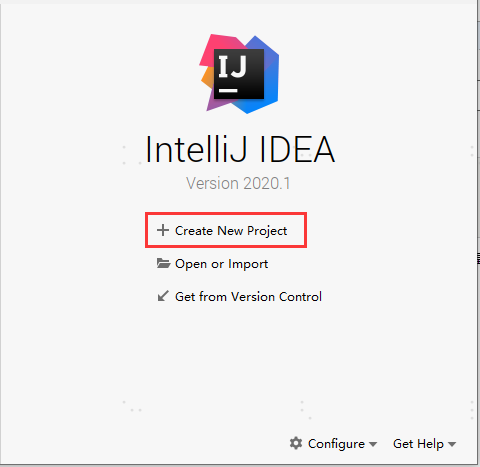
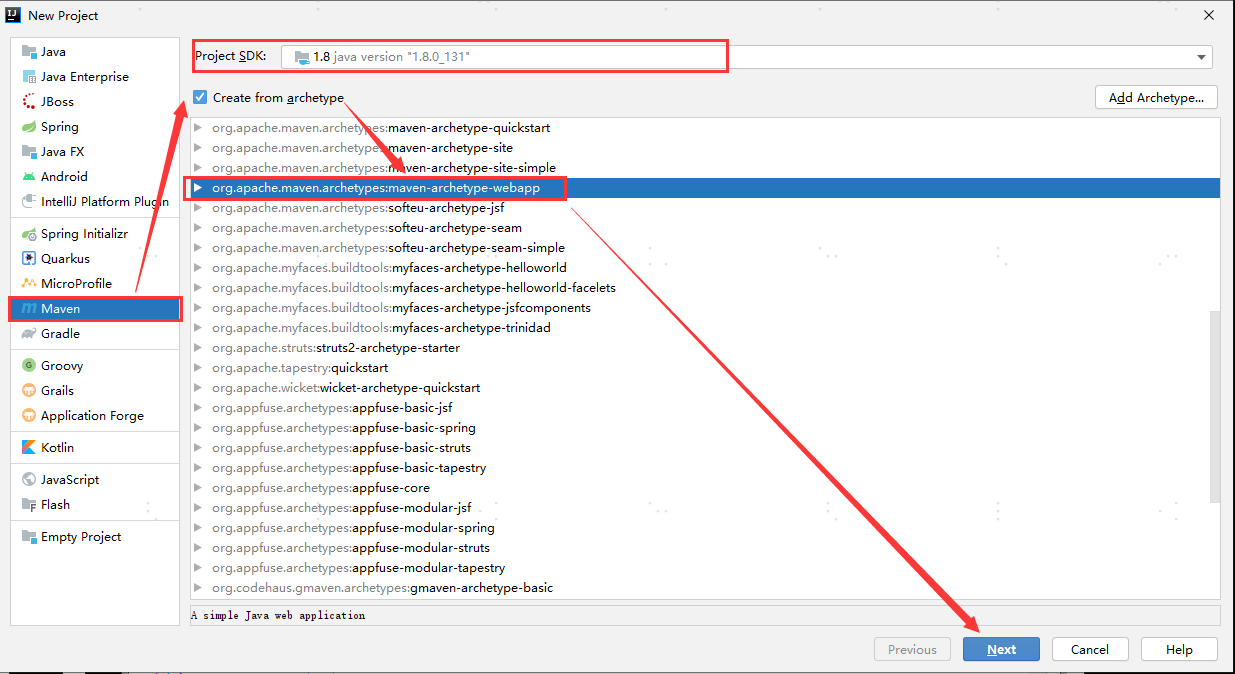
**由于工作原因，很久没有进行编码，对基础的springmvc工程已忘记如何搭建，因此花费点时间从新回忆下，顺便熟悉下IDEA的基础功能（随时保持学习啊！！！）**

## 1.创建工程

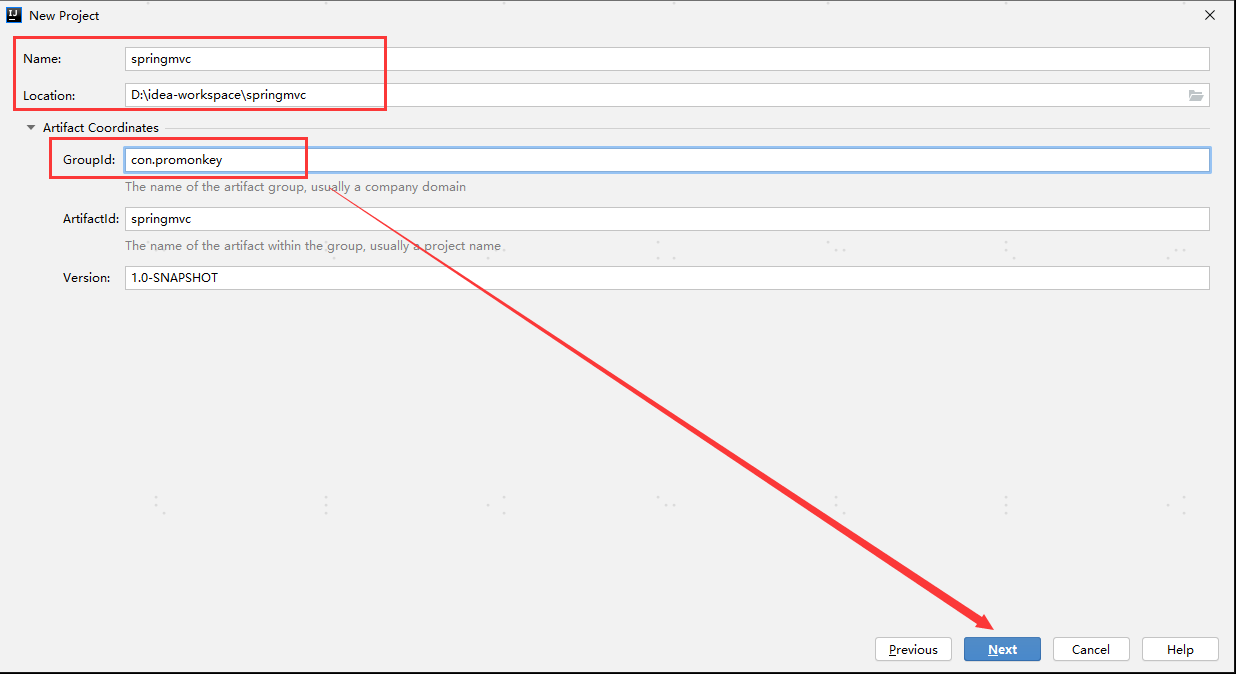
* 启动IDEA，选择“Create New Project”



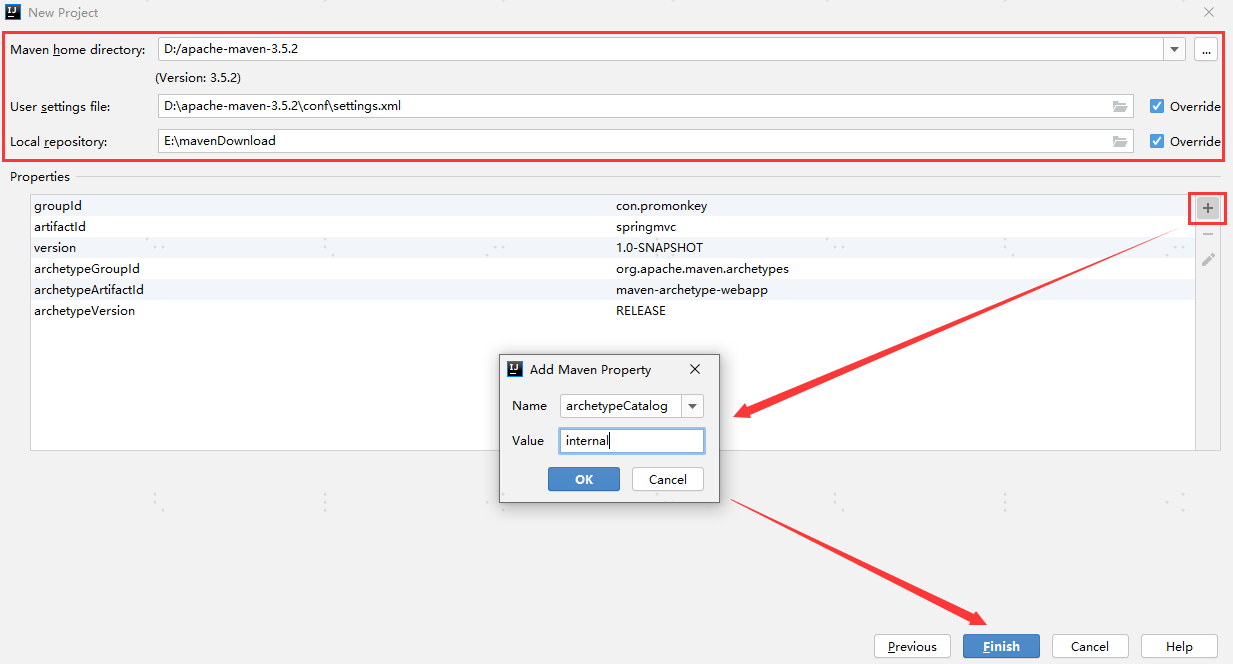
* 项目基于JDK1.8搭建
* 左侧选择“Maven”
* 右侧选择“org.apache.maven.archetypes:maven-archetype-webapp”



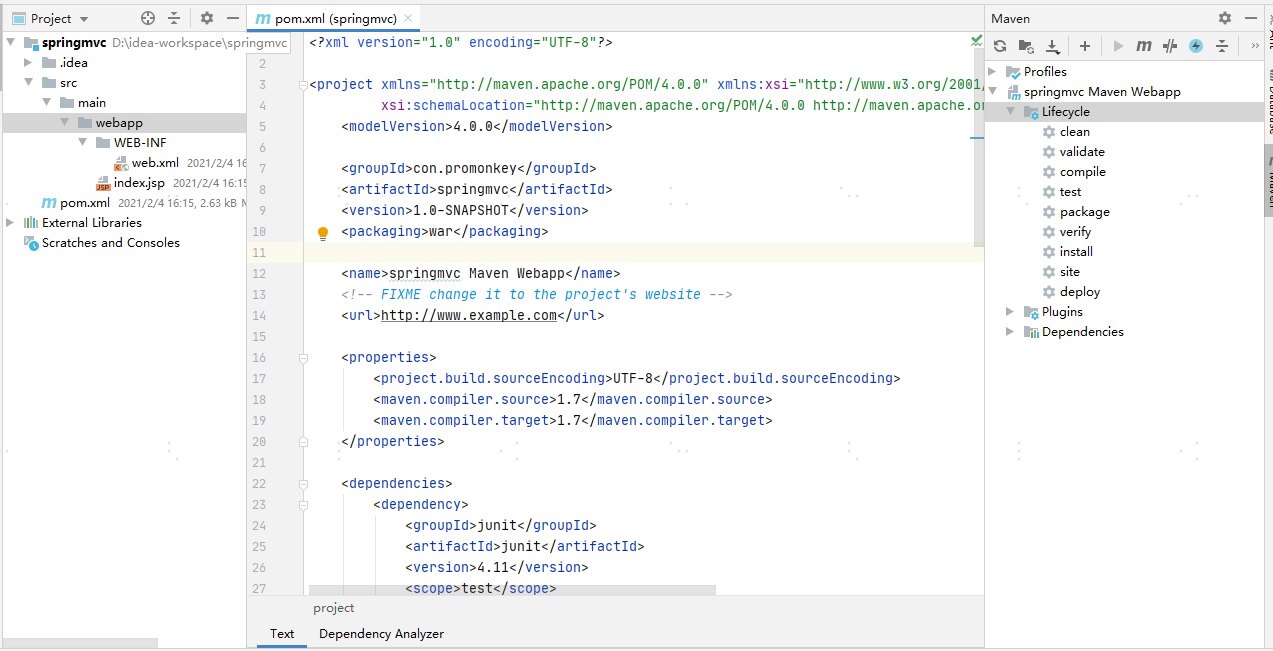
* 填写工程名称（将来war包的名称）
* 选择工程路径



* 指定Maven信息
* 添加属性name：archetypeCatalog  value：internal，使用模板目录

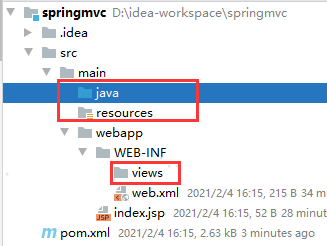


* 工程创建完之后如下图所示



## 2.配置工程

* 完善工程目录，创建java、resource目录
* 在WEB-INF下创建views目录，用于存放页面文件



* 添加maven依赖

[IMG_262](https://www.cnblogs.com/chenty/p/javascript:void(0);)

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

<spring.version>4.2.6.RELEASE</spring.version>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.11</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

<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>2.2</version>

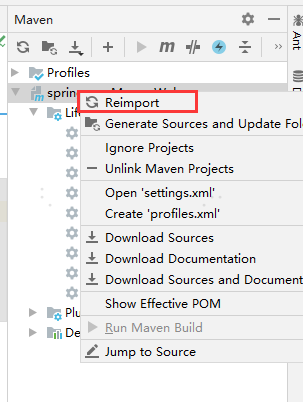
<scope>provided</scope>

</dependency>

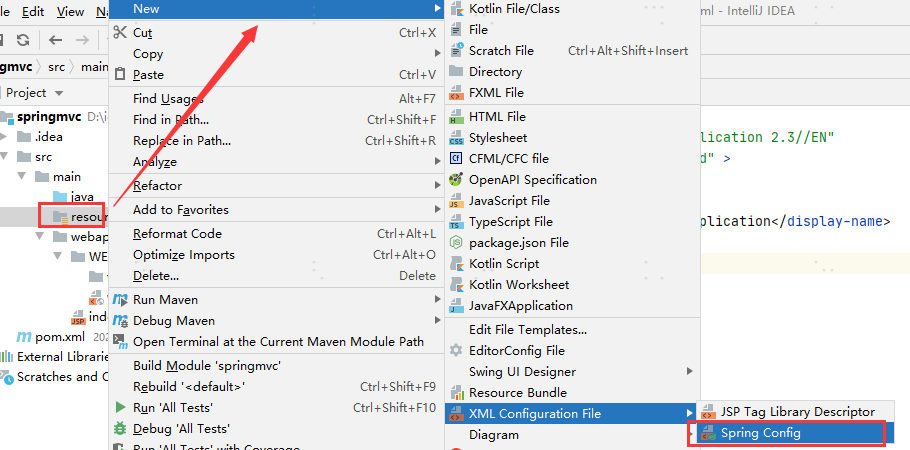
</dependencies>

[IMG_263](https://www.cnblogs.com/chenty/p/javascript:void(0);)

* 如果添加完之后有依赖没有正常进入，可以点击“Reimport”进行从新导入



* 配置sprinv-mvc.xml文件





[IMG_267](https://www.cnblogs.com/chenty/p/javascript:void(0);)

<!--开启MVC框架注解-->

<mvc:annotation-driven/>

<!--开启注解扫描-->

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

<!-- 添加视图解析 -->

<bean id="internalResourceViewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<!-- 前缀（目录） -->

<property name="prefix" value="/WEB-INF/views/"/>

<!-- 后缀 -->

<property name="suffix" value=".jsp"/>

</bean>

[IMG_268](https://www.cnblogs.com/chenty/p/javascript:void(0);)

* 配置web.xml,支持springmvc框架

[IMG_269](https://www.cnblogs.com/chenty/p/javascript:void(0);)

<!--配置DispatcherServlet-->

<servlet>

<servlet-name>springMVC</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<!--配置dispatcher.xml作为mvc的配置文件-->

<param-name>contextConfigLocation</param-name>

<param-value>classpath:spring-mvc.xml</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

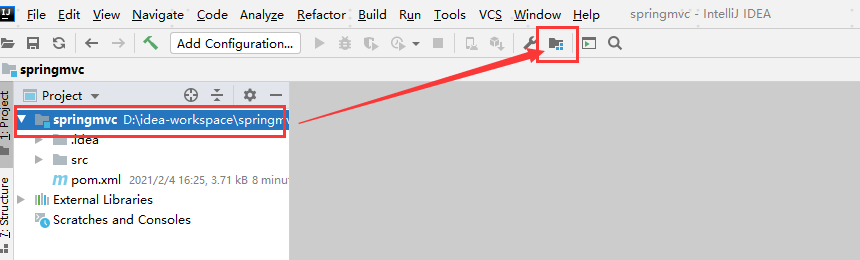
<servlet-name>springMVC</servlet-name>

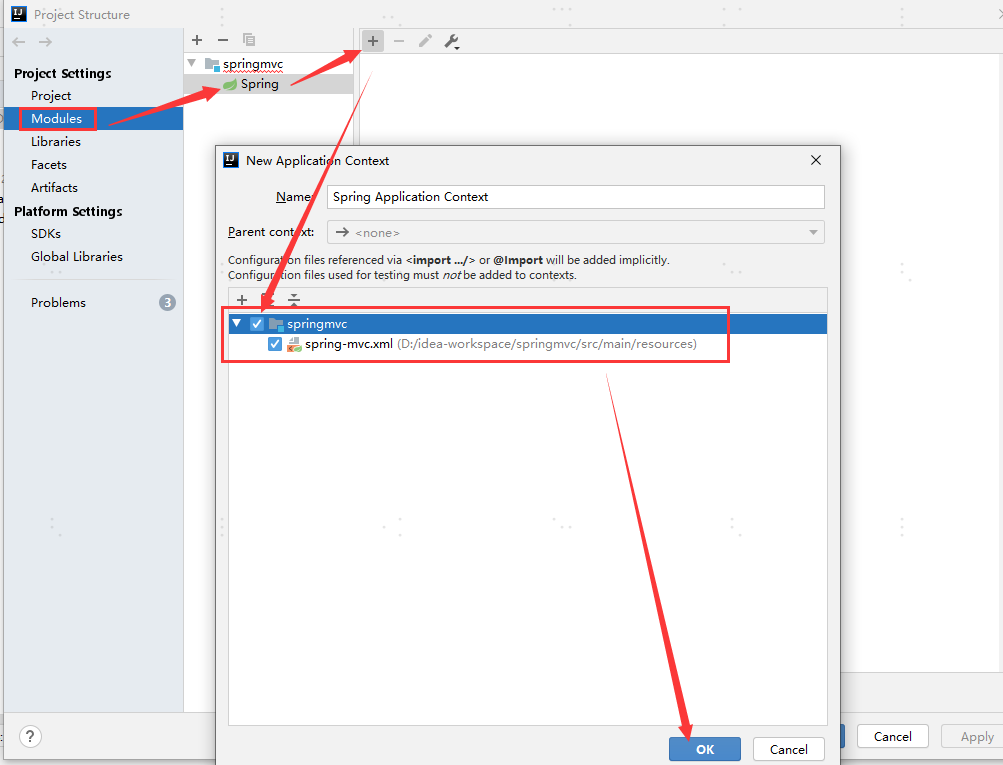
<url-pattern>/</url-pattern>

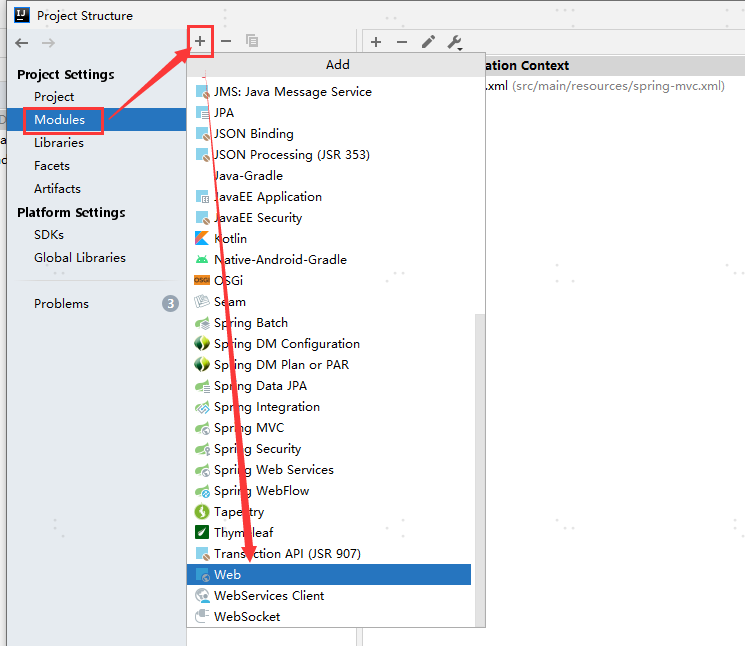
</servlet-mapping>

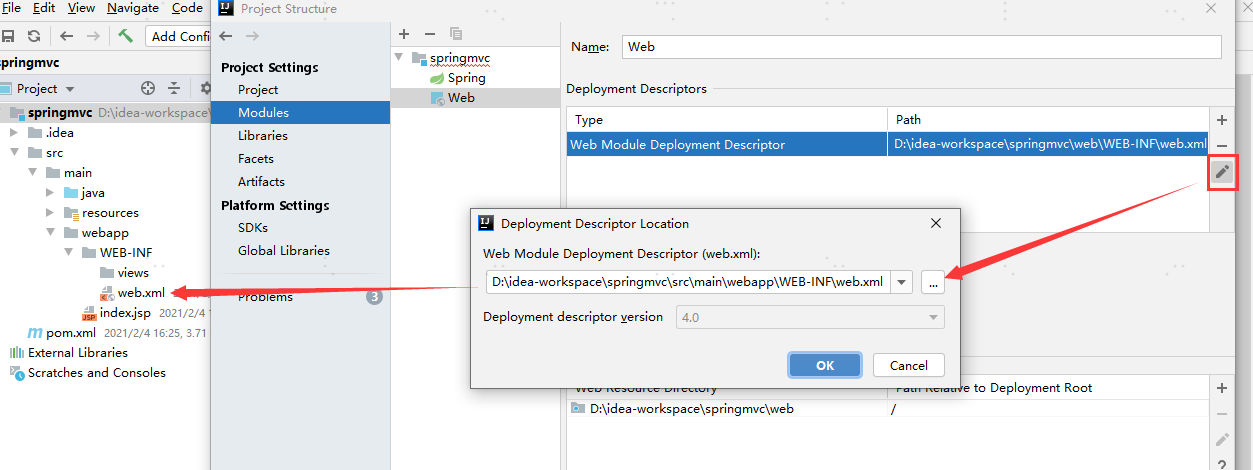
[IMG_270](https://www.cnblogs.com/chenty/p/javascript:void(0);)

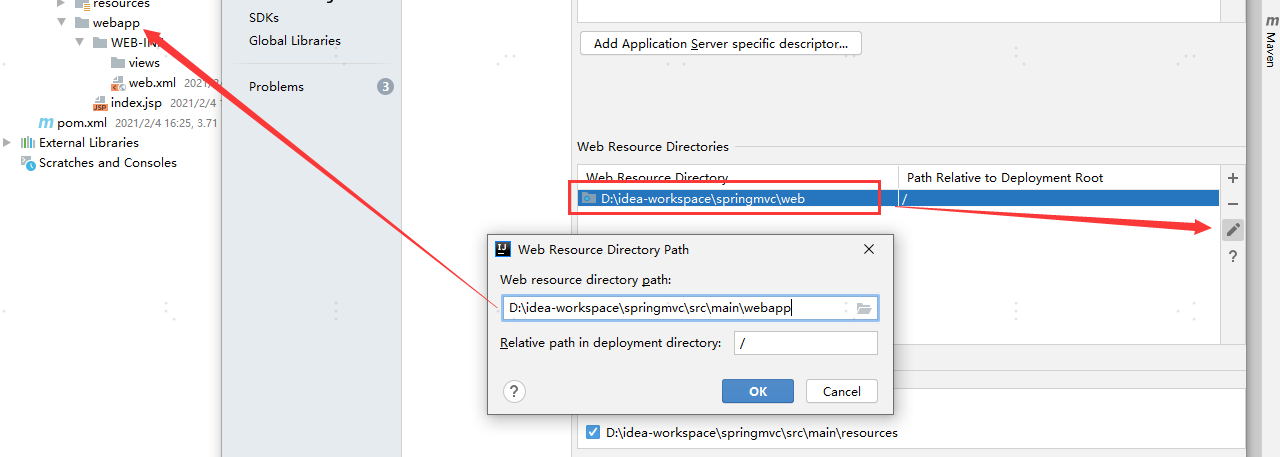
* 配置项目结构





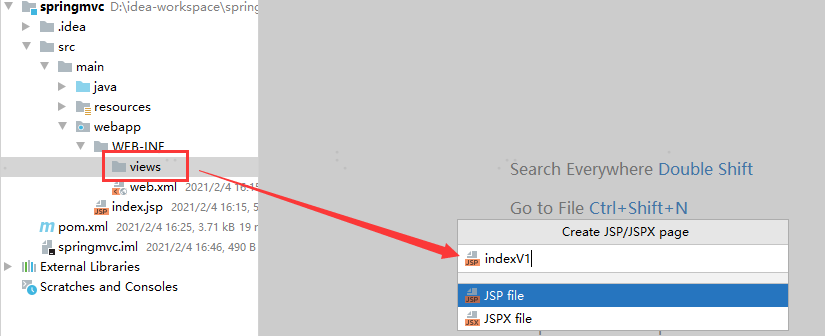






* 配置完成后，“webapp”目录的图标将从IMG_276变成IMG_277

在views目录下创建indexV1.jsp



[IMG_279](https://www.cnblogs.com/chenty/p/javascript:void(0);)

<%@ page import="java.util.Date" %><%@ page contentType="text/html;charset=UTF-8" language="java" %><html><head>

<title>Title</title></head><body>

当前时间：<%=new Date()%></body></html>

[IMG_280](https://www.cnblogs.com/chenty/p/javascript:void(0);)

* 创建IndexController

[IMG_281](https://www.cnblogs.com/chenty/p/javascript:void(0);)

package com.promonkey.controller;

import org.springframework.stereotype.Controller;import org.springframework.web.bind.annotation.RequestMapping;

@Controller

@RequestMappingpublic class IndexController {

@RequestMapping("/index")

public String index(){

return "indexV1";

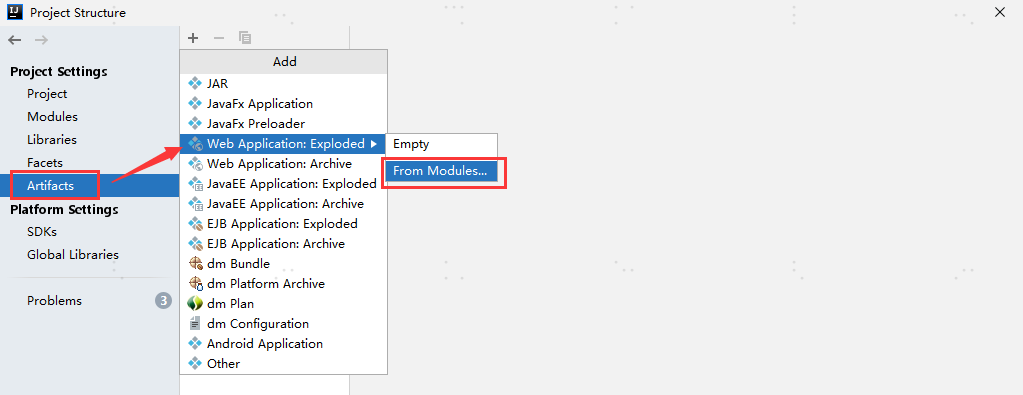
}

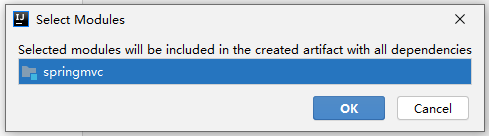
}

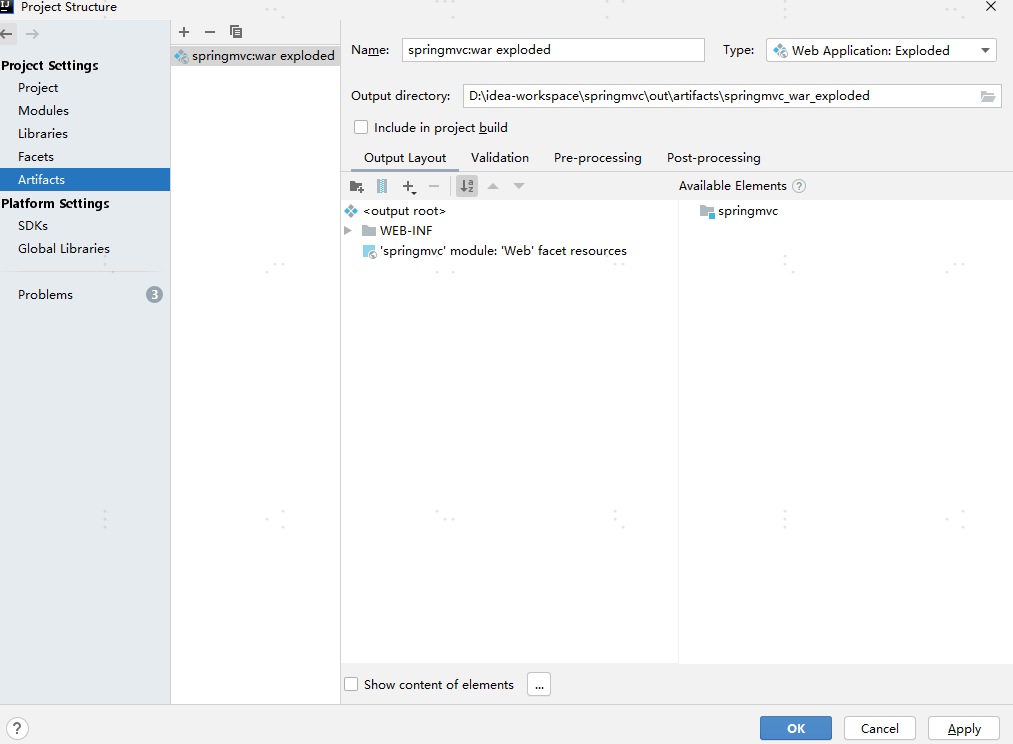
[IMG_282](https://www.cnblogs.com/chenty/p/javascript:void(0);)

## 3.配置发布包

* 创建Artifacts，选择Project Structure，选择“Artifacts”

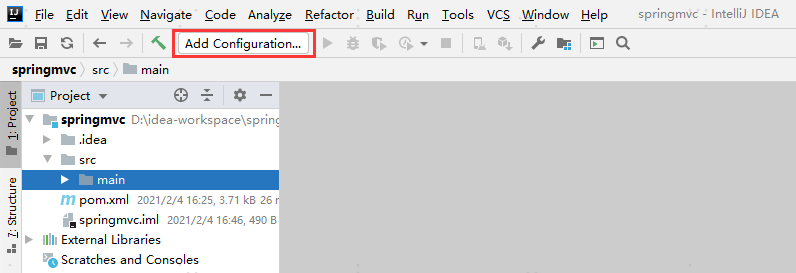


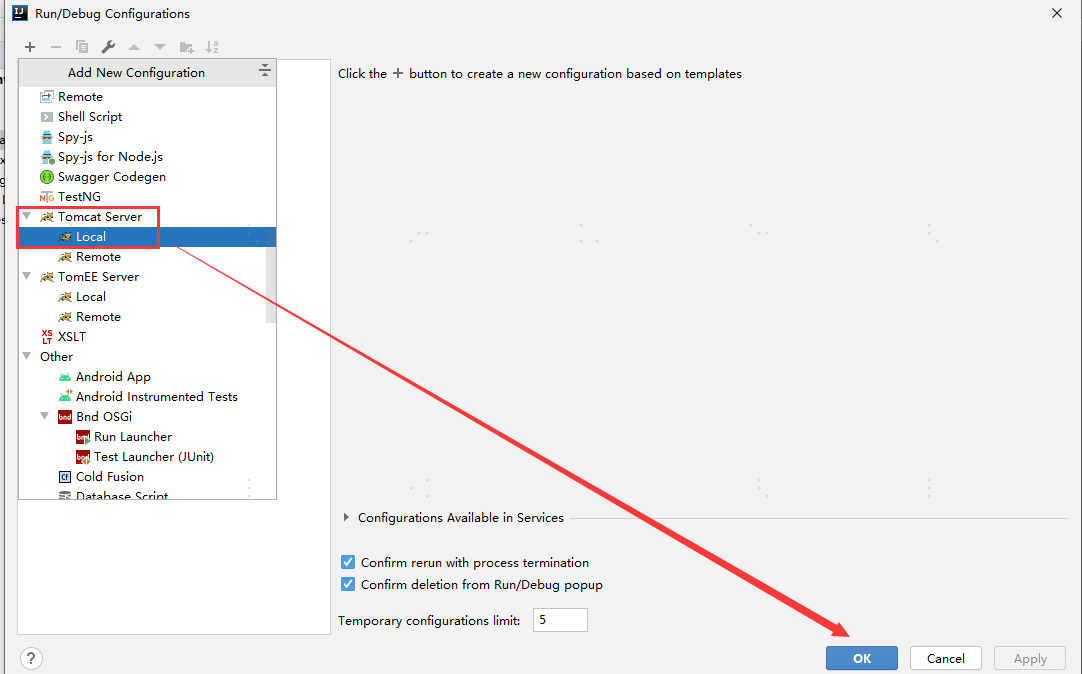




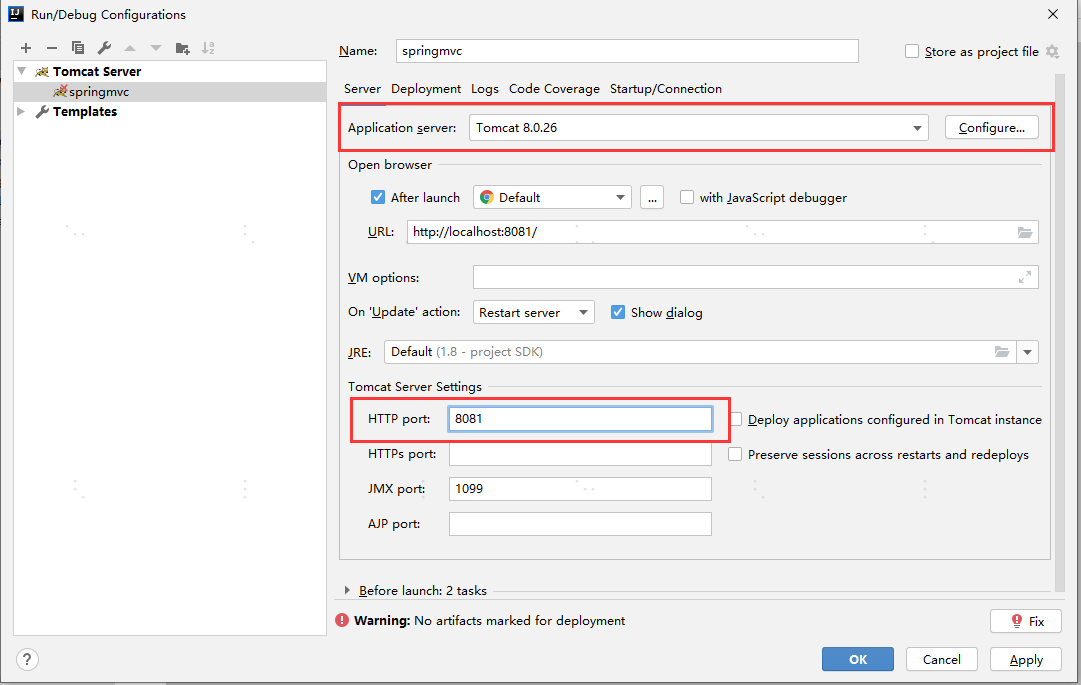
## 4.配置Tomcat

* 创建Tomcat

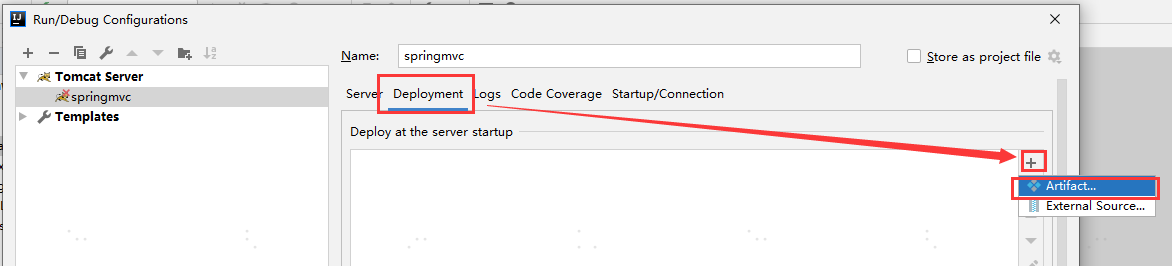


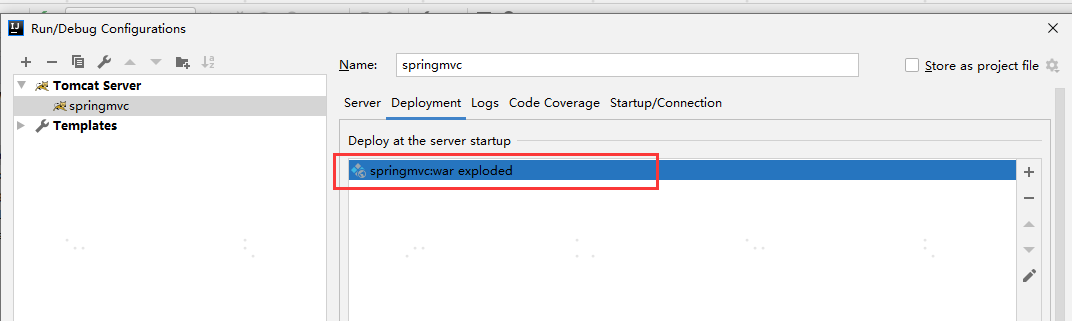


* 配置tomcat，使用本地tomcat
* 配置启动端口



* 选择发布包





* 启动tomcat



## 备注：

1. 如果要修改上下文，可以在发布配置上进行修改

