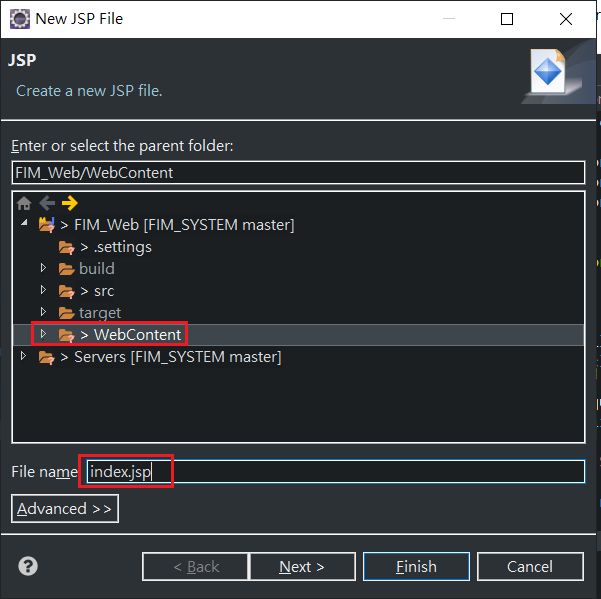
* CrunchifyHelloWorld.java程式碼:

|  |
| --- |
| **package** com.crunchify.controller;    **import** **org.springframework.stereotype.Controller**;  **import** **org.springframework.web.bind.annotation.RequestMapping**;  **import** **org.springframework.web.servlet.ModelAndView**;    /\*  \* author: Crunchify.com  \*  \*/    **@Controller**  **public** **class** **CrunchifyHelloWorld** {    **@RequestMapping**("/welcome")  **public** ModelAndView **helloWorld**() {    String message = "<br><div style='text-align:center;'>"  + "<h3>\*\*\*\*\*\*\*\*\*\* Hello World, Spring MVC Tutorial</h3>This message is coming from CrunchifyHelloWorld.java \*\*\*\*\*\*\*\*\*\*</div><br><br>";  **return** **new** **ModelAndView**("welcome", "message", message);  }  } |



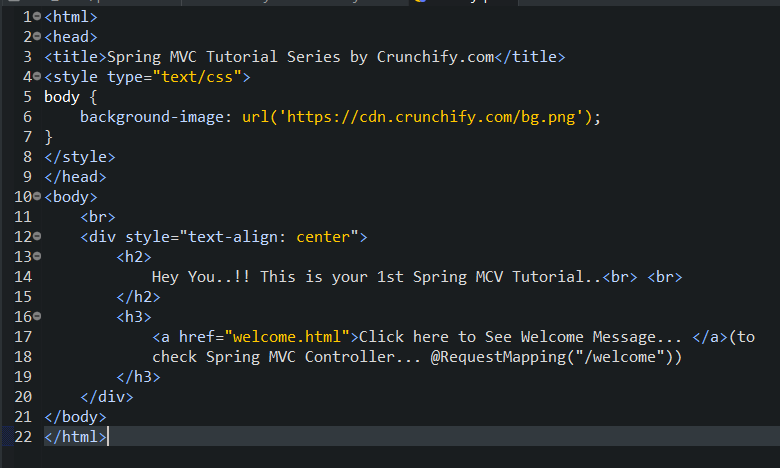
1. **Create index.jsp**

* Create new file /WebContent/index.jsp



* index.jsp程式碼:

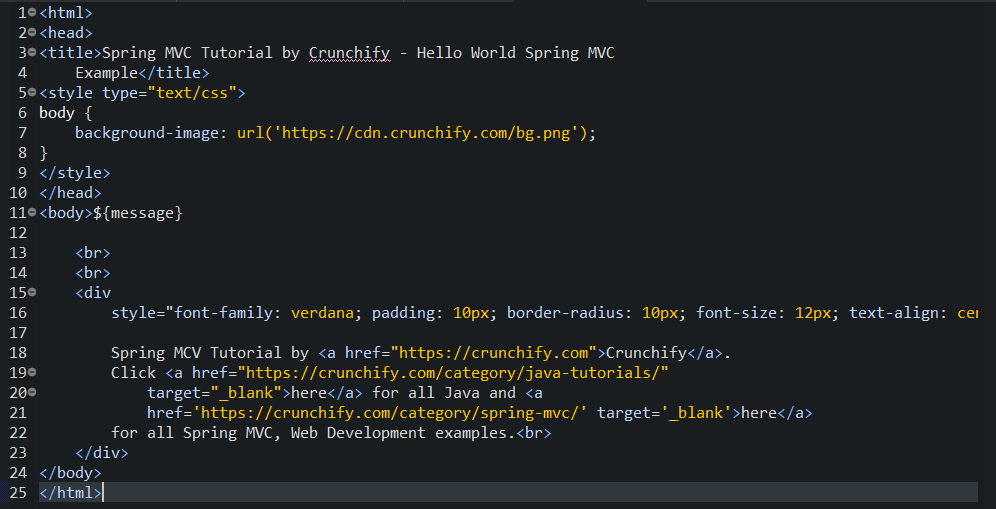
|  |
| --- |
| <html>  <head>  <title>Spring MVC Tutorial Series by Crunchify.com</title>  <style type="text/css">  body {  background-image: url('https://cdn.crunchify.com/bg.png');  }  </style>  </head>  <body>  <br>  <div style="text-align: center">  <h2>  Hey You..!! This is your 1st Spring MCV Tutorial..<br> <br>  </h2>  <h3>  <a href="welcome.html">Click here to See Welcome Message... </a>(to  check Spring MVC Controller... @RequestMapping("/welcome"))  </h3>  </div>  </body>  </html> |



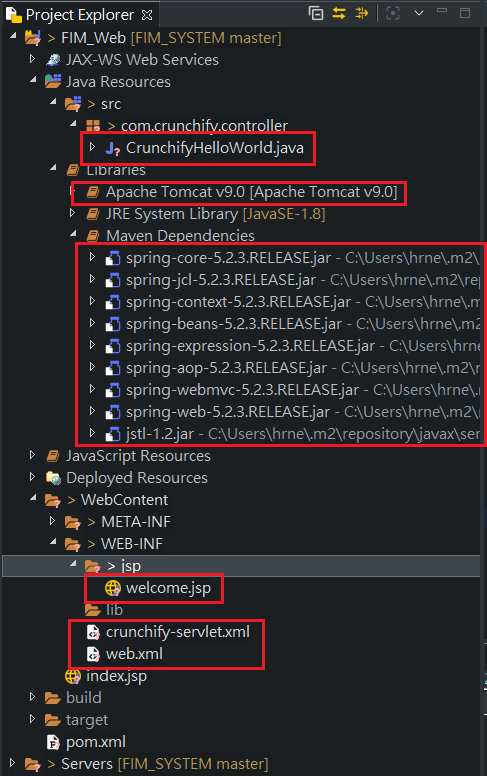
1. **Create welcom.jsp**

* Create jsp folder: /WebContent/WEB-INF/jsp
* Create another file /WebContent/WEB-INF/jsp/welcome.jsp
* welcome.jsp程式碼:

|  |
| --- |
| <html>  <head>  <title>Spring MVC Tutorial by Crunchify - Hello World Spring MVC  Example</title>  <style type="text/css">  body {  background-image: url('https://cdn.crunchify.com/bg.png');  }  </style>  </head>  <body>${message}    <br>  <br>  <div  style="font-family: verdana; padding: 10px; border-radius: 10px; font-size: 12px; text-align: center;">    Spring MCV Tutorial by <a href="https://crunchify.com">Crunchify</a>.  Click <a href="https://crunchify.com/category/java-tutorials/"  target="\_blank">here</a> for all Java and <a  href='https://crunchify.com/category/spring-mvc/' target='\_blank'>here</a>  for all Spring MVC, Web Development examples.<br>  </div>  </body>  </html> |

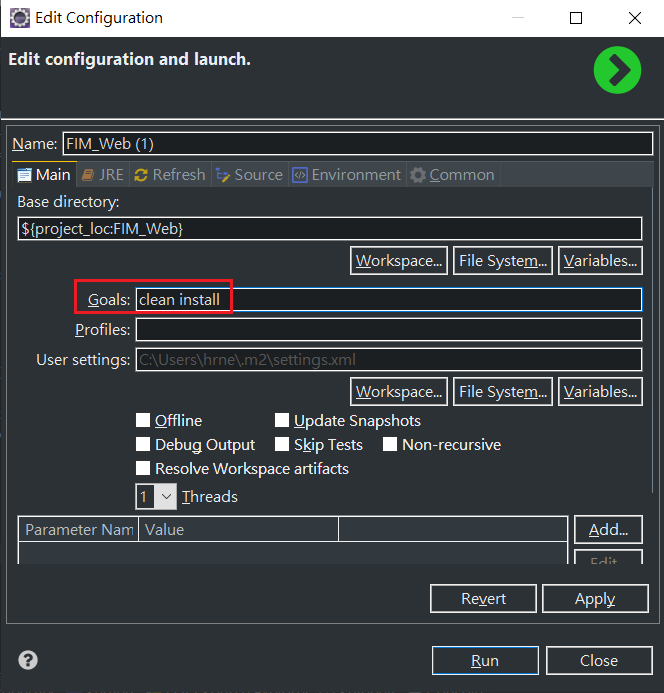
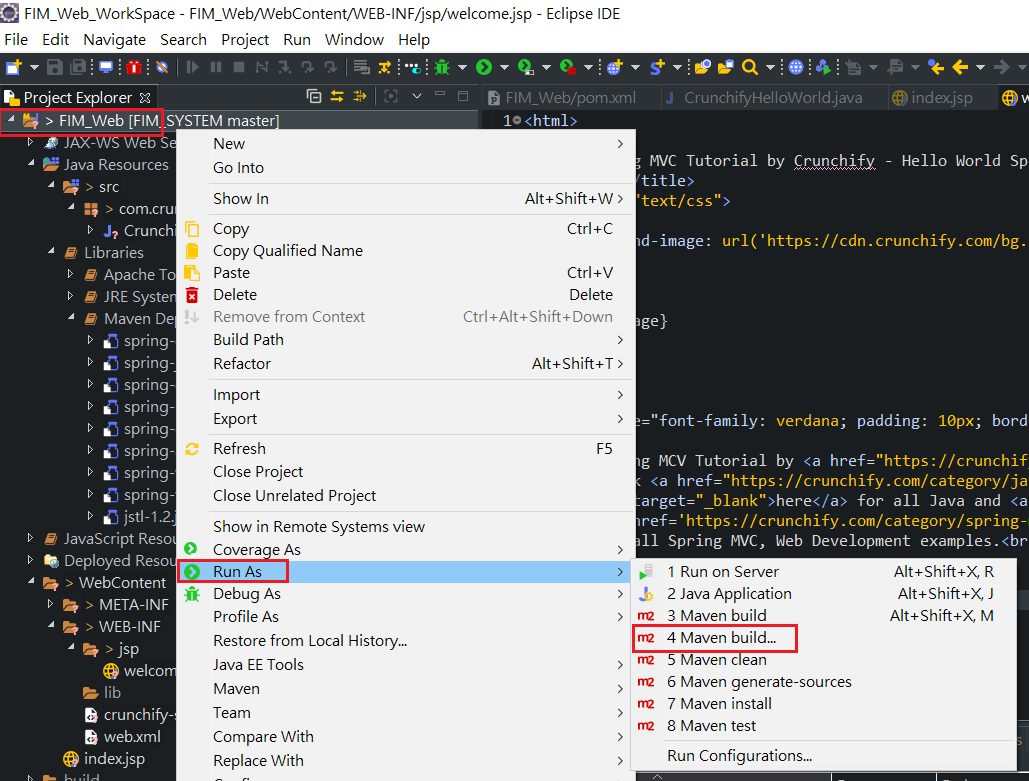


1. **確認目錄架構**

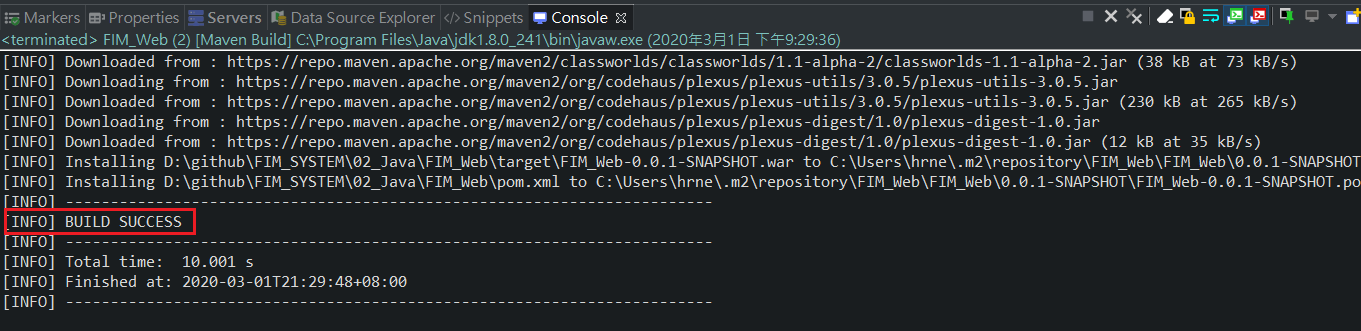


1. **Project build**

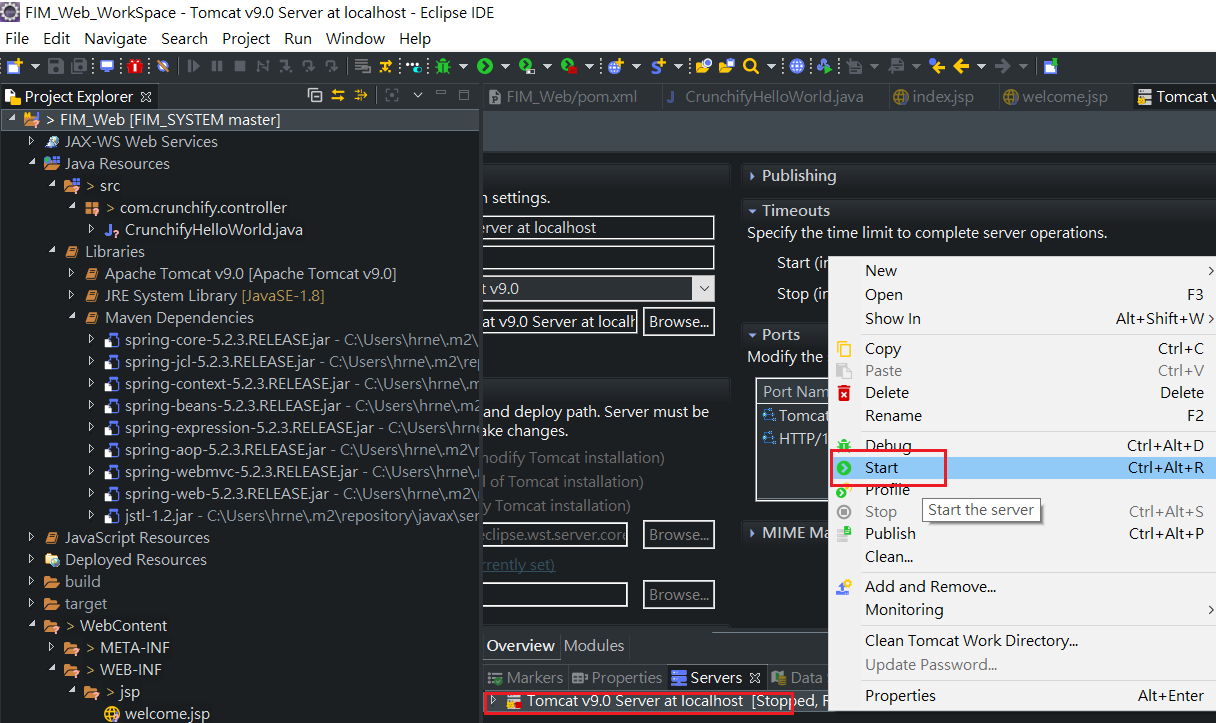
* Right Click on Project -> Run As -> Maven Build...
* Add Goals: clean install. Click Apply and Run

****

* Console出現: success message



* Tomcat v9.0右鍵
* Start運行

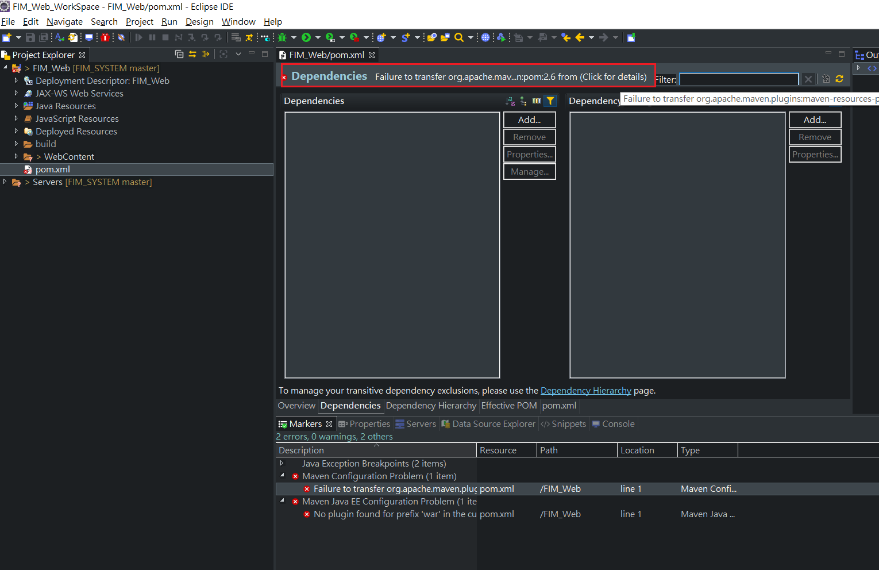
****

****

# 專案問題處理

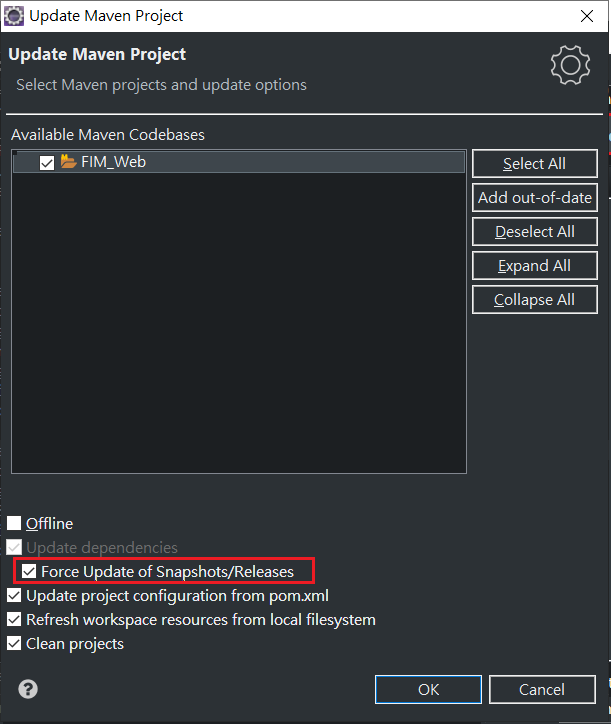
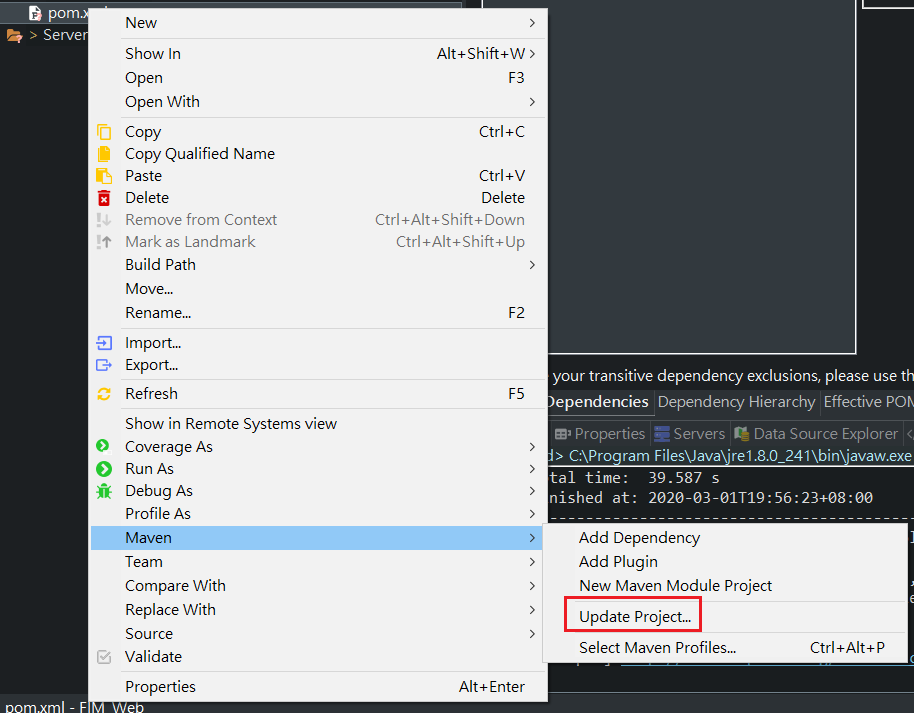
## 導入maven，pom.xml報錯

狀況:pom.xml一直出現問題。



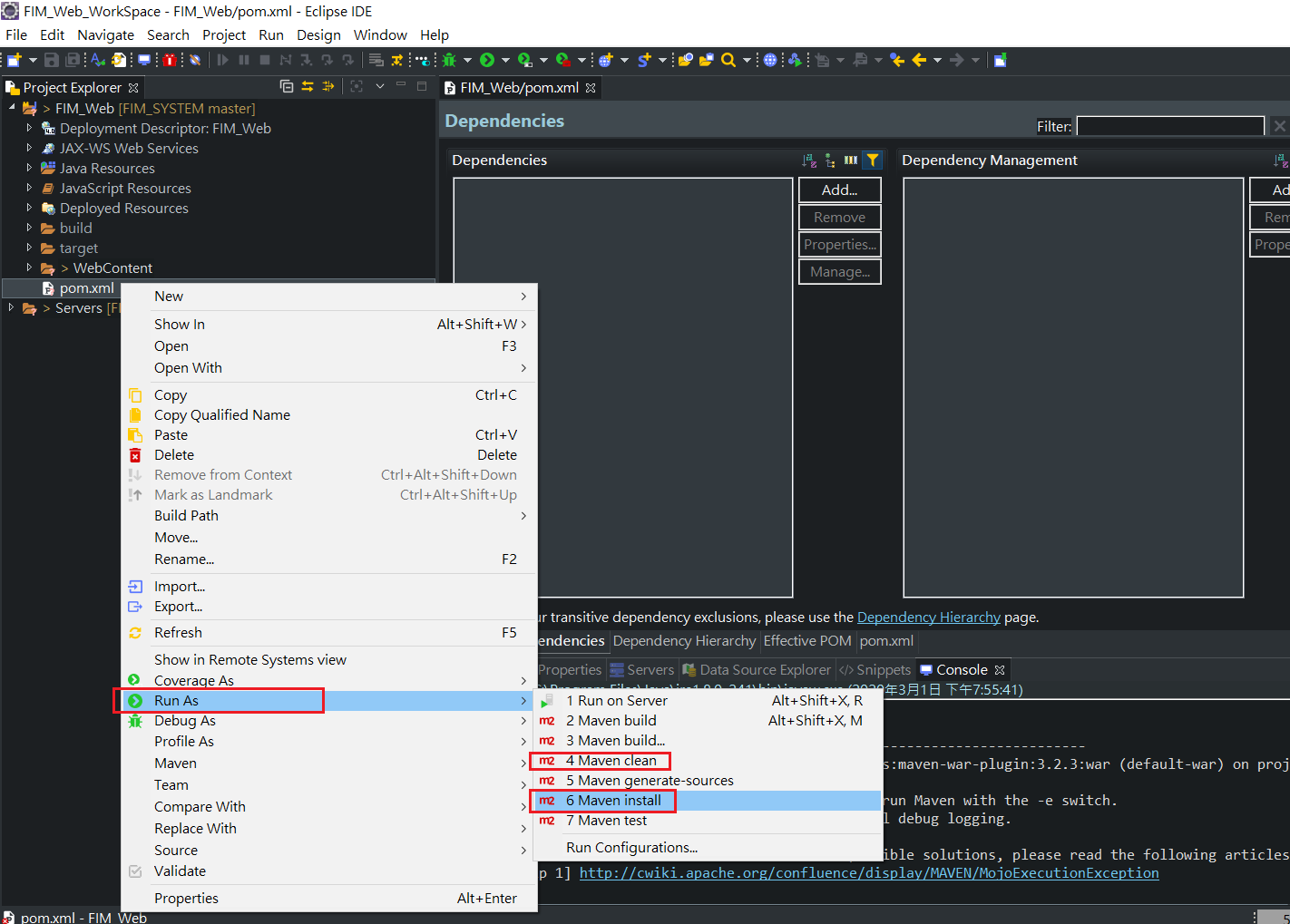
解決方式一:

* pom.xml右鍵
* 選Maven
* 選Update Project
* 下方Force Update of Snapshots/Release
* 按OK



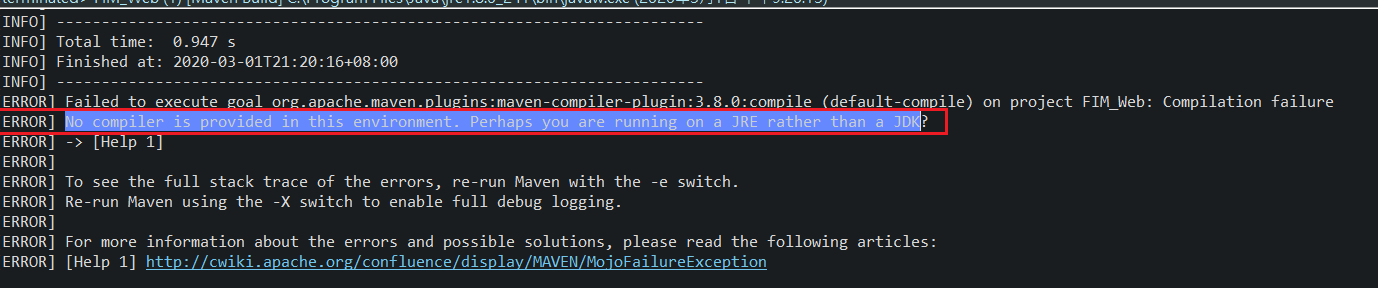
解決方式二:

* pom.xml右鍵
* 選Maven clean
* 完成後再選Maven install



## No compiler is provided in this environment. Perhaps you are running on a JRE rather than a JDK

1. 狀況: maven run的時候出現錯誤



1. 解決: eclipse預設是运行在jre上的，
2. 但是maven需要使用jdk，因此需要在eclipse修改Installed JRES

