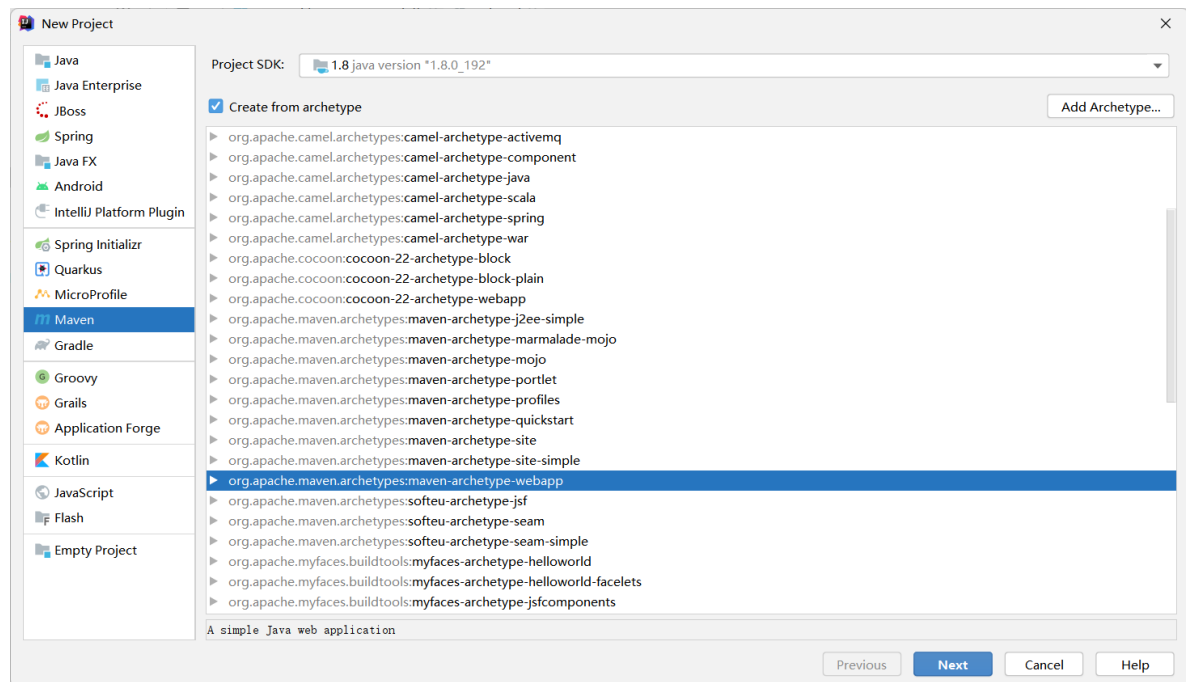


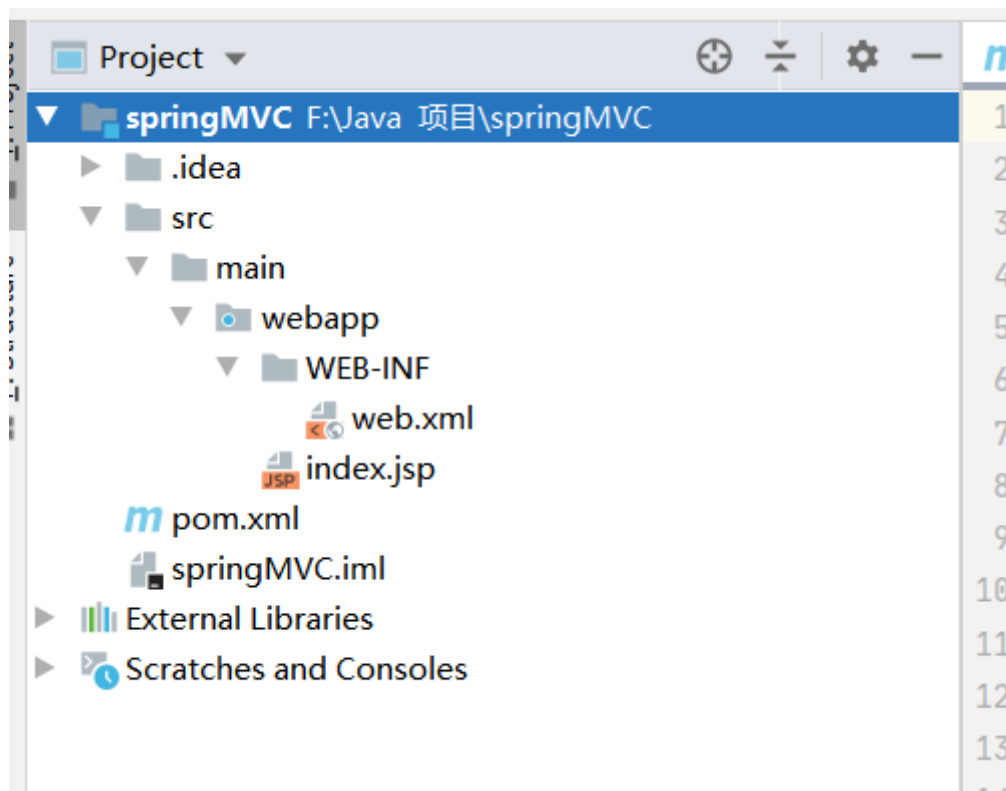
Maven 创建 Servlet 项目的过程

一、创建maven项目，引入web-app模板，创建项目源代码目录

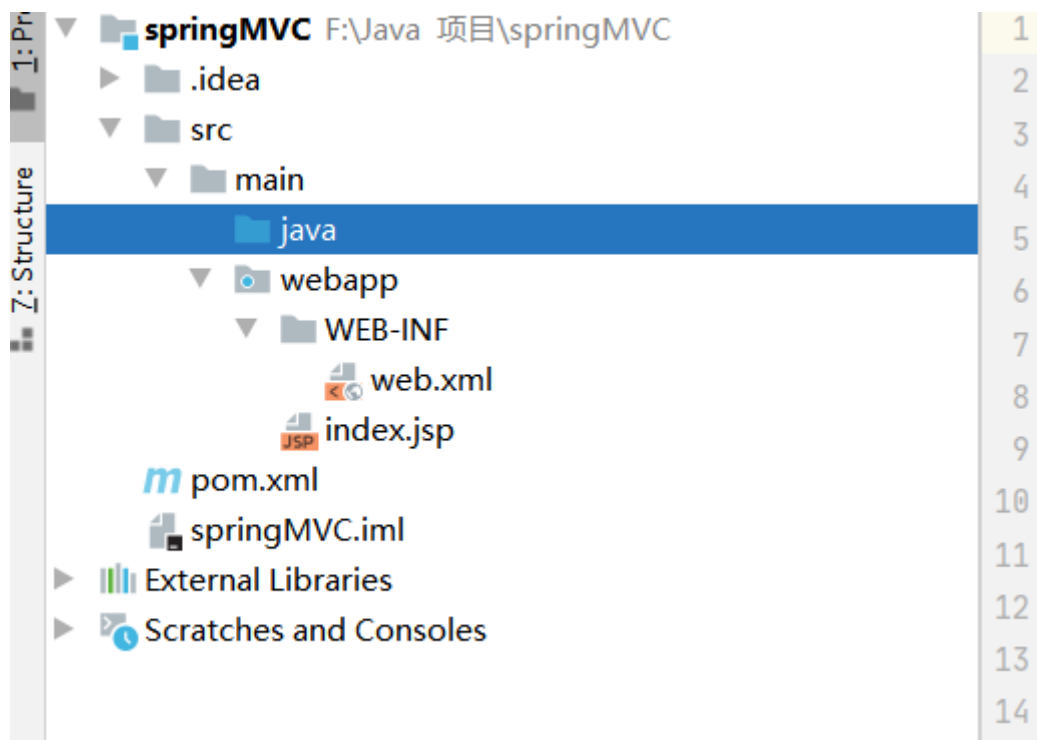
maven搭建项目，使用web-app模板



初始目录结构



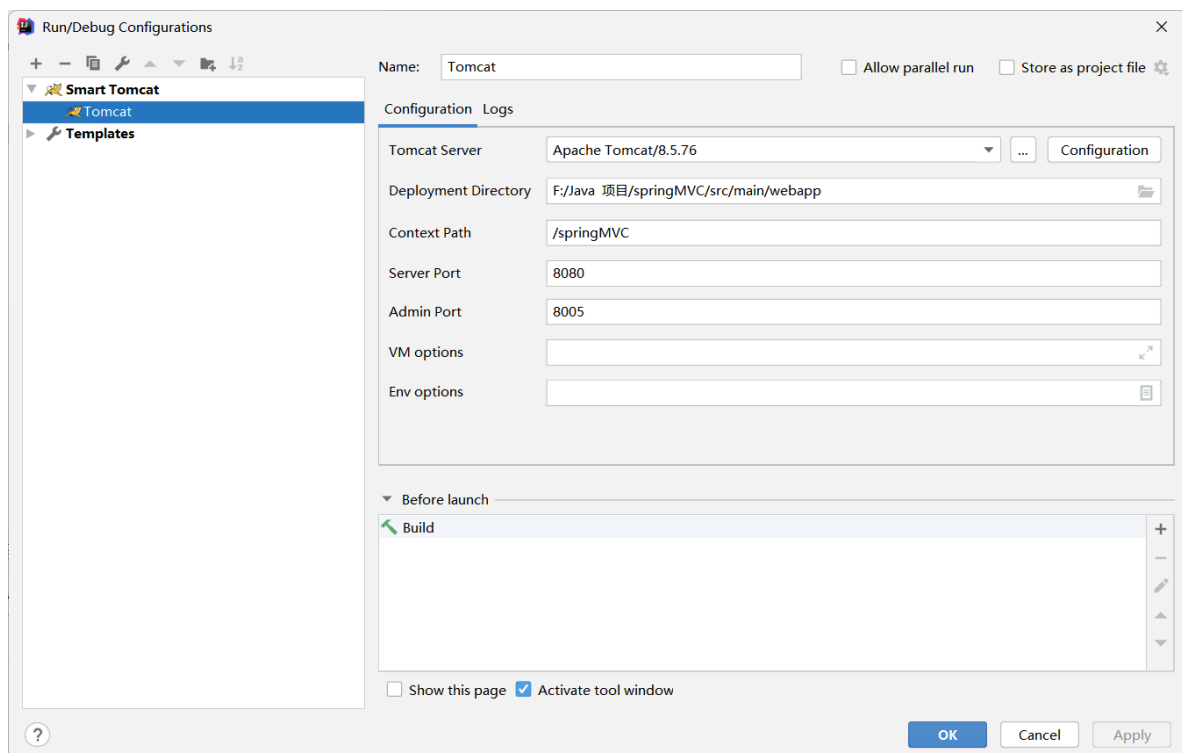
添加项目的源文件的根目录java



二、根据Tomcat 版本添加对应版本的servlet依赖，引入Tomcat插件

Servlet Spec	JSP Spec	EL Spec	WebSocket Spec	Authentication (JASPIC) Spec	Apache Tomcat Version	Latest Released Version	Supported Java Versions
6.0	3.1	5.0	2.1	3.0	10.1.x	10.1.0-M11 (alpha)	11 and later
5.0	3.0	4.0	2.0	2.0	10.0.x	10.0.17	8 and later
4.0	2.3	3.0	1.1	1.1	9.0.x	9.0.59	8 and later
3.1	2.3	3.0	1.1	1.1	8.5.x	8.5.76	7 and later
3.1	2.3	3.0	1.1	N/A	8.0.x (superseded)	8.0.53 (superseded)	7 and later
3.0	2.2	2.2	1.1	N/A	7.0.x (archived)	7.0.109 (archived)	6 and later (7 and later for WebSocket)
2.5	2.1	2.1	N/A	N/A	6.0.x (archived)	6.0.53 (archived)	5 and later
2.4	2.0	N/A	N/A	N/A	5.5.x (archived)	5.5.36 (archived)	1.4 and later
2.3	1.2	N/A	N/A	N/A	4.1.x (archived)	4.1.40 (archived)	1.3 and later
2.2	1.1	N/A	N/A	N/A	3.3.x (archived)	3.3.2 (archived)	1.1 and later

加入smart Tomcat 插件，配置本地的tomcat到项目中



在pom.xml文件中加入 servlet依赖

```

1  <!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
2  <dependency>
3      <groupId>javax.servlet</groupId>
4      <artifactId>javax.servlet-api</artifactId>
5      <version>3.1.0</version>
6      <scope>provided</scope>
7  </dependency>
8

```

三、配置servlet配置文件和路由文件

1、在web.xml中定义servlet类以及映射路由,下面是模板

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
3         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
5                             http://xmlns.jcp.org/xml/ns/javaee/web-
6 app_3_1.xsd"
7         version="3.1"
8         metadata-complete="true">
9     <servlet>
10         <servlet-name>hello</servlet-name>
11         <servlet-class>HelloServlet</servlet-class>
12     </servlet>
13
14     <servlet-mapping>
15         <servlet-name>hello</servlet-name>
16         <url-pattern>/hello</url-pattern>
17     </servlet-mapping>
18
19
20 </web-app>
```

2、或者使用注解 @WebServlet("/hello") 定义映射路由

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
3         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
5                             http://xmlns.jcp.org/xml/ns/javaee/web-
6 app_3_1.xsd"
7         version="3.1"
8         metadata-complete="false">
9
10 </web-app>
```

四、写servlet类的代码

继承HttpServlet, 重写doGet方法, 确定返回的resp的类型、编码、状态码

```
1 import javax.servlet.ServletException;
2 import javax.servlet.annotation.WebServlet;
3 import javax.servlet.http.HttpServlet;
4 import javax.servlet.http.HttpServletRequest;
```

```
5 import javax.servlet.http.HttpServletResponse;
6 import java.io.IOException;
7
8 @WebServlet("/hello")
9 public class HelloServlet extends HttpServlet {
10     @Override
11     protected void doGet(HttpServletRequest req, HttpServletResponse resp)
12     throws ServletException, IOException {
13         //1. 获取参数（参数校验）
14         String name = req.getParameter("name");
15
16         // 2. 访问数据库
17         // 伪代码不写了
18
19         //3. 将组装好的数据返回给前端
20         resp.setContentType("text/html");
21         resp.setStatus(200);
22         resp.setCharacterEncoding("utf8");
23         resp.getWriter().write(name);
24     }
25
26     @Override
27     protected void doPost(HttpServletRequest req, HttpServletResponse resp)
28     throws ServletException, IOException {
29         this.doGet(req, resp);
30     }
31 }
```

五、运行tomcat服务器，在url进行访问以及传参

