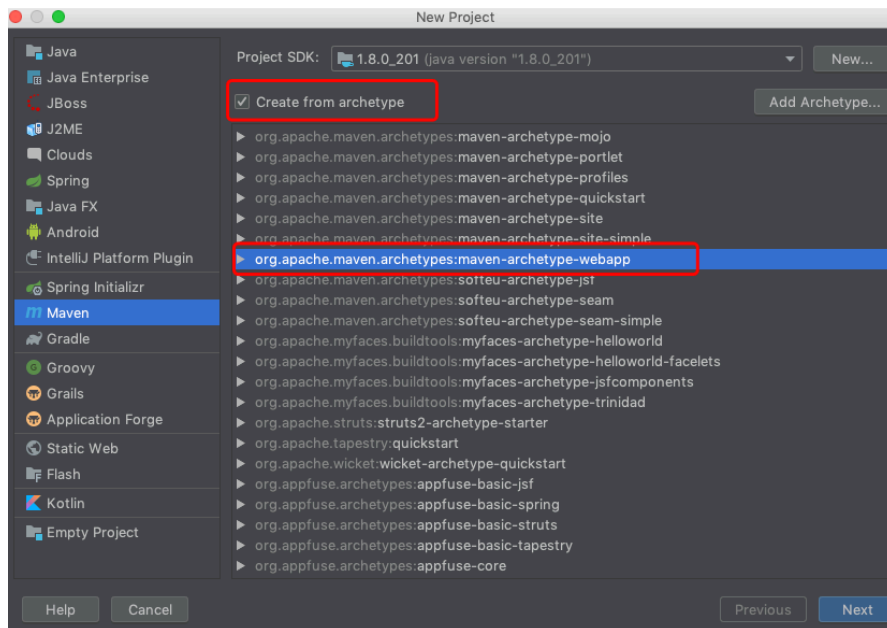
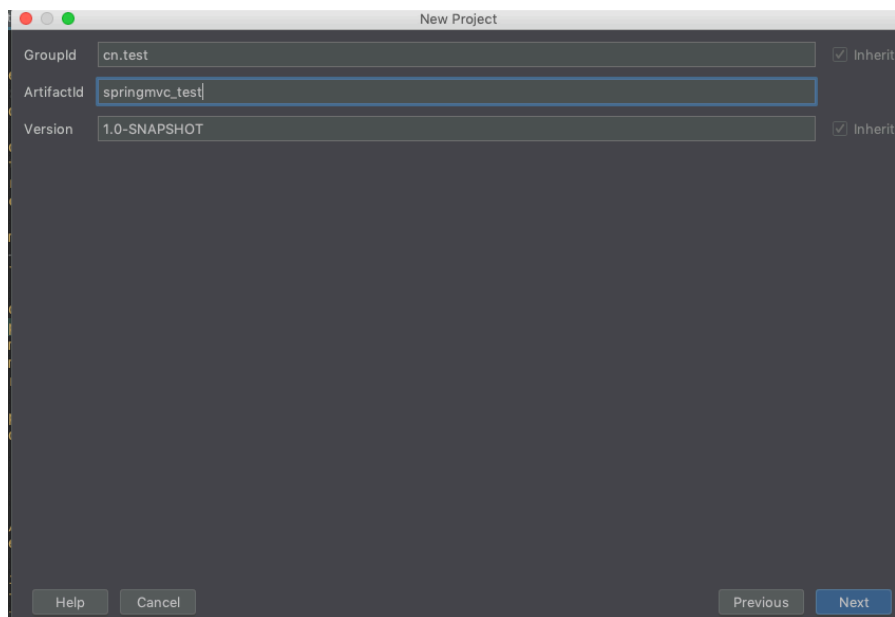


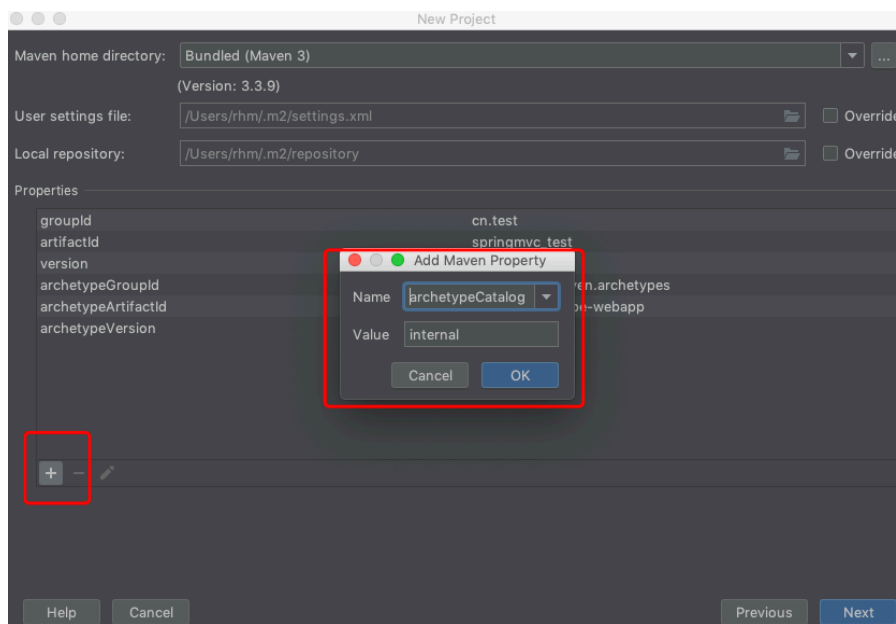
创建maven+springMvc项目



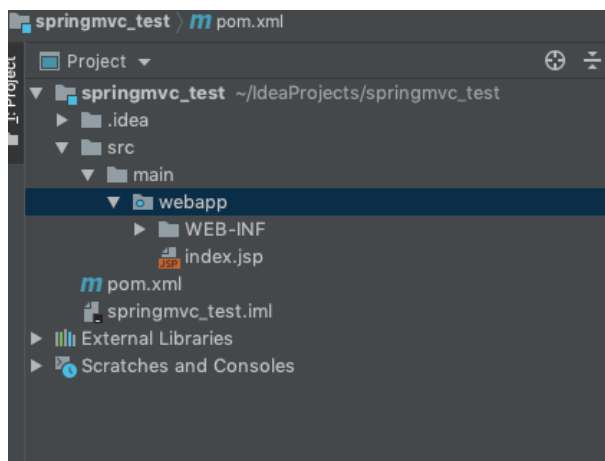
然后下一步



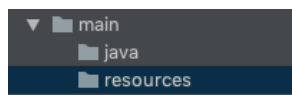
为了解决maven加载过慢的问题，添加 archetypeCatalog = internal



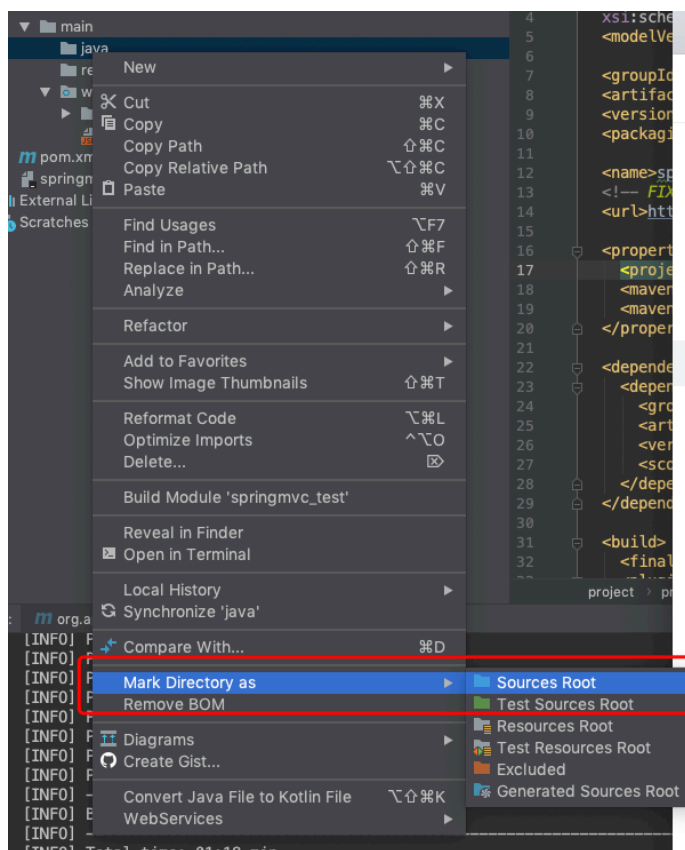
等一会儿创建成功后，目录显示为：



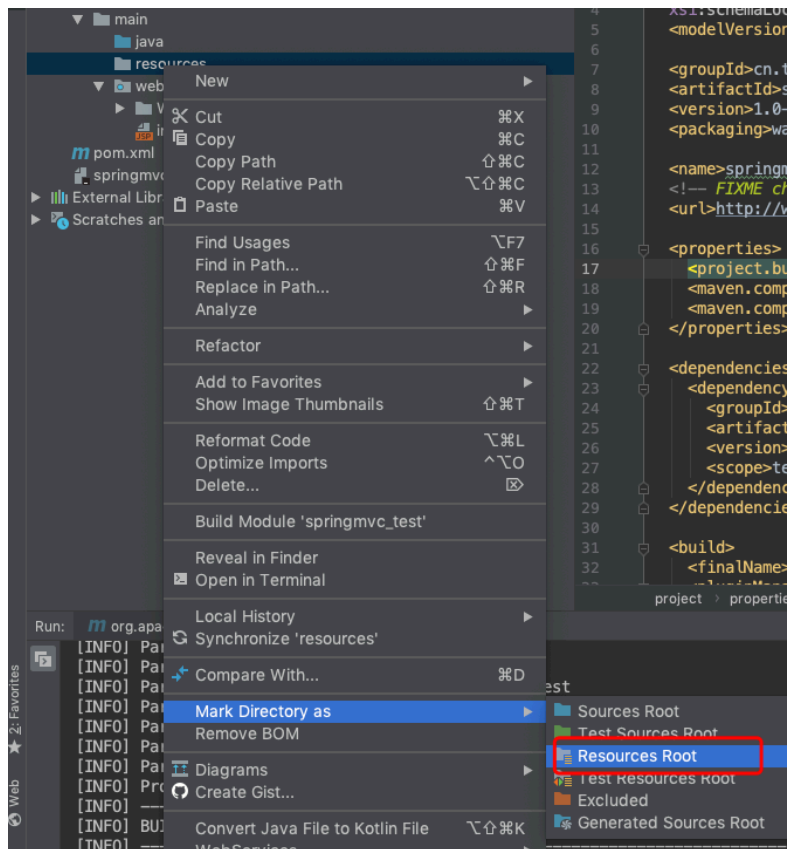
然后创建两个文件夹



让java文件夹变为sources root



设置resources为resources root



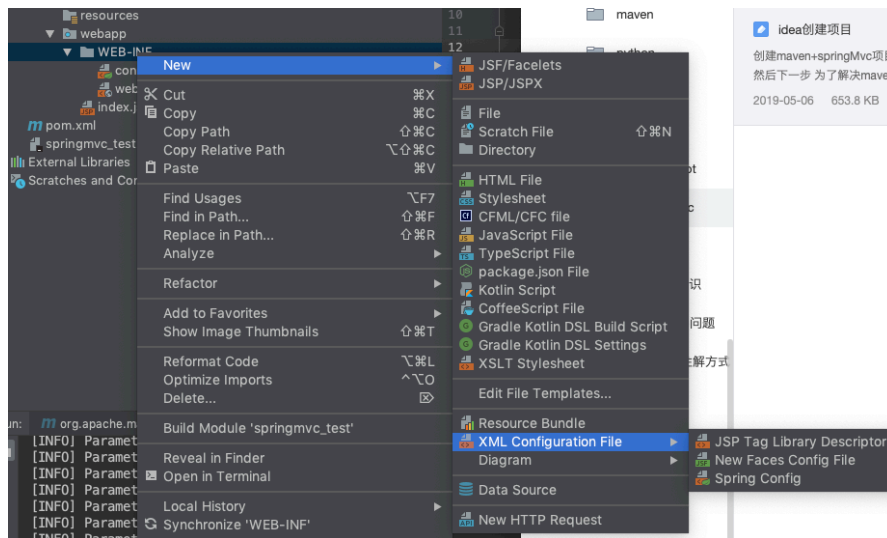
添加spring依赖

```

1  <!-- 添加spring工程依赖-->
2  <dependency>
3      <groupId>org.springframework</groupId>
4      <artifactId>spring-context</artifactId>
5      <version>4.3.13.RELEASE</version>
6  </dependency>
7
8  <dependency>
9      <groupId>org.springframework</groupId>
10     <artifactId>spring-webmvc</artifactId>
11     <version>4.3.13.RELEASE</version>
12 </dependency>

```

然后便可以添加spring配置文件



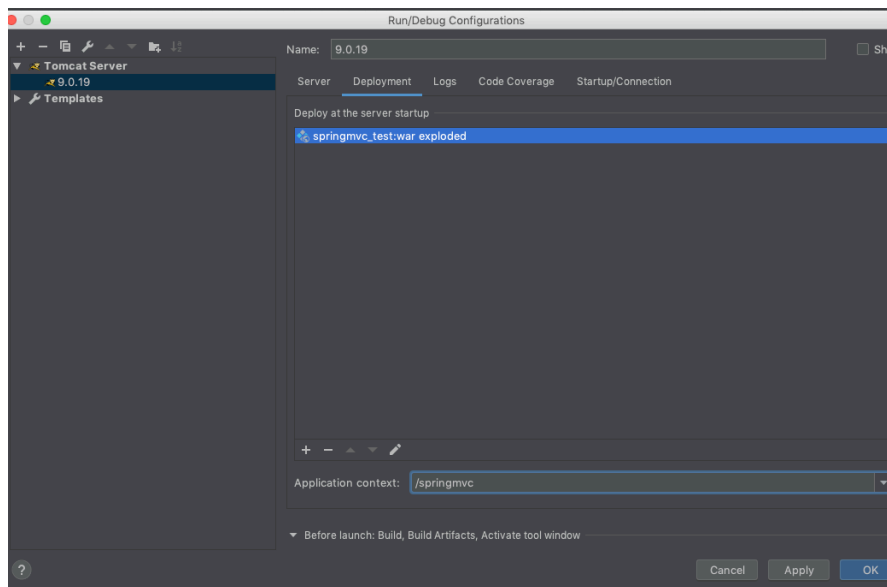
我们在web.xml中添加前端控制器，并让此Servlet在服务器启动时启动。这个前端控制器可以拦截所有Servlet请求

```

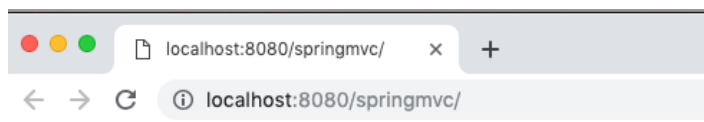
1 <!-- 配置springmvc dispatcherServlet,前端控制器-->
2 <servlet>
3     <servlet-name>springmvc</servlet-name>
4     <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
5
6     <init-param>
7         <param-name>contextConfigLocation</param-name>
8         <param-value>WEB-INF/context.xml</param-value>
9     </init-param>
10
11     <load-on-startup>1</load-on-startup>
12 </servlet>
13
14 <!-- 拦截器配置，拦截所有请求-->
15 <servlet-mapping>
16     <servlet-name>springmvc</servlet-name>
17     <url-pattern>/</url-pattern>
18 </servlet-mapping>

```

配置tomcat



此时运行



Hello World!

添加视图控制器，success表示 success.jsp页面

```
1 package cn.rhm.controller;  
2  
3 import org.springframework.stereotype.Controller;  
4 import org.springframework.web.bind.annotation.RequestMapping;  
5  
6 @Controller  
7 public class HelloController {  
8  
9     @RequestMapping("/hello")  
10    public String sayHello() {  
11        return "success";  
12    }  
13 }
```

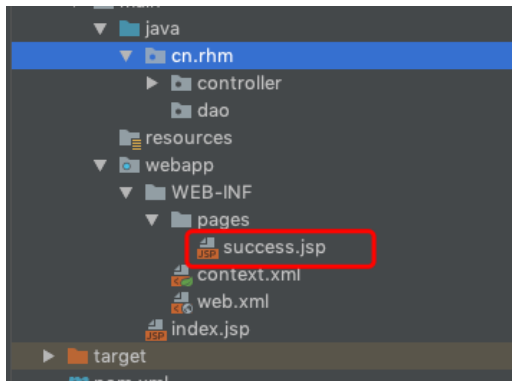
接着添加spring配置,配置视图解析器，即配置视图的路径，让控制器能够打开指定的jsp页面

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

```
<context:component-scan base-package="cn.rhm"></context:component-scan>
```

配置视图解析器-->

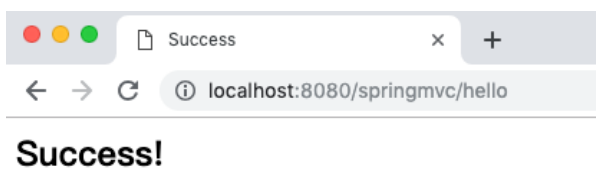
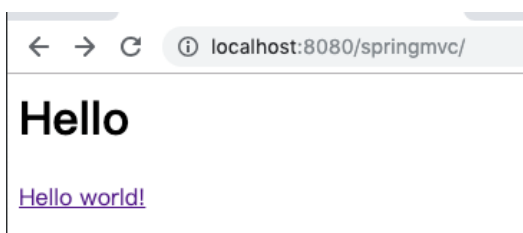
```
<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
  <property name="prefix" value="/WEB-INF/pages/"></property>
  <property name="suffix" value=".jsp"></property>
</bean>
```



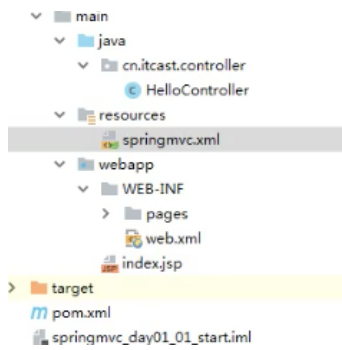
修改index.jsp

```
1 <body>
2 <h1>Hello</h1>
3
4 <a href="hello">Hello world!</a>
5 </body>
```

点击链接便打开 success页面。



注意：我们原先配置了resources的文件夹，可以用来放置spring的配置文件



那么我们在web.xml中声明时，使用的是classpath:

```
1 <param-value>classpath:springmvc.xml</param-value>
```

构建流程图

