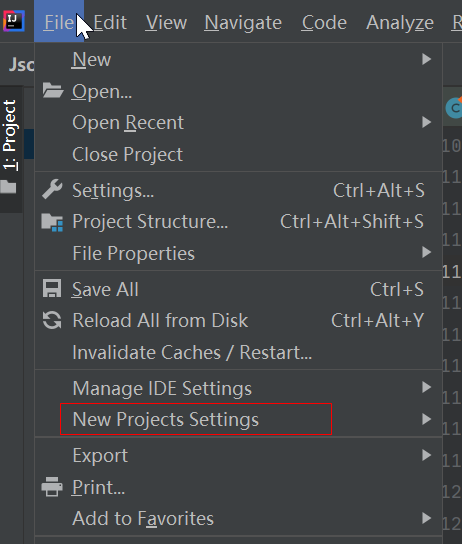
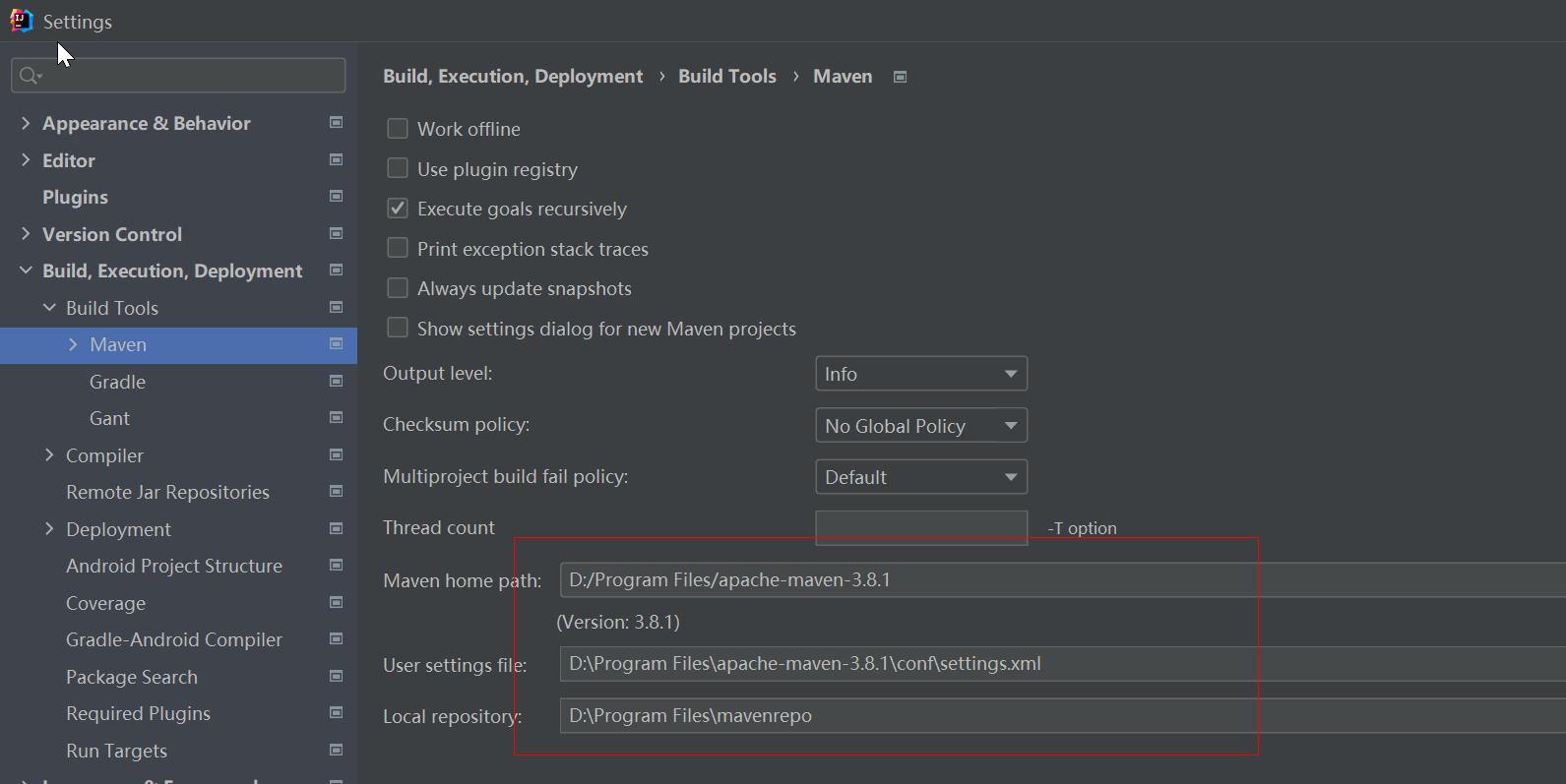
## 创建一个mavn web工程

### 第一步：创建项目

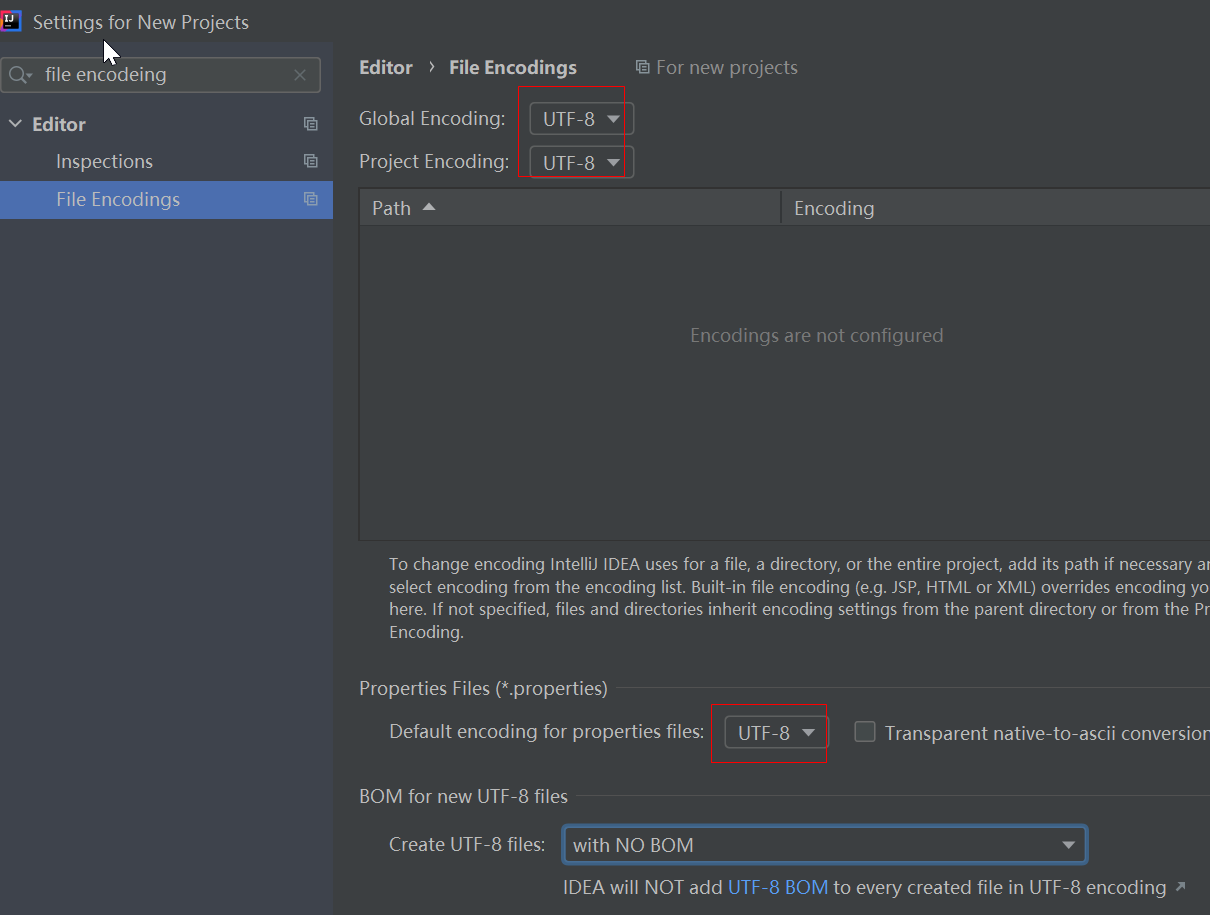
在IDEA中 New Projects Settings中，设置新工程对应的maven配置等



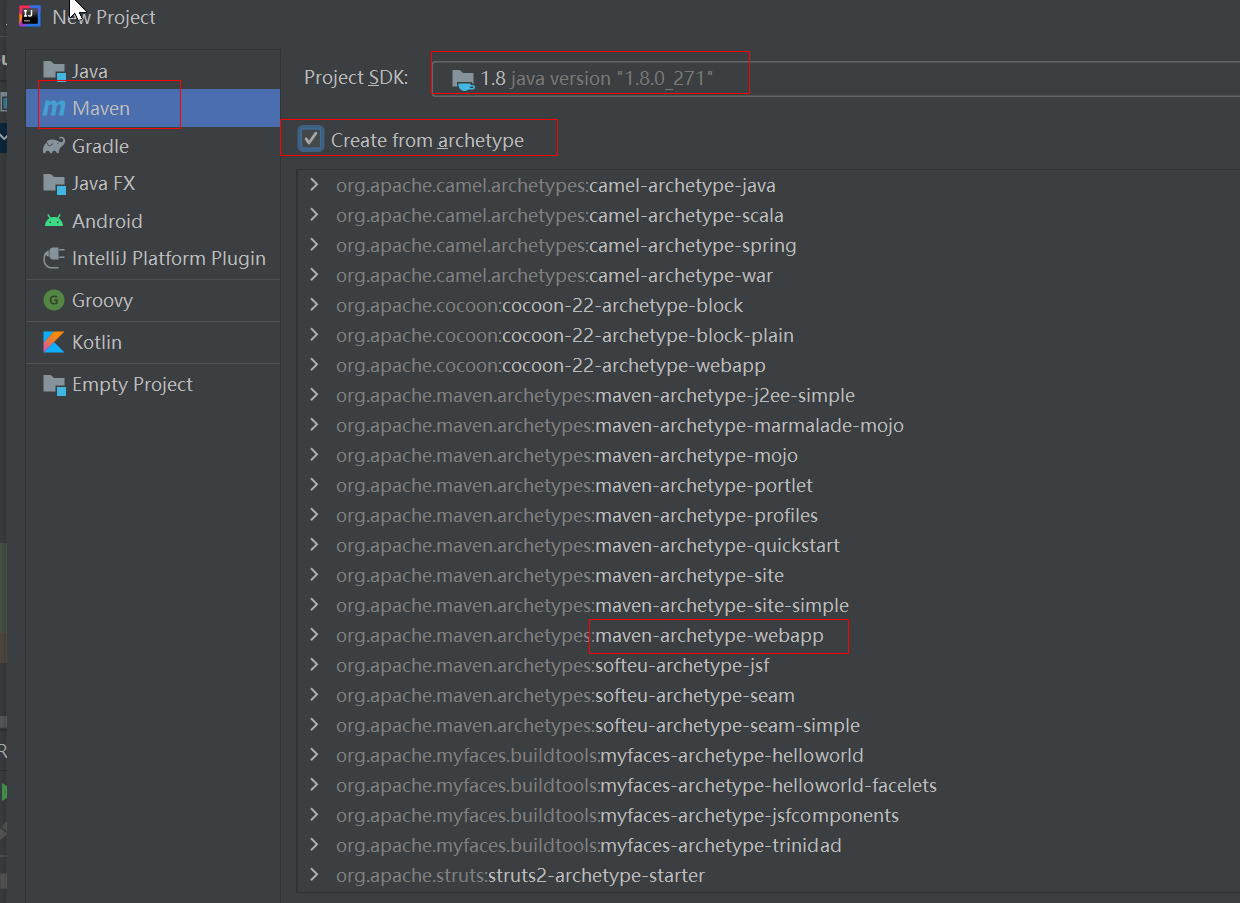
1)设置好maven工程，Maven home path不包含bin，创建web工程后需要1-2分钟加载资源

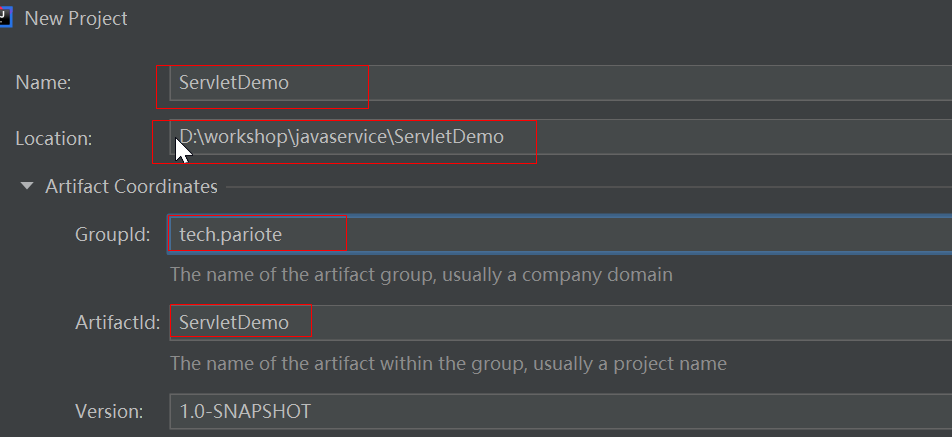


2)设置编码字符集

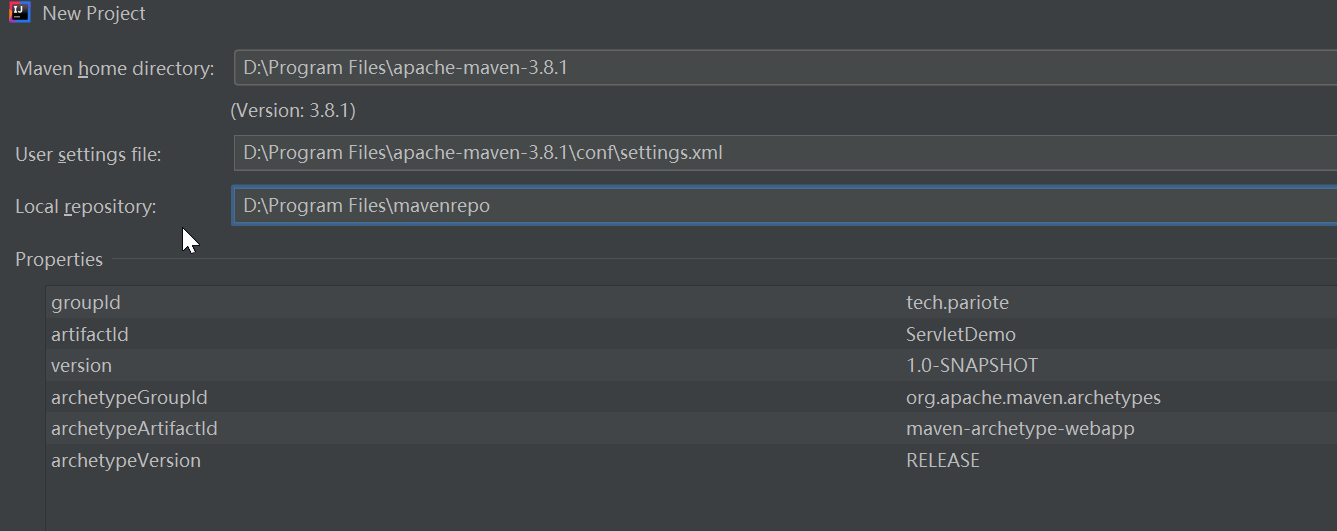


Maven模板选择org.apache.maven.archetypes.maven-archetype-webapp

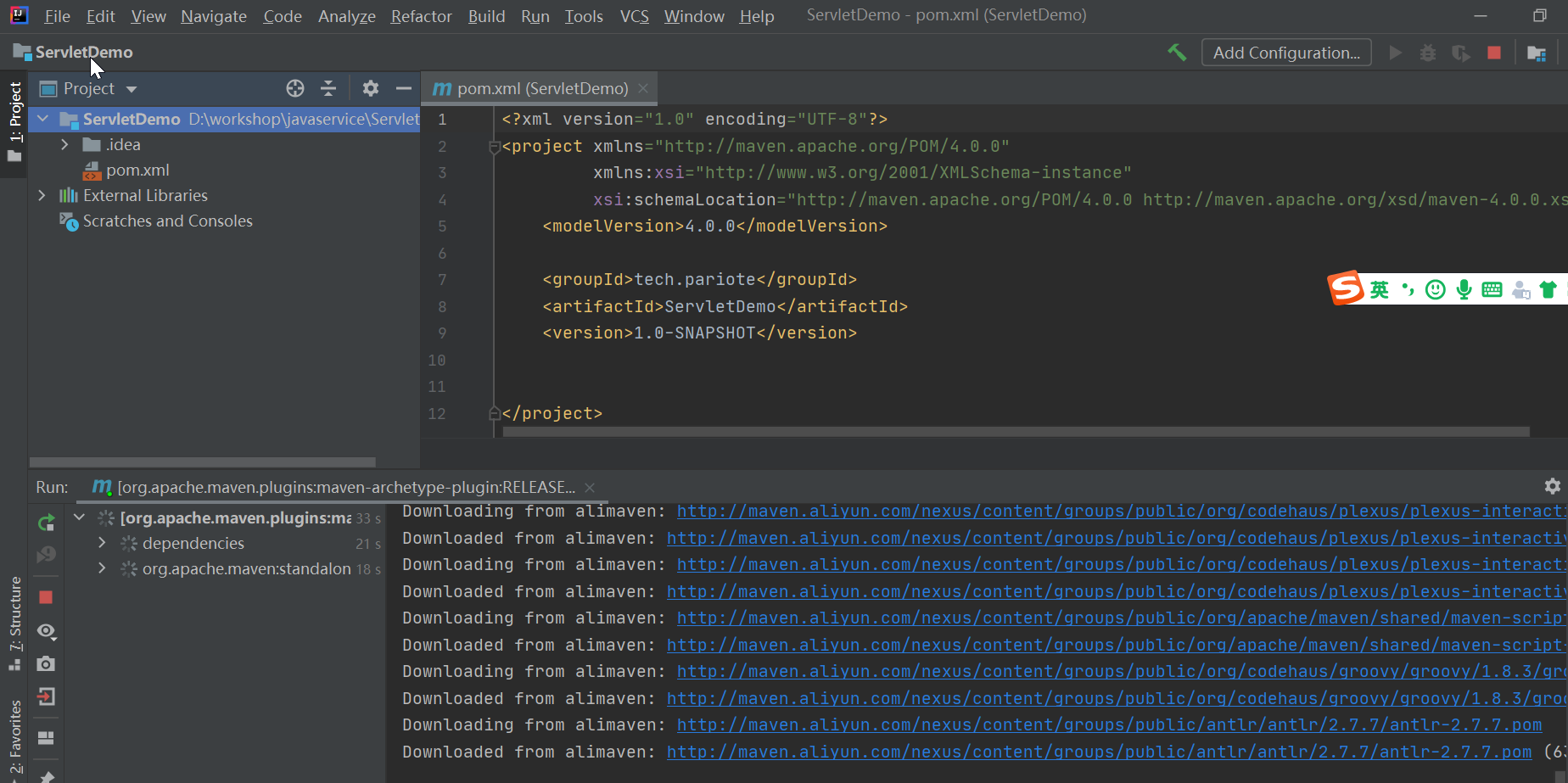




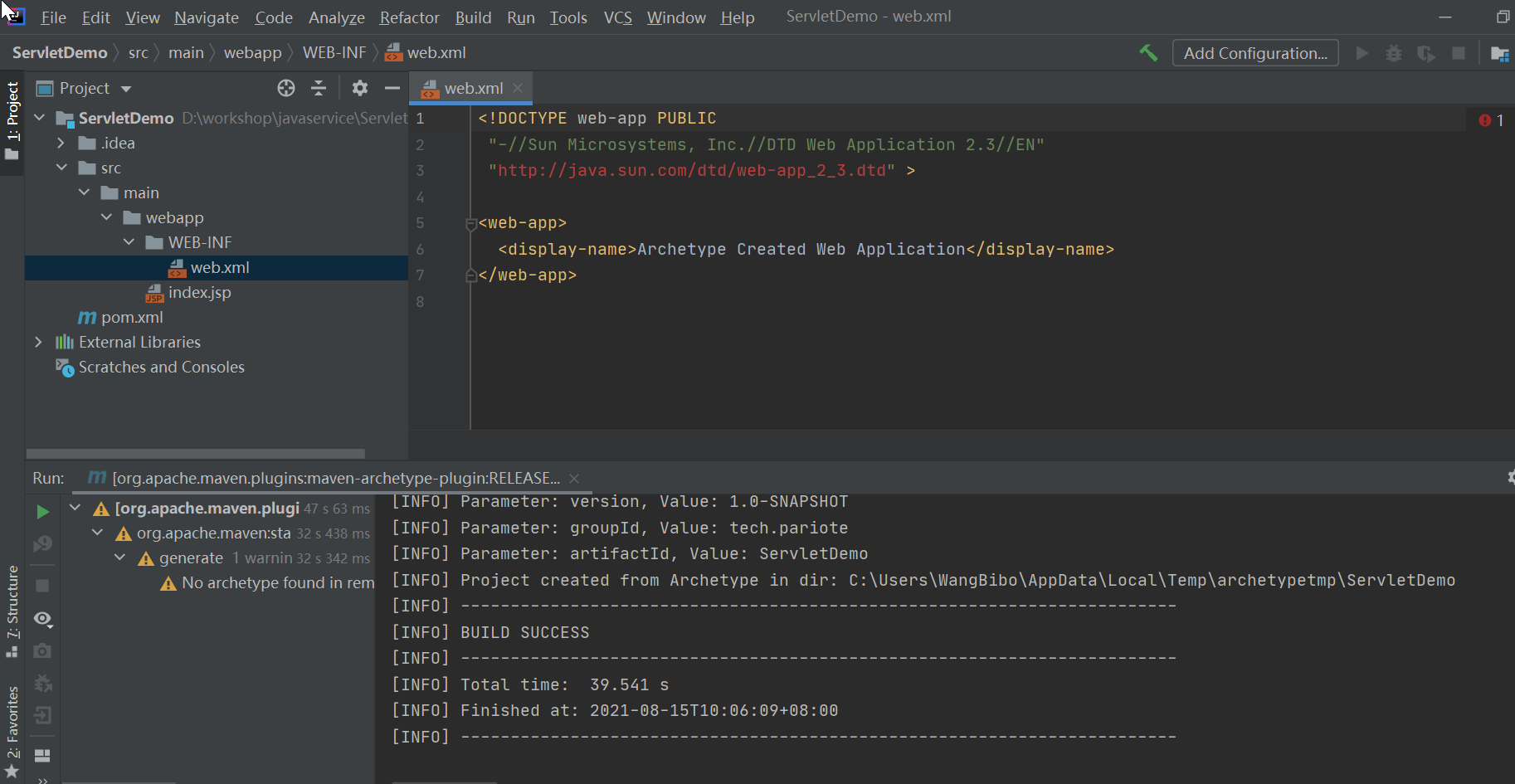
设置maven相关设置



需要下载一大堆依赖，耐心等待，网络和maven配置有问题可能导致失败，根据报错处理。



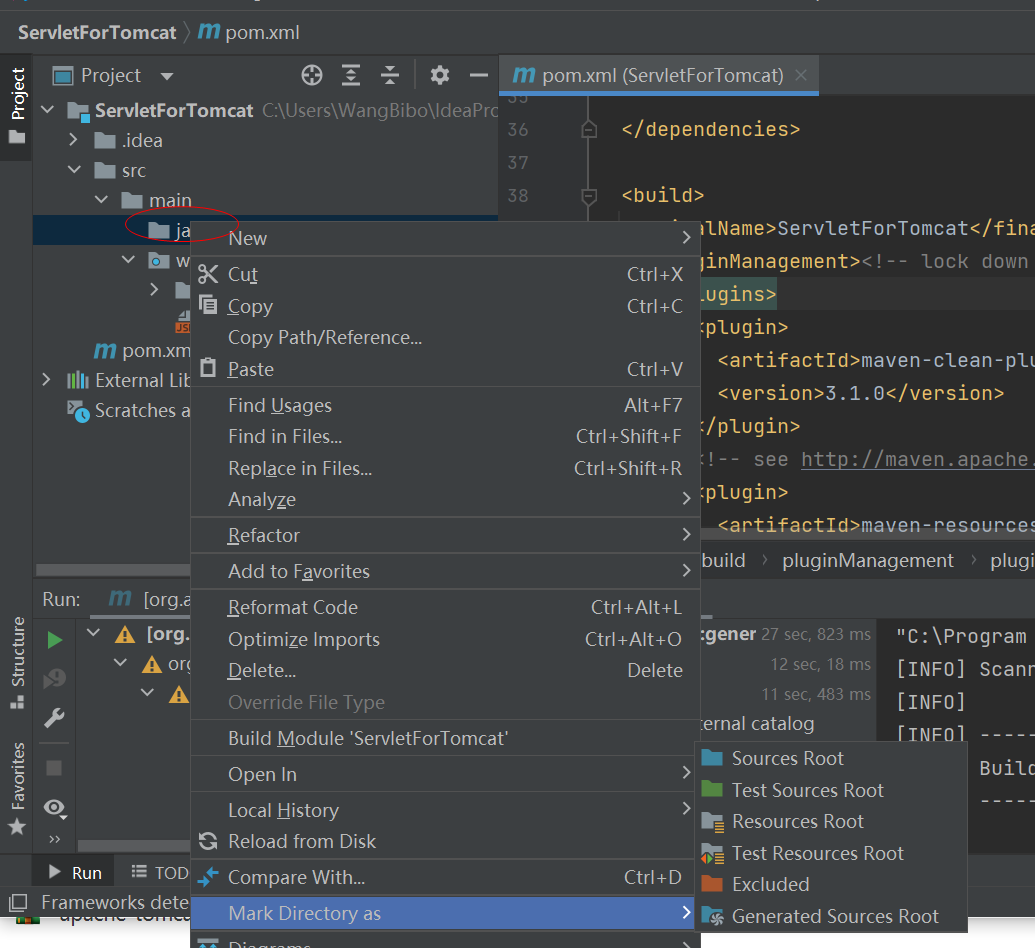
初始化结束目录结构



如果本地仓库中某些插件找不到，可以在pom中<dependencies>增加一个依赖，使用maven reload 下载下来。

<dependency>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-site-plugin</artifactId>  
 <version>3.3</version>  
</dependency>

创建一个java 目录后需要标记为Source Root后（目录颜色是蓝色），才能才能创建package



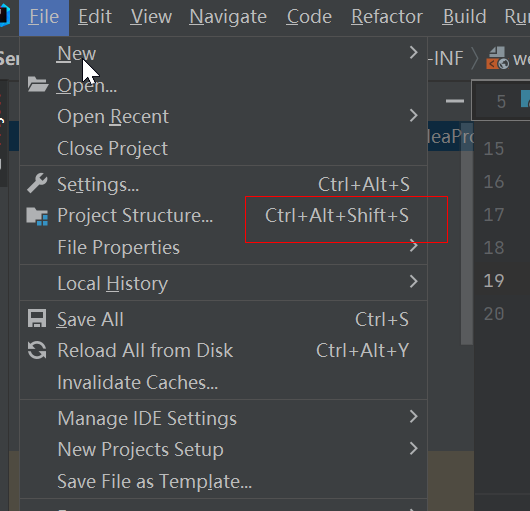
### 第二步，设置module facts artifacts等并打包一个tomcat下可以运行的程序

* 利用IDEA工具编译打包 build和maven打包没有任何关系，是两个不同的体系。IDEA编译打包可以配合tomcat进行本地调试，而maven主要是流水线打包和部署等相关，功能强大。
* IDEA社区版功能有限，不直接支持tomcat，facts支持有限，无法配置tomcat本地调试环境。正版30天可以试用，试用期间享受正版功能。

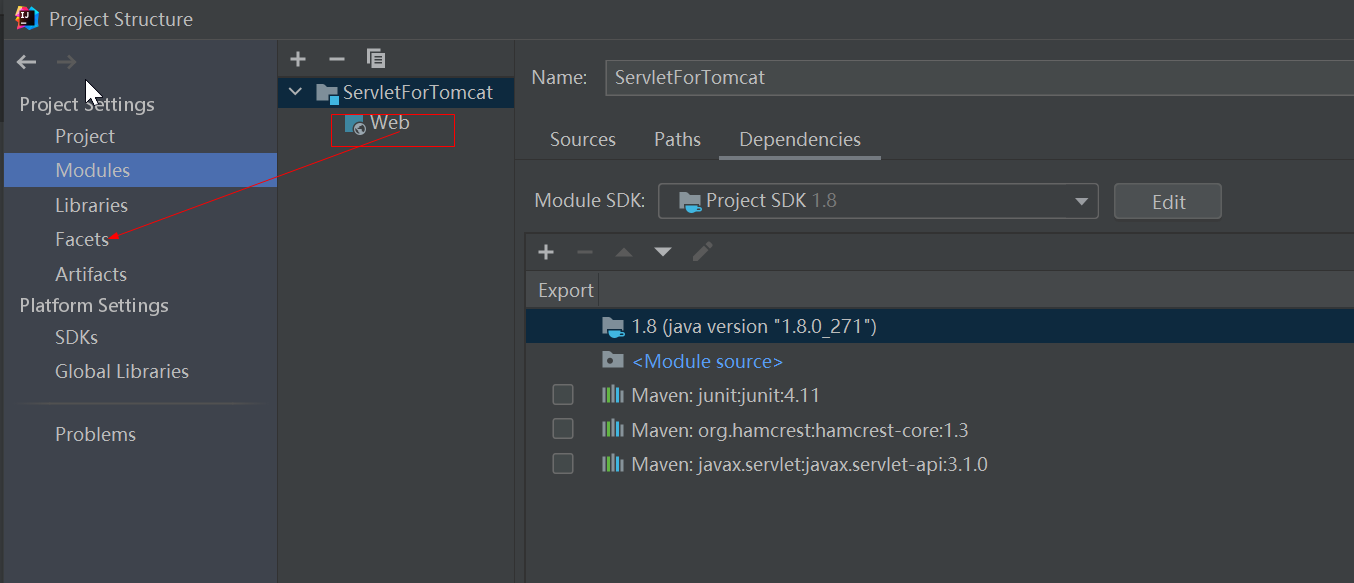
下载激活码方法：（idea知道你是谁在试用破解版本，个人试用促进了idea生态。企业试用破解就要被维权，收取费用）

<https://blog.csdn.net/weixin_43866709/article/details/88593704>

Ctr+Shift+Alt+S打开Project Structure设置面板

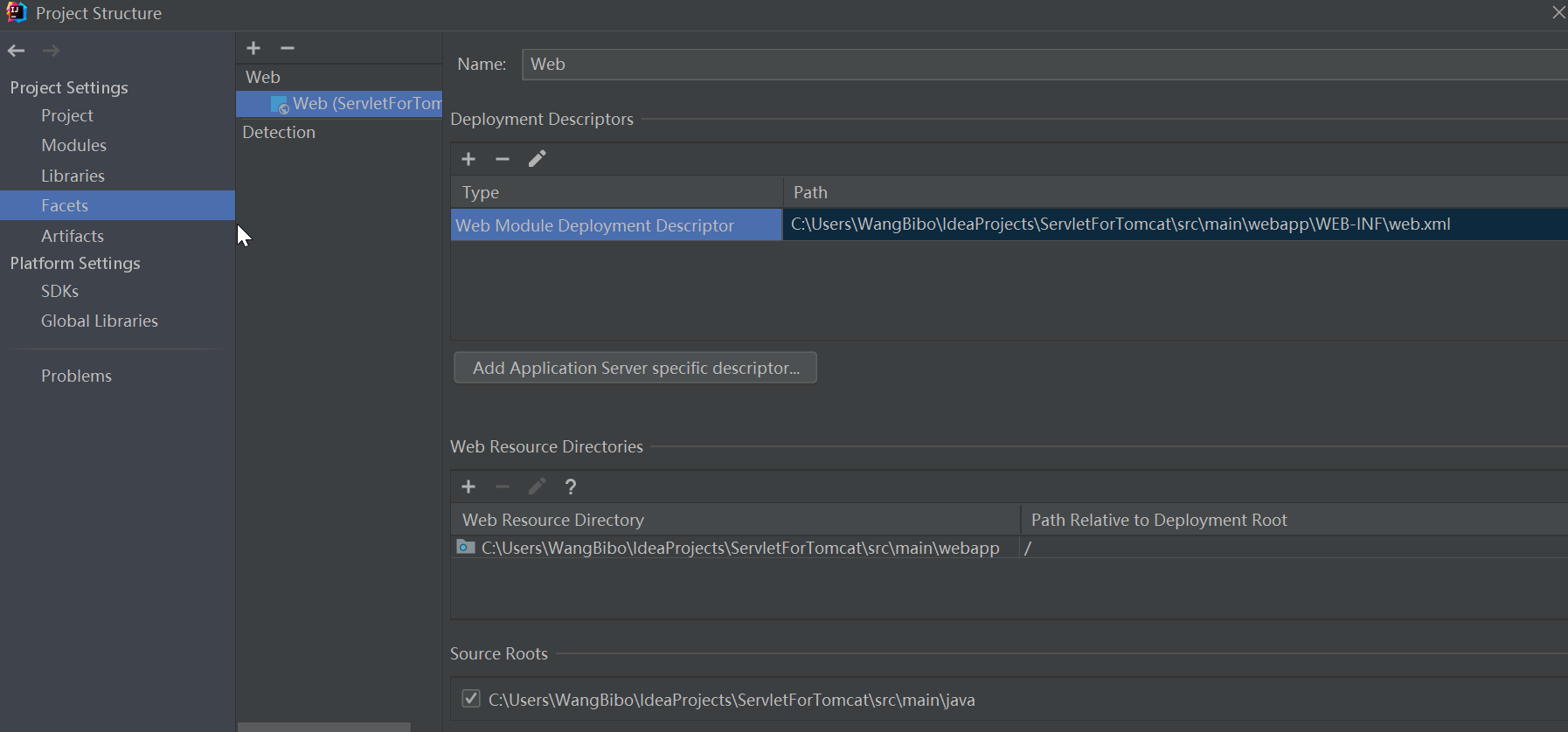


Module中的Web就是下面Facts中要配置的，这里可以设置Sources Paths Dependencies等项目，可以看到module具备的Facets.



Facts设置，其向IDE表明当前工程的类型，以便IDE以适当形式处理对应的资源。

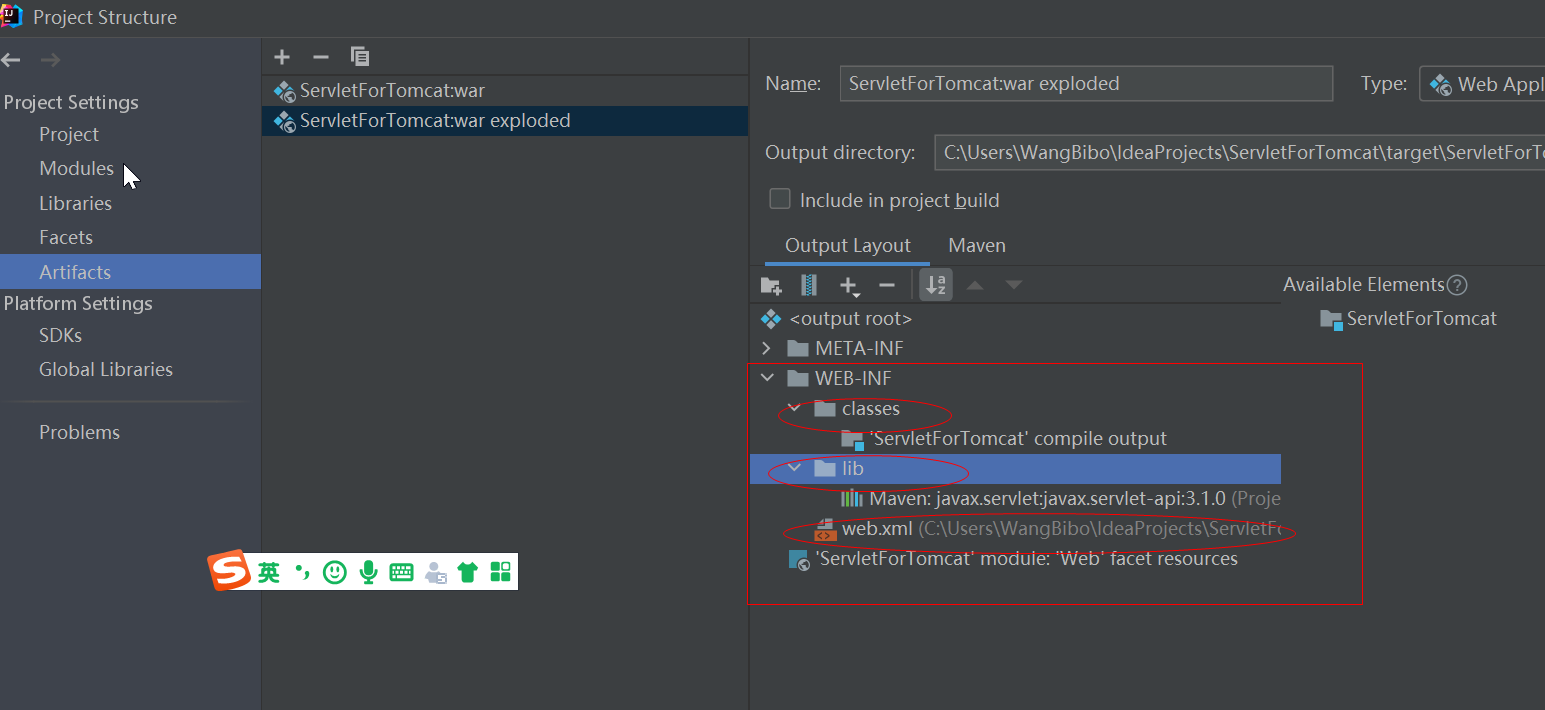
最好创建项目是就选定好maven和web工程类型后，IDEA自动生成，会比人工配置少些坑。类型迁移以创建新工程复制代码和配置的方式更好。



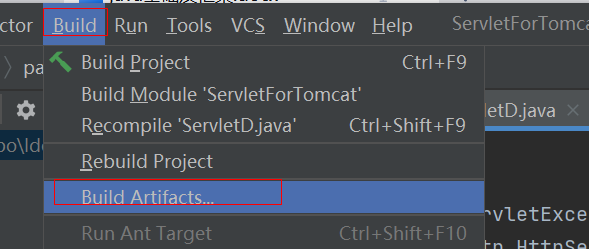
配置artifacts,在WEB-INF目录下引入web.xml

打包出来的tomcat webapp下可以运行的程序核心目录，WEB-INF下有classes lib web.xml文件。classes是 类和配置文件的根目录，lib是依赖，web.xml是应用配置文件.支持添加文件目录等功能

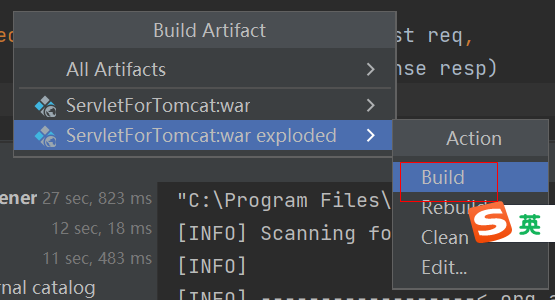
Output directory 是编译后生成文件的目录。



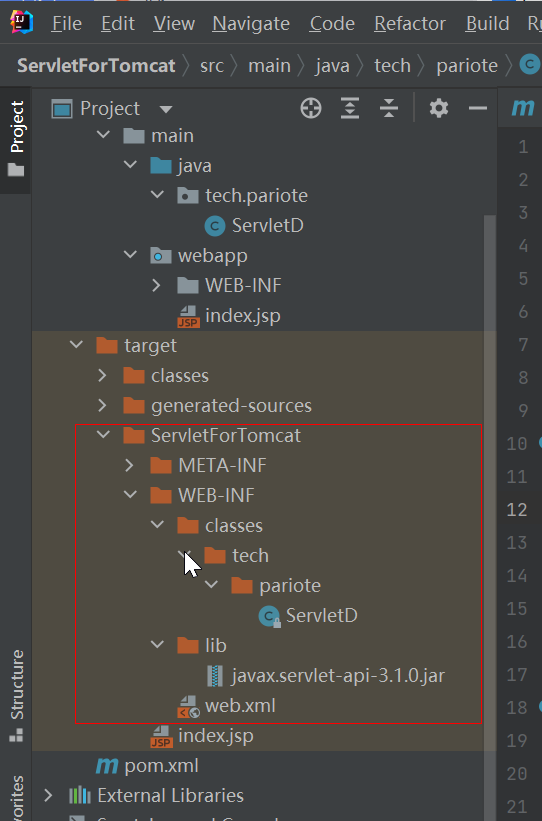
编译Build -> Buid ARtifacts...打一个war-explored 包



选择编译



编译后生成程序在target目录下对应的目录下。



### 第三步 程序编写

tech.pariote.ServletD类试下一个基本的servlet

|  |
| --- |
| package tech.pariote; import javax.servlet.ServletException; import javax.servlet.http.HttpServlet; import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse; public class ServletD extends HttpServlet {  protected void doGet(HttpServletRequest req,  HttpServletResponse resp)  throws ServletException,  java.io.IOException{  resp.getWriter().println("ninhao,get");  resp.getWriter().println("<h1>hhhh</h1>");  }  protected void doPost(HttpServletRequest req,  HttpServletResponse resp)  throws ServletException,  java.io.IOException{  resp.getWriter().println(""+req.getParameter("username"));  } } |

pom.xml中加载对应的servlet依赖

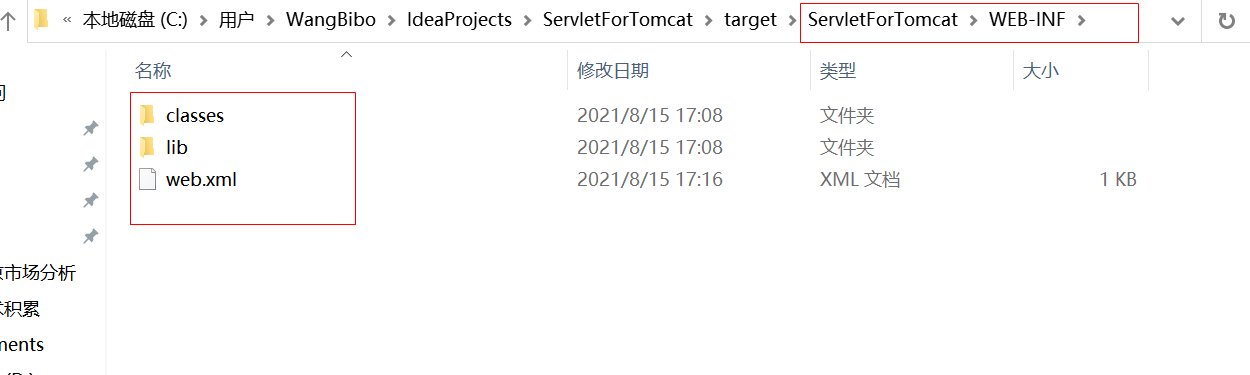
|  |
| --- |
| <dependency>  <groupId>javax.servlet</groupId>  <artifactId>javax.servlet-api</artifactId>  <version>3.1.0</version> </dependency> |

web.xml设置转发规则

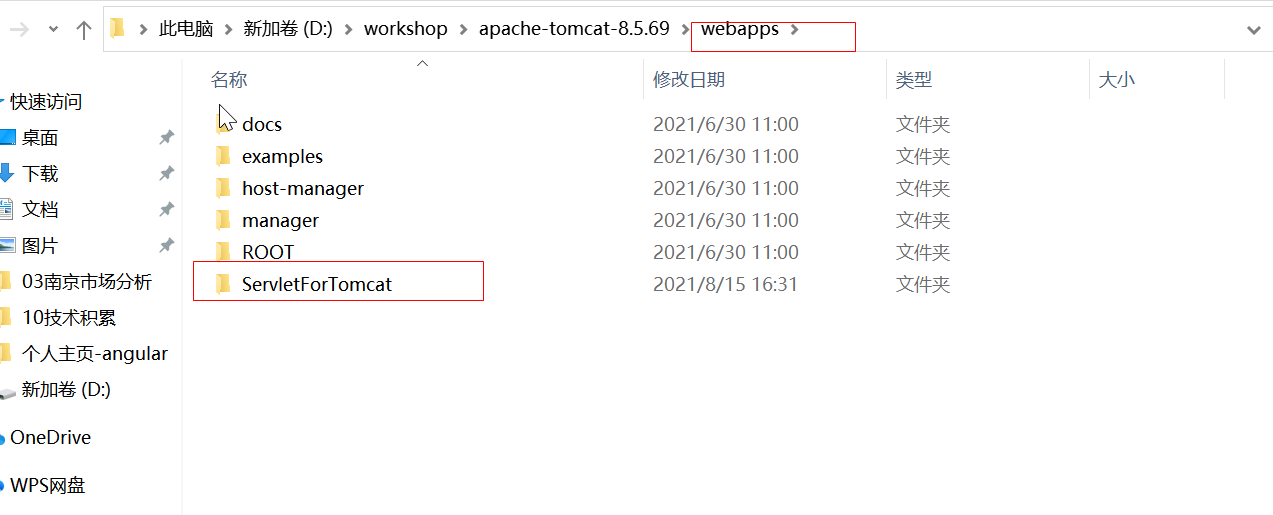
|  |
| --- |
| <!DOCTYPE web-app PUBLIC  "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"  "http://java.sun.com/dtd/web-app\_2\_3.dtd" >  <web-app>  <display-name>Archetype Created Web Application</display-name>  <servlet>  <servlet-name>dispatcher</servlet-name>  <servlet-class>tech.pariote.ServletD</servlet-class>  </servlet>  <servlet-mapping>  <servlet-name>dispatcher</servlet-name>  <url-pattern>/sp</url-pattern>  </servlet-mapping> </web-app> |

### 第四步 配置tomcat本地测试

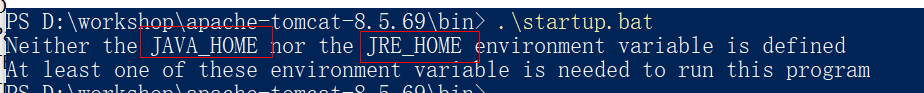
Build对应的Artifacts后得到包



将生成的程序放到tomcat webapps目录下：



启动tomcat

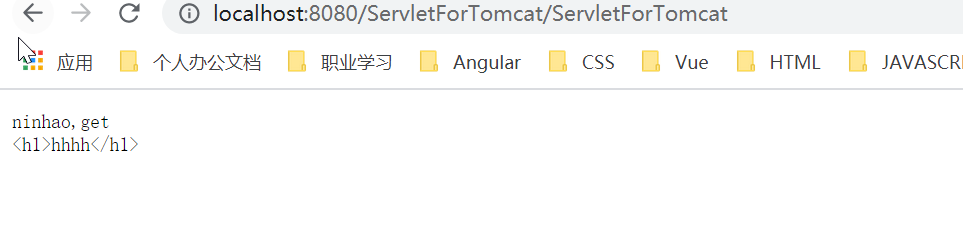


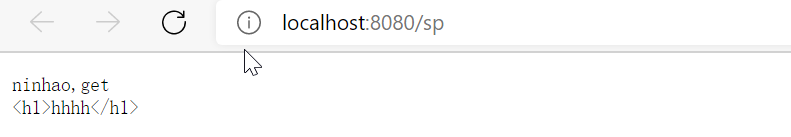
IDEA运行可以从PATH环境变量中找到JDK位置，tomcat运行需要依赖JAVA\_HOME JRE\_HOME

访问连接

<http://localhost:8080/ServletForTomcat/sp>

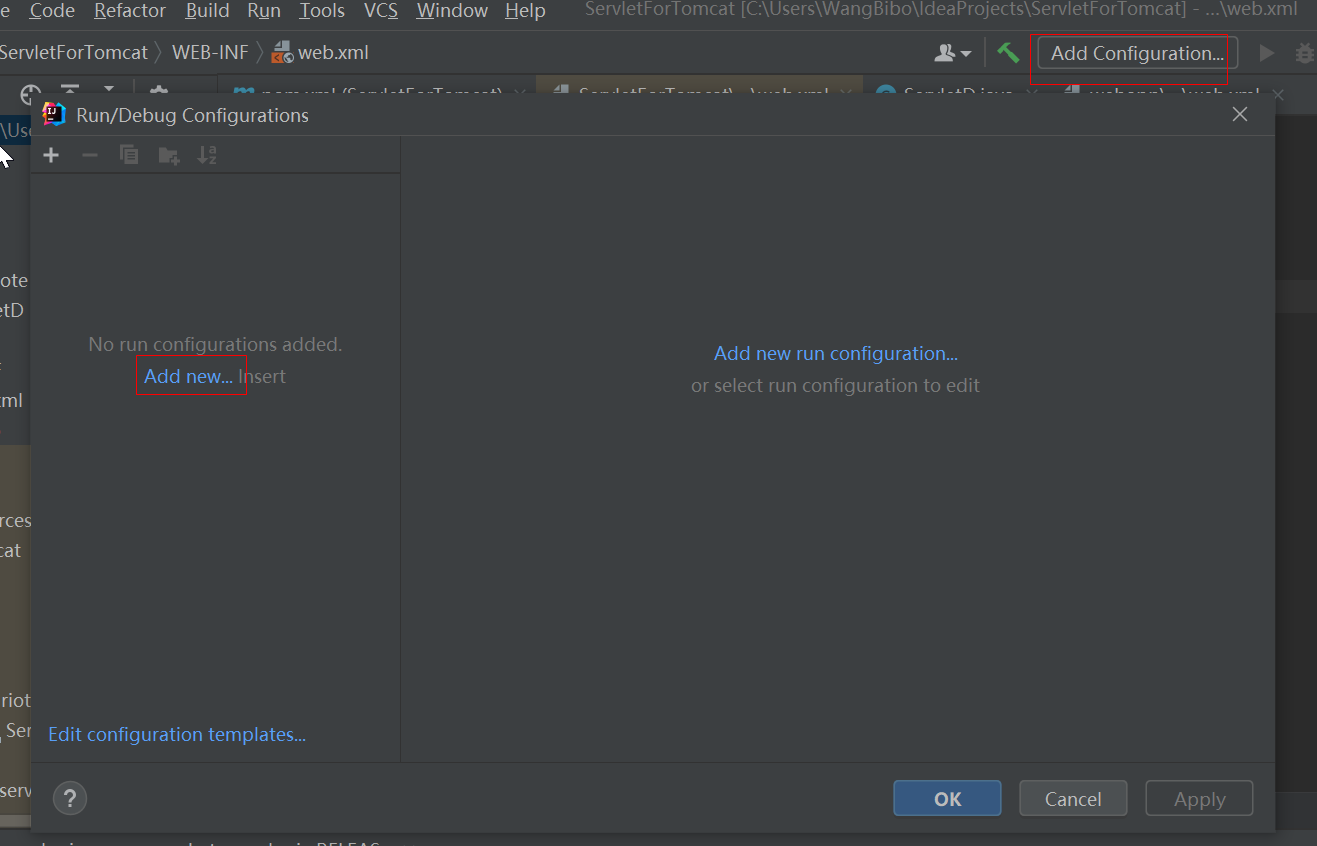
|  |
| --- |
| 在tomcat webapps下部署目录为ServletForTomcat的应用  1、配置/时默认访问目录http://localhost:8080/ServletForTomcat/ServletForTomcat  <servlet-mapping>  <servlet-name>dispatcher</servlet-name>  <url-pattern>/</url-pattern>  </servlet-mapping>    2配置非/时如/sp,默认访问目录http://localhost:8080/ServletForTomcat/sp  <servlet-mapping>  <servlet-name>dispatcher</servlet-name>  <url-pattern>/sp</url-pattern>  </servlet-mapping> |



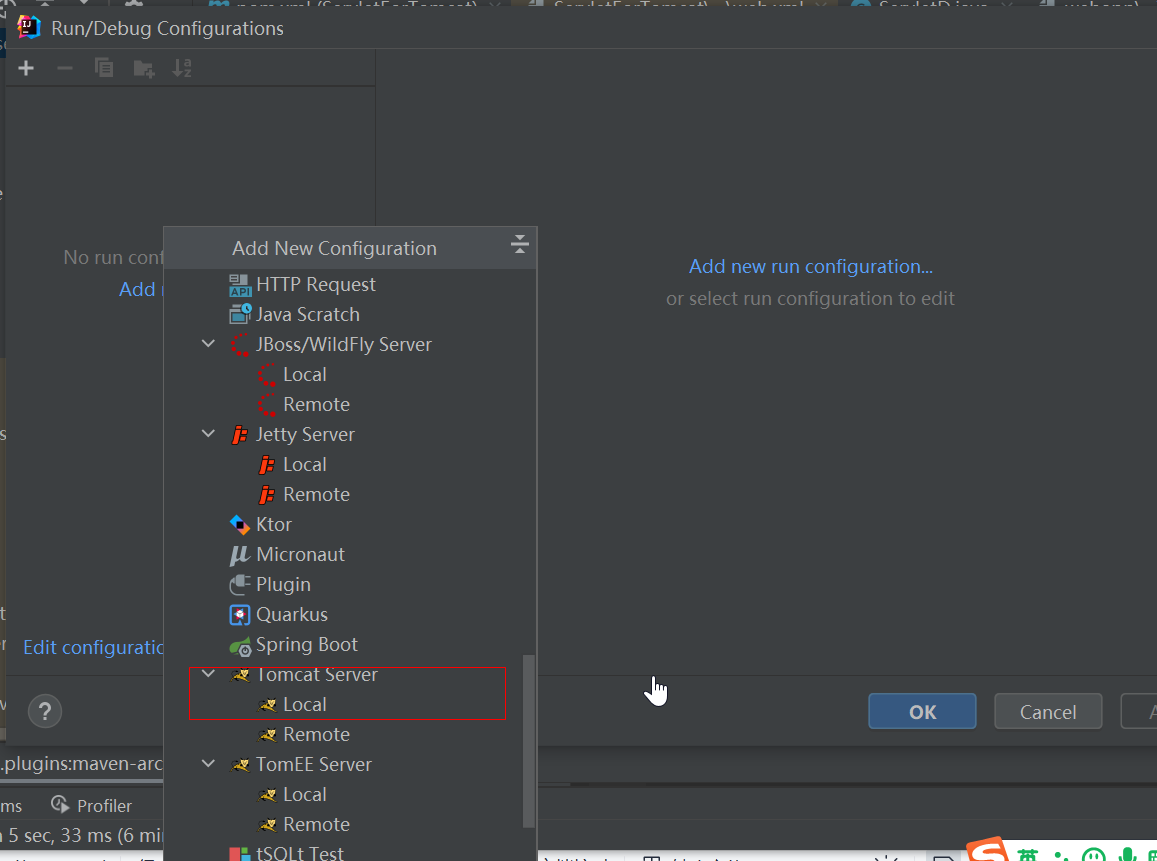


### 第五步 IDEA中配置tomcat并启动测试

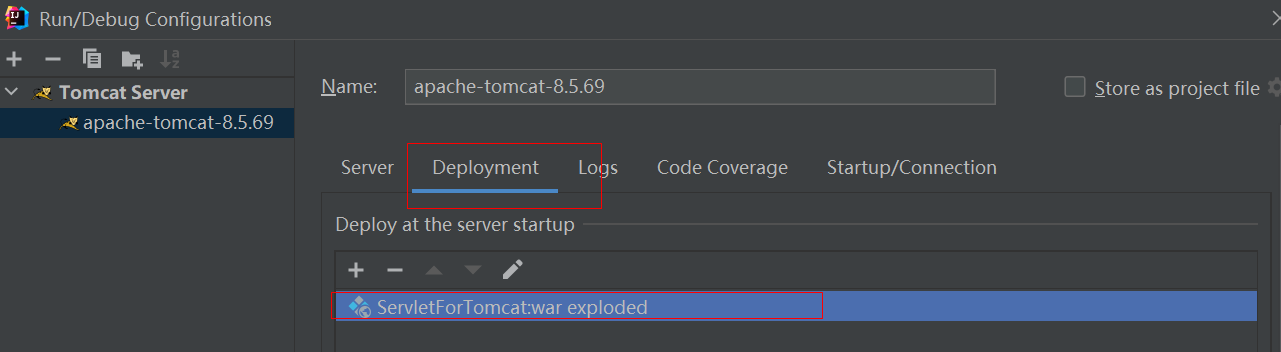
点击环境配置按钮



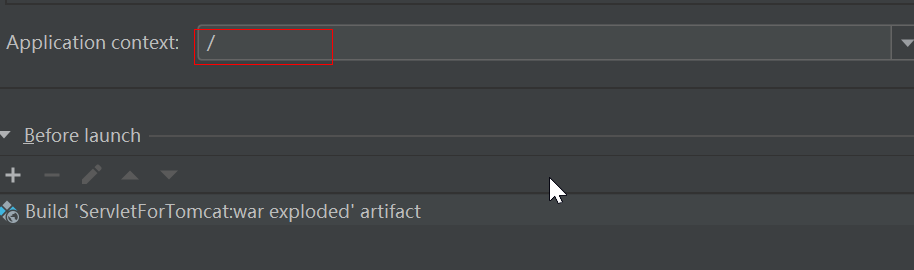
选择Local tomcat容器



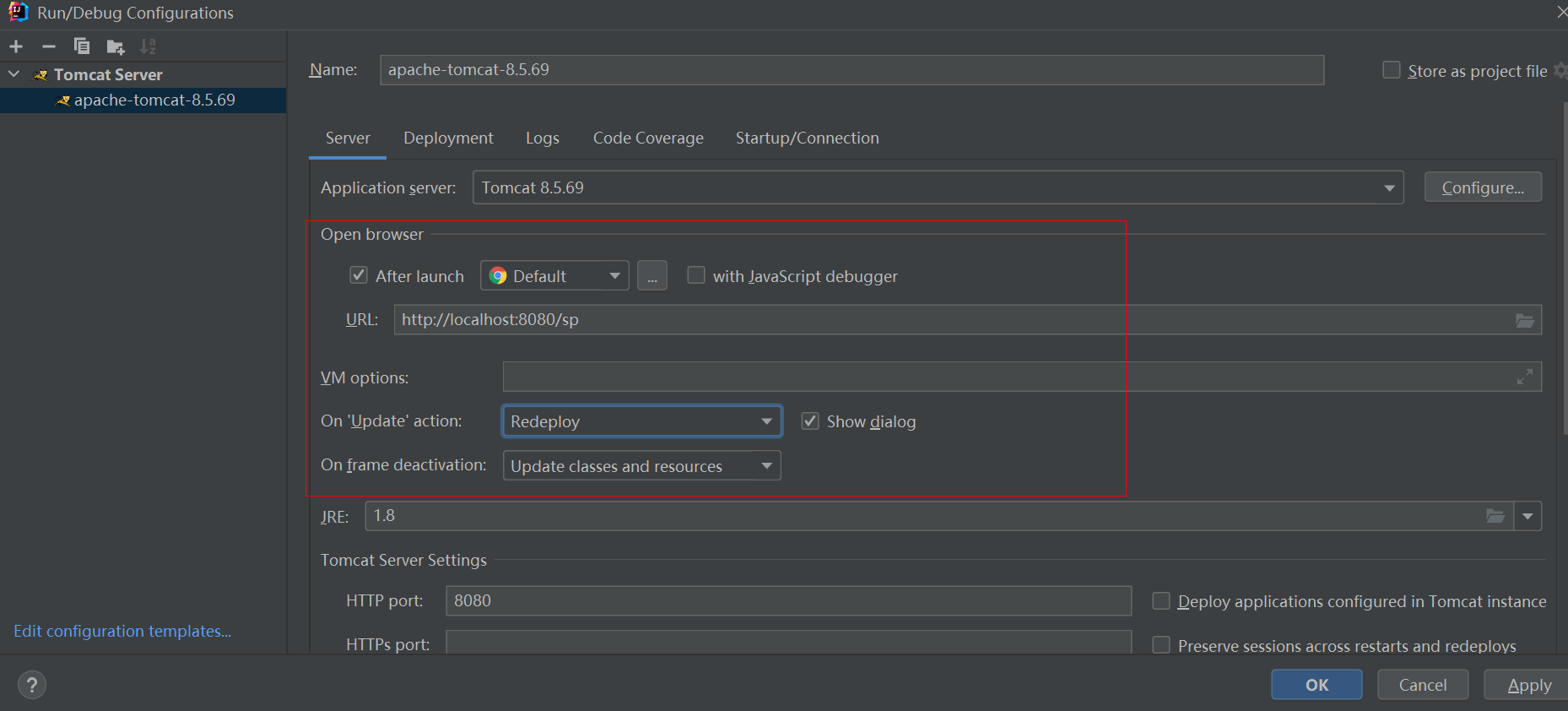
先点击Deployment选择包



往下翻设置Application Context的值Path

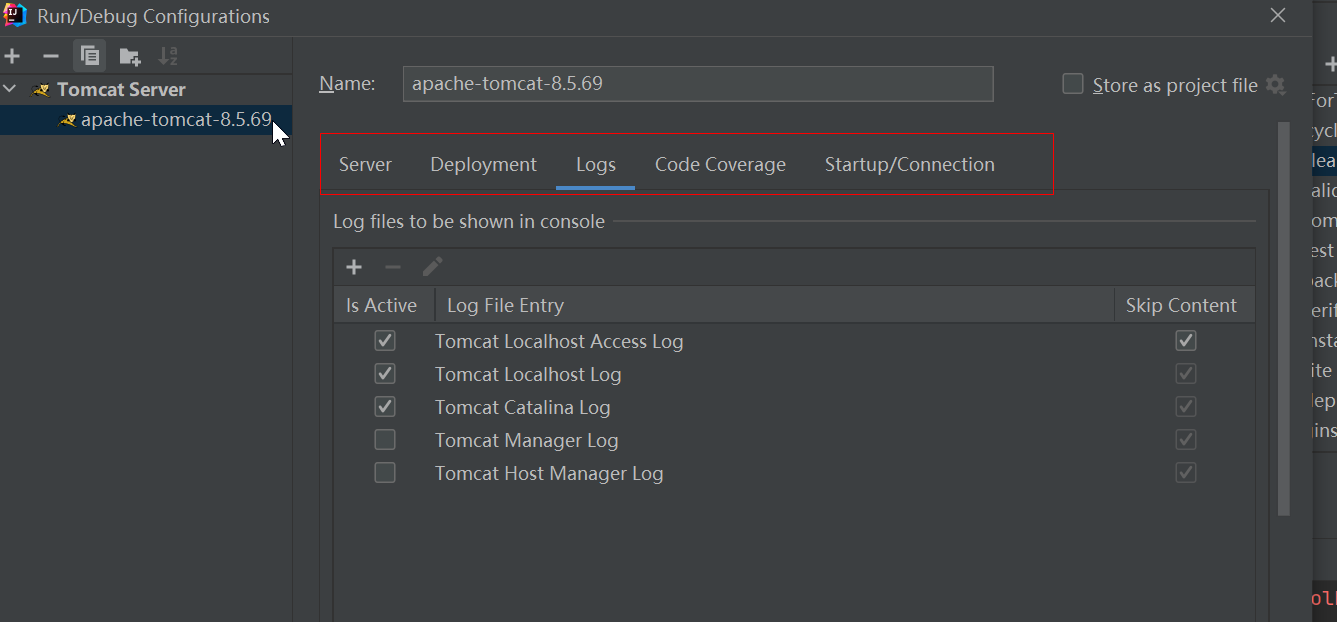


回到server标签，红框中的值需要设置了deploy后才能设置。其中浏览器打开的URL为Deploy标签中设置的Application Context的值Path。没有打包时的目录，这个时和直接在tomcat下部署的区别。

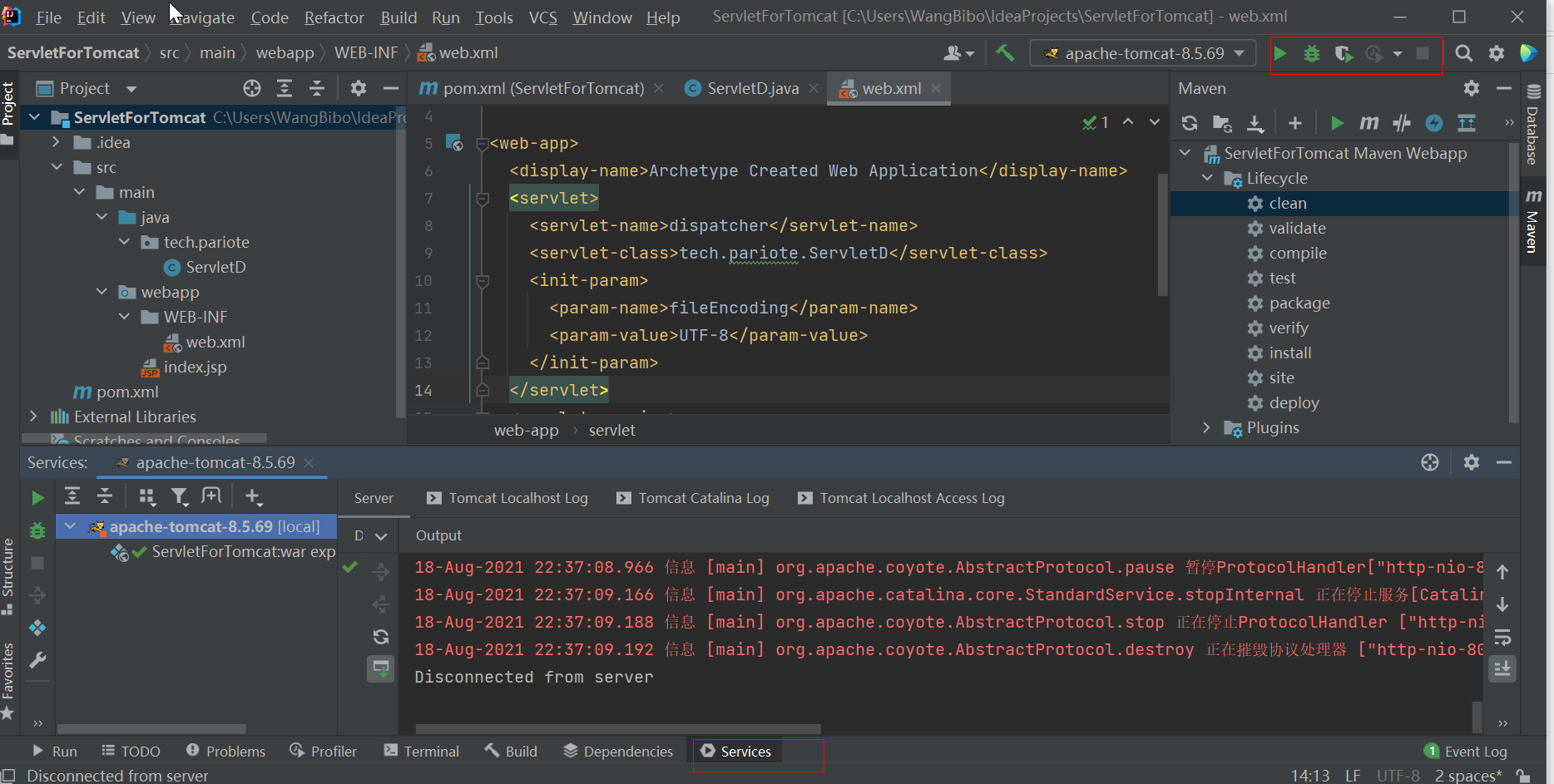


重新设置每次运行的行为，更新源代码

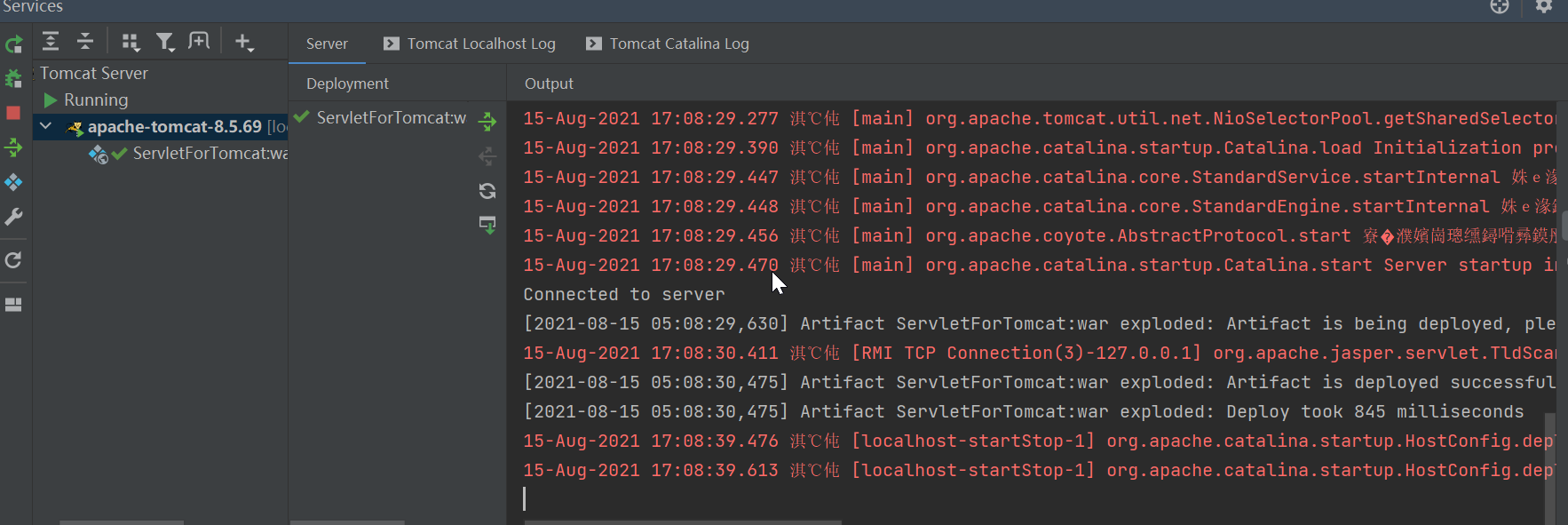
设置日志等



运行，调试，结束按钮及日志查看，可以打断点接收请求并调试。



问题：tomcat日志乱码



****修改logging.properties配置****

a.打开tomcat/conf/logging.properties

b.添加语句：

java.util.logging.ConsoleHandler.encoding = GBK

c.重启tomcat，查看日志数据即可