

用Intellij Idea调试本地Tomcat

群里有师傅问Tomcat怎么调试：



以前星球里也有人问过，<https://t.zsxq.com/MRBQF6M>，这里就写个详细的方法吧。

有的同学可能比较喜欢远程调试，我是比较喜欢本地调试的，毕竟电脑配置比较差，能不用虚拟机的时候就不用。本地调试，那么我们需要先将环境搭起来。

0x01 下载Tomcat

要调试Tomcat，当然要先下载Tomcat。首先说明一点，就是本地调试Tomcat并不意味着一定要本地编译Tomcat和main函数开始调起。所以，我们要下载的是Tomcat编译好的压缩包。

我这里以Windows系统下Tomcat 8为例，打开<https://tomcat.apache.org/download-80.cgi>，下载Core中的zip压缩包和Deployer中的zip压缩包：

Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

- Core:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
 - [32-bit Windows zip \(pgp, sha512\)](#)
 - [64-bit Windows zip \(pgp, sha512\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)
- Full documentation:
 - [tar.gz \(pgp, sha512\)](#)
- Deployer:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
- Extras:
 - [JMX Remote jar \(pgp, sha512\)](#)
 - [Web services jar \(pgp, sha512\)](#)
- Embedded:
 - [tar.gz \(pgp, sha512\)](#)
 - [zip \(pgp, sha512\)](#)

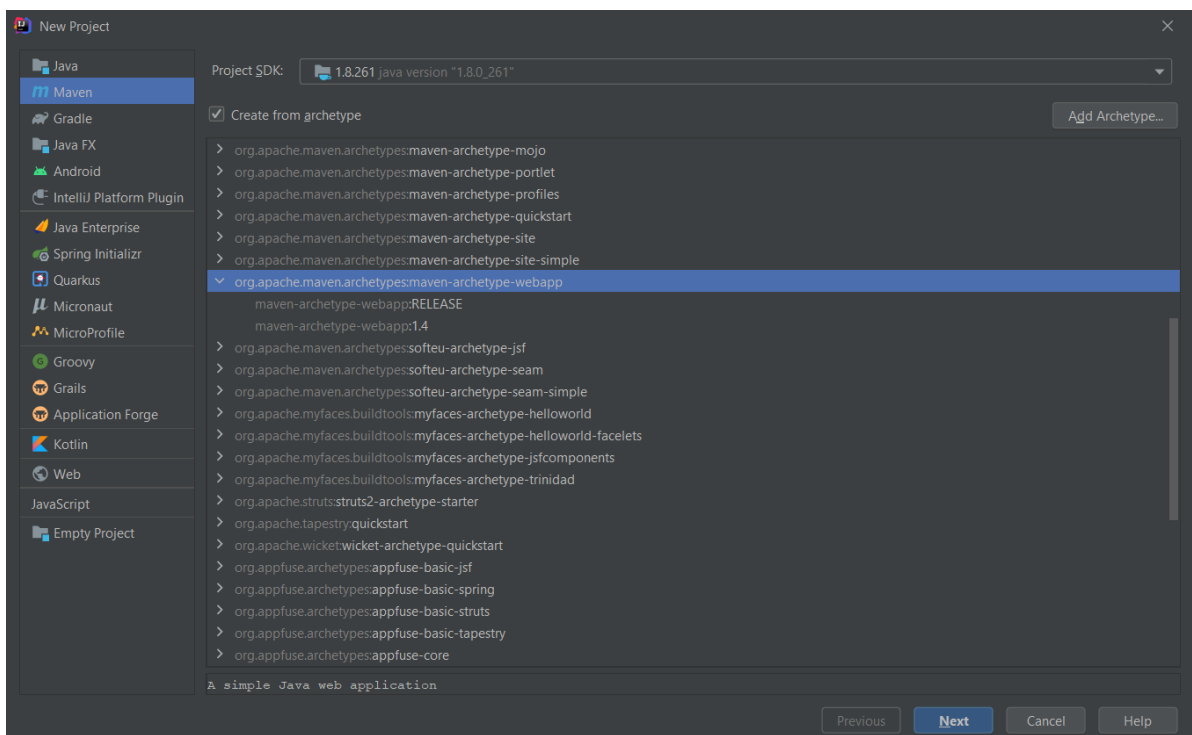
Source Code Distributions

- [tar.gz \(pgp, sha512\)](#)
- [zip \(pgp, sha512\)](#)

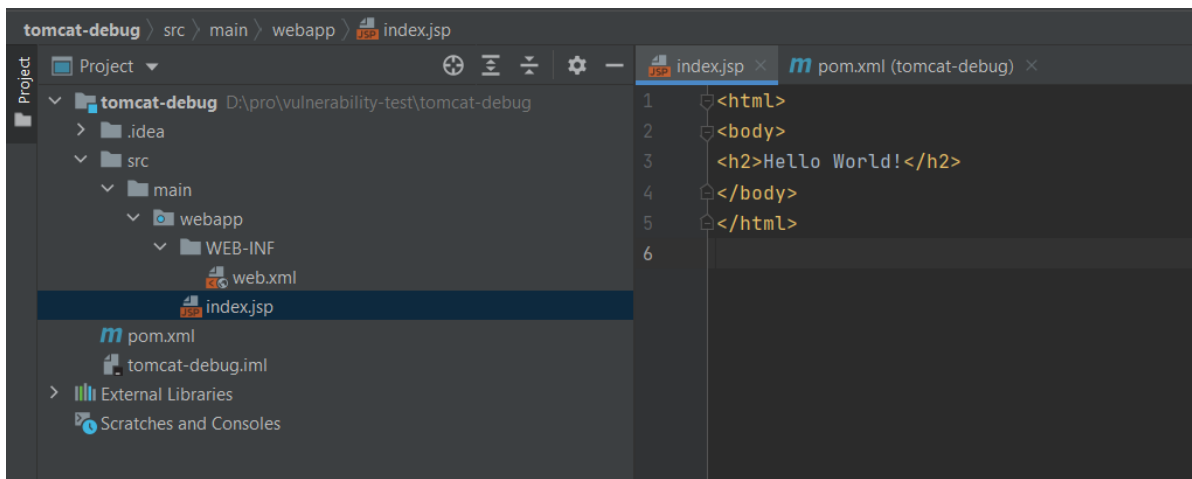
完了以后解压即可。

0x02 创建一个Java Web环境

在IDEA里创建一个Web项目。选择创建Maven项目，然后Maven提供了一些模板（archetype），我们选择maven-archetype-webapp：



创建完项目。此时生成的文件大概是这样：



我们简单地在index.jsp里面写行动态的代码吧：



你会发现IDEA会报错，找不到这个 `println`。原因是，在正常Tomcat中，是会附带servlet和jsp相关的依赖的，但是我们项目里没有这两个依赖，所以找不到。

此时有两种解决方案，一是，修改Maven的配置文件，加入依赖：

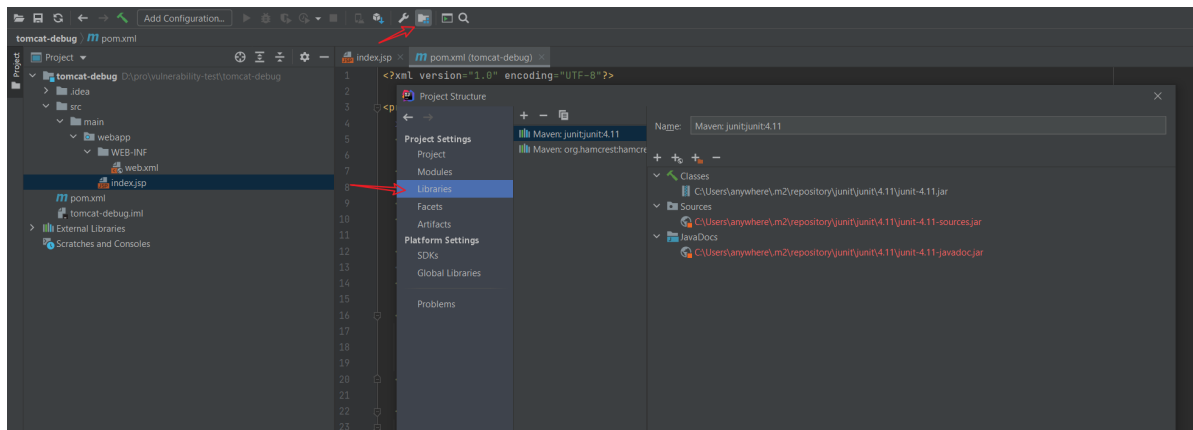
```
1 <dependency>
2   <groupId>javax.servlet</groupId>
3   <artifactId>javax.servlet-api</artifactId>
4   <version>3.1.0</version>
5   <scope>provided</scope>
6 </dependency>
7
8 <dependency>
9   <groupId>javax.servlet.jsp</groupId>
10  <artifactId>jsp-api</artifactId>
11  <version>2.2</version>
12  <scope>provided</scope>
13 </dependency>
```

注意，要设置scope为provided，这个意思就是，在编译打包的时候不会将这两个依赖的代码打进包中，因为Tomcat中已经带了这两个依赖。

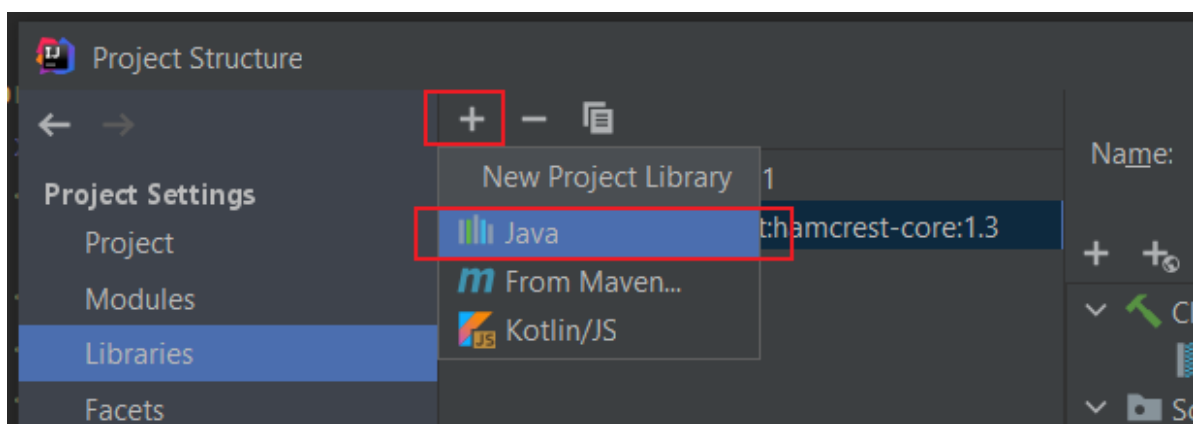
0x03 运行Tomcat环境

不过本文要讲的是方法二，方法二是将我们前面下载的Tomcat里所有的依赖直接添加到我们项目中。

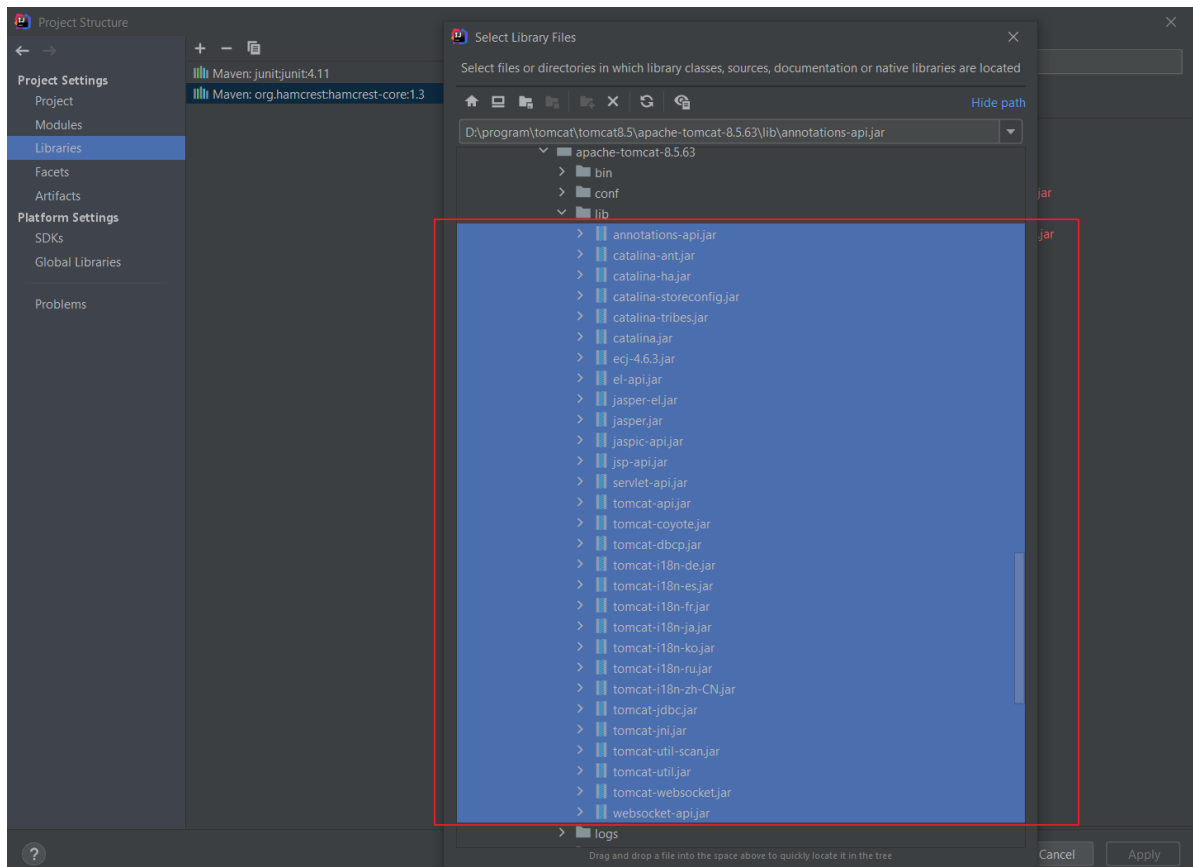
打开**Project Structure**窗口，进入**Libraries**页面，如图：



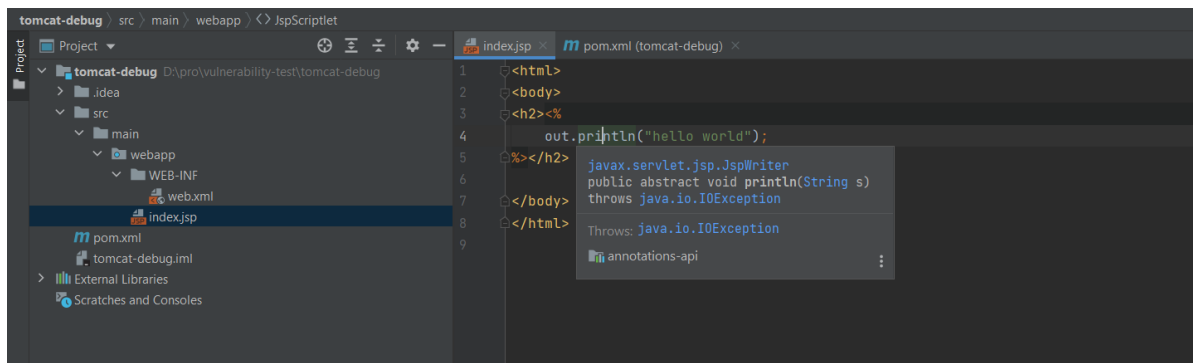
这个就是依赖的管理页面，此时能看到Maven默认安装的依赖junit。然后我们点击这里的加号（+），选择其中的**Java**：



然后找到前面下载的Tomcat文件夹下的lib文件夹，选择里面所有的jar包，点击确定：

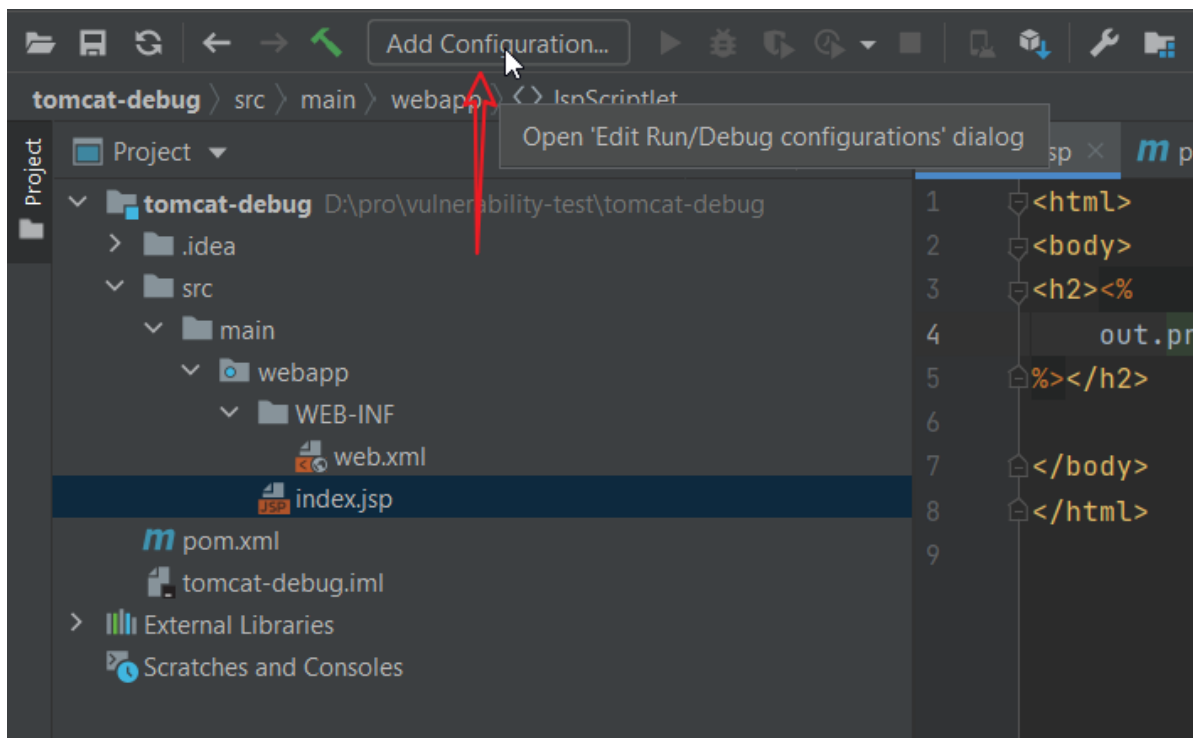


此时，这些jar包就全部被加入当前项目作为依赖了。再返回看index.jsp，就不报错了：

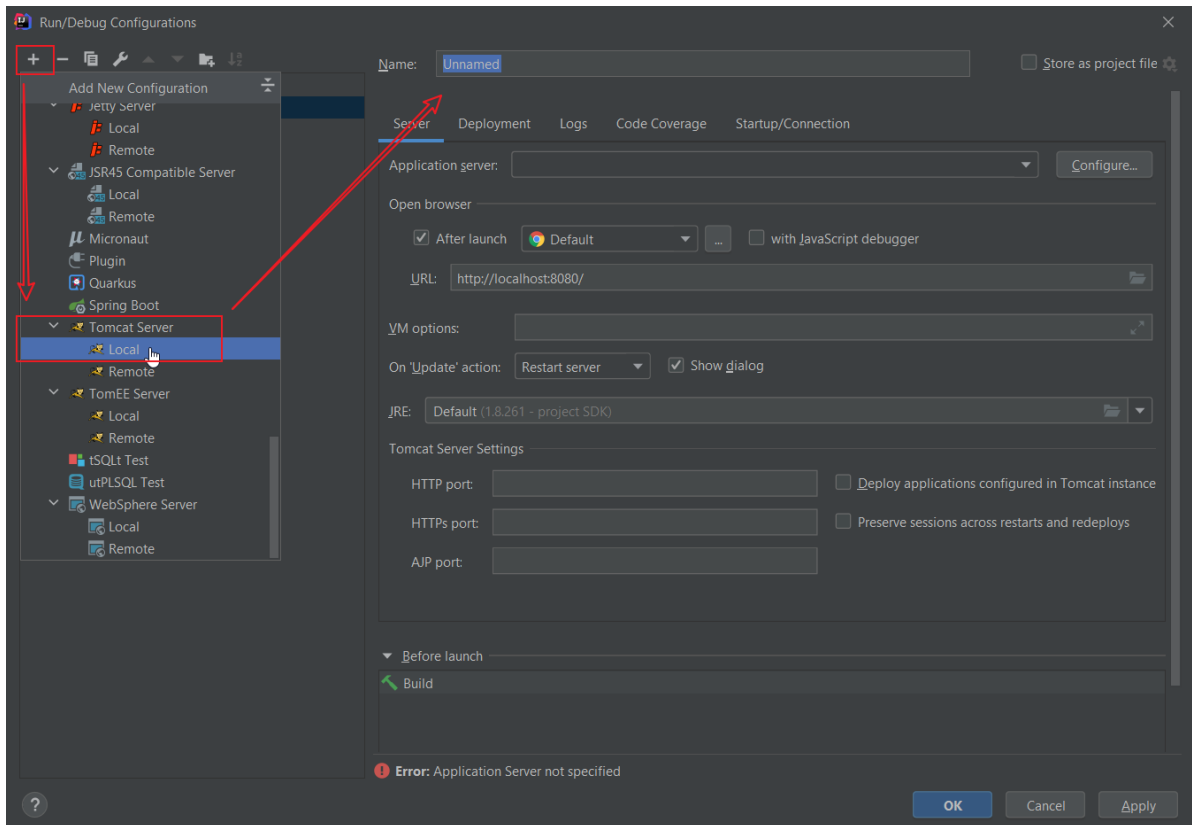


我为什么要用方法二呢？明明改Maven配置文件更简单。但是因为我们之后要调试到Tomcat，这些jar包就是我们要调试的内容。

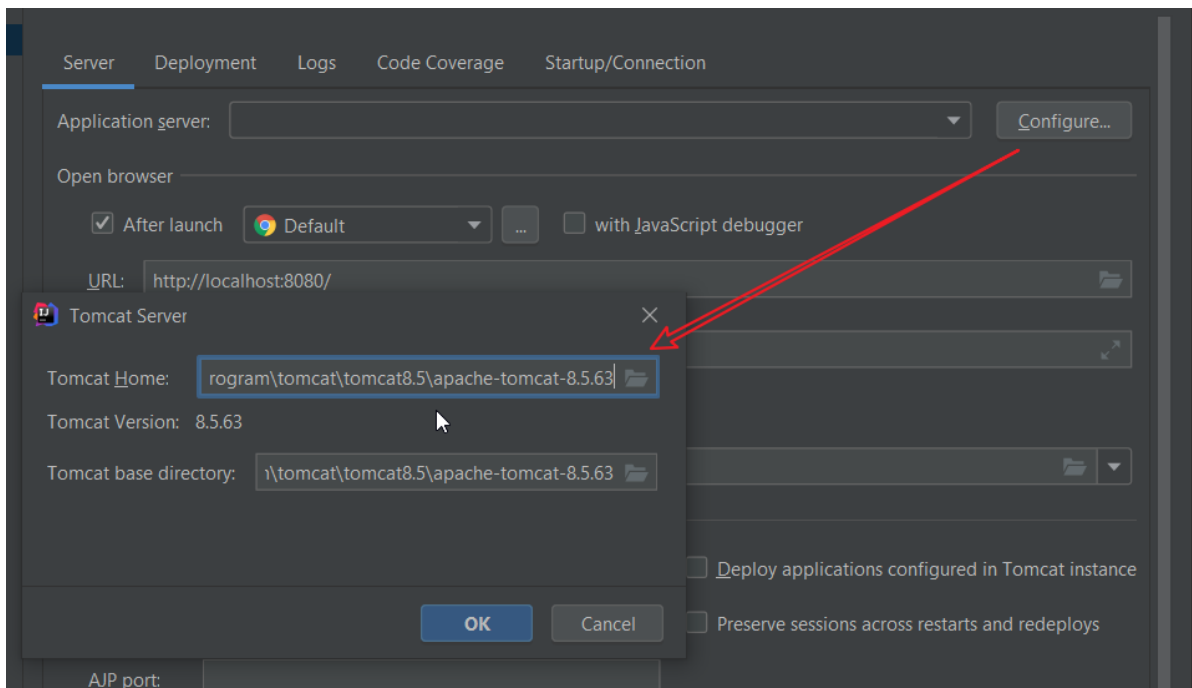
然后，我们添加一个Run/Debug配置：



添加一个本地的Tomcat配置：



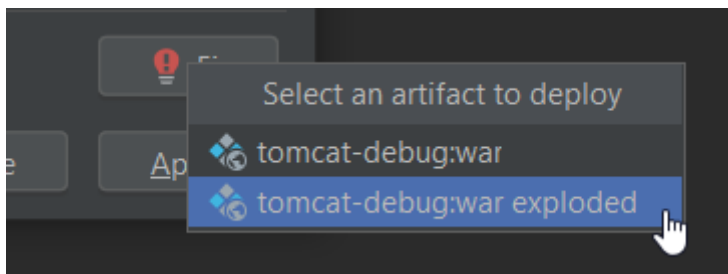
此时没有Application server，我们添加一个，点击**Application server**右边的**Configure**，选择Tomcat的目录，IDEA能自动识别出当前Tomcat的版本：



添加后，窗口下面还有个Warning，就是我们没有选择要部署的内容：

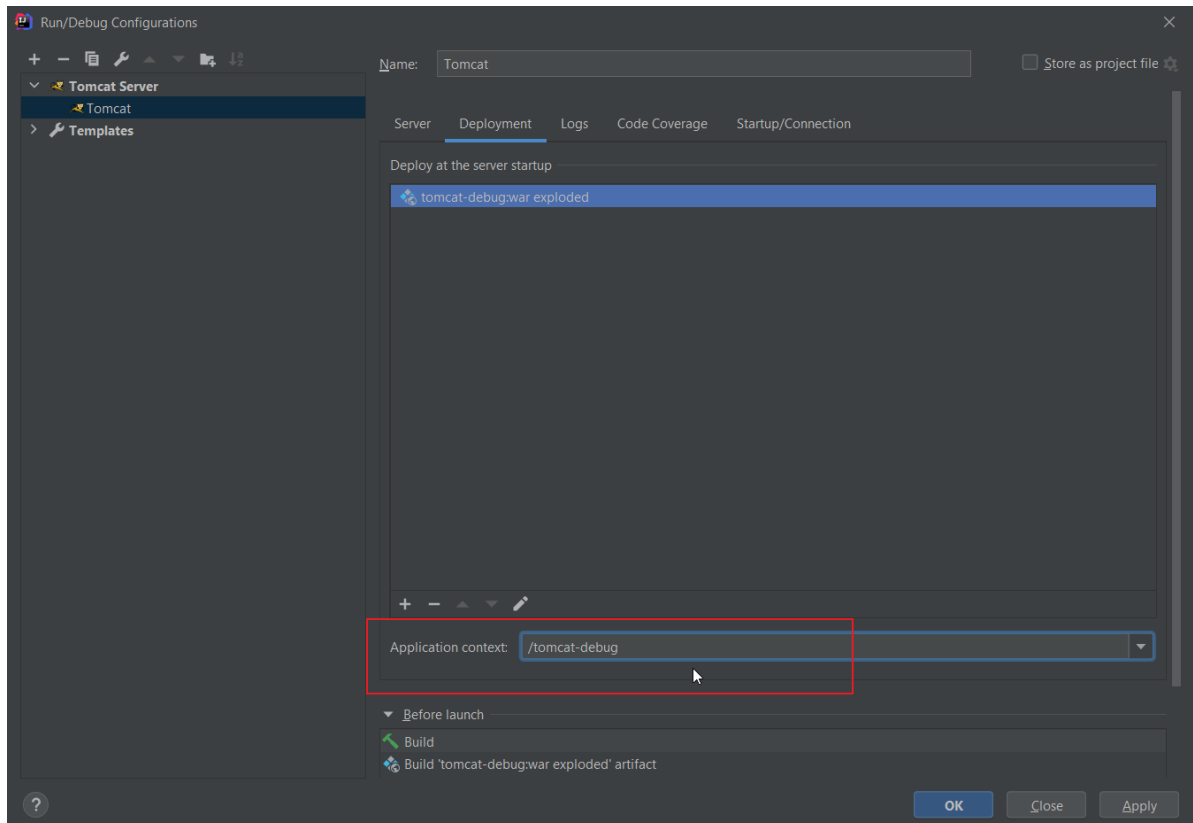


点击右侧的**Fix**，修复这个告警。点击Fix后会有两个选项，一个是war，一个是war exploded：

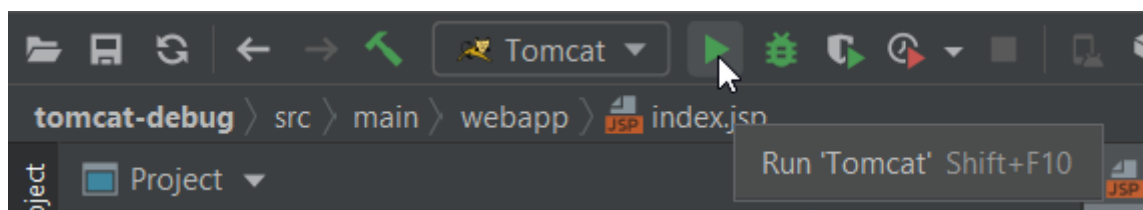


我们选择**war exploded**。其实前者是说将本地项目打包成war，部署到Tomcat中；后者是直接将war解压后的目录部署到Tomcat中，用哪个都可以。

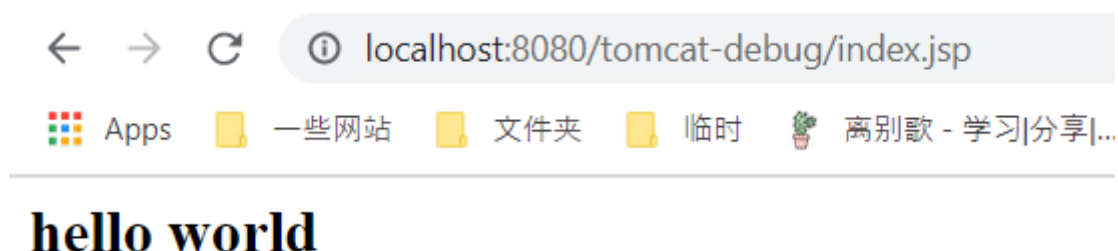
fix后会自动帮我们添加一个Deployment，其中有个**Application context**，就是这个Web应用的目录。我这里改成 `/tomcat-debug`，这样比较简单：



点击OK。然后配置好了，我们将其运行起来：

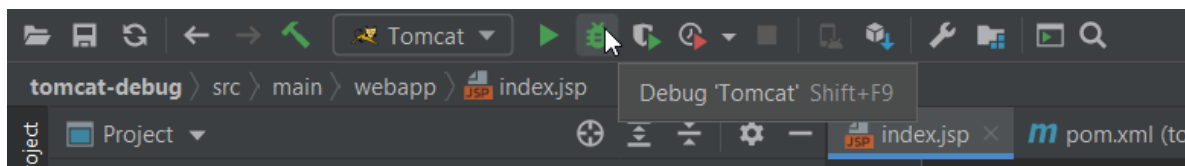


运行后，我们浏览器里访问 `http://localhost:8080/tomcat-debug/index.jsp`，即可看到页面成功输出了hello world。



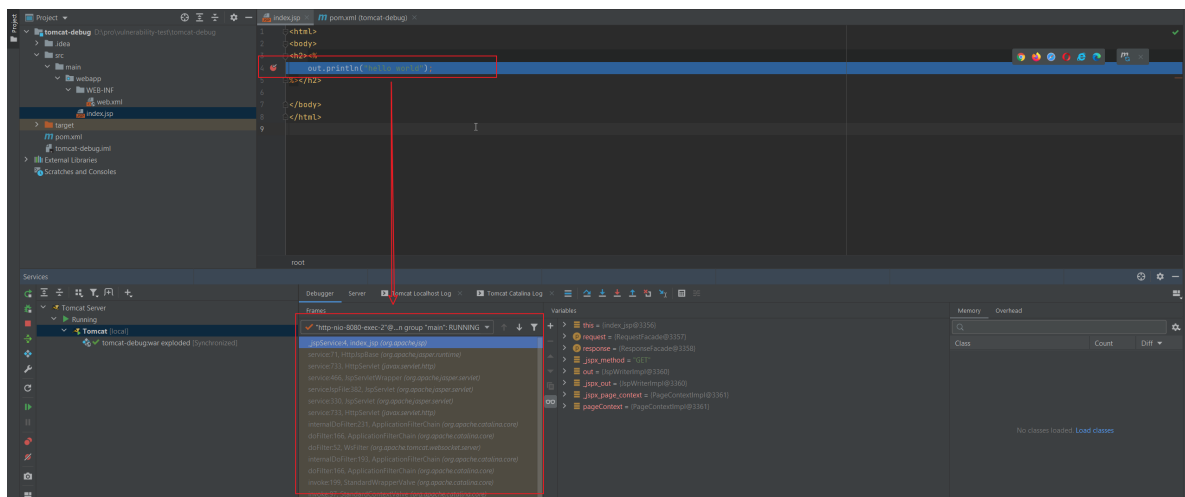
0x04 调试Tomcat环境

已经成功运行了，调试就简单了。点击调试按钮，将Tomcat以调试的模式运行起来：

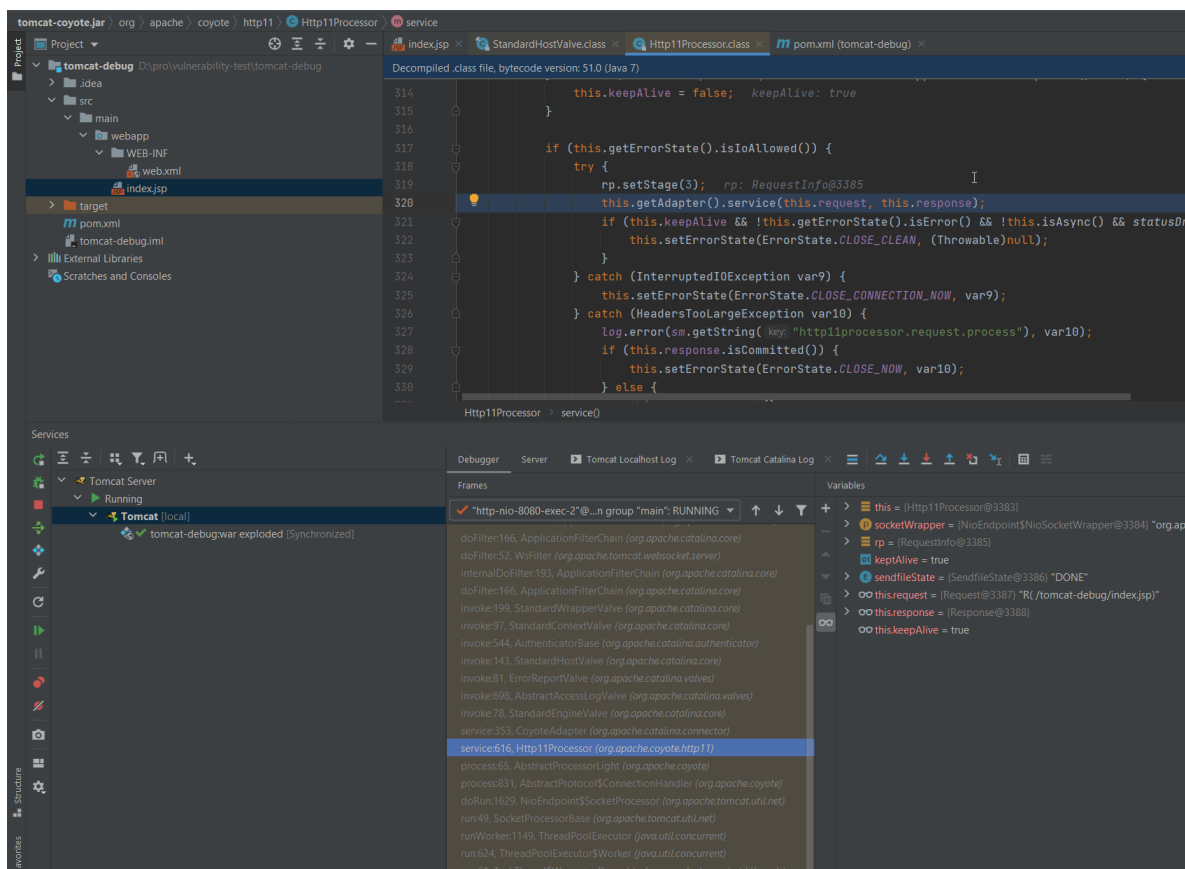


此时没有下任何断点，所以正常访问 `/tomcat-debug/index.jsp` 仍然可以看到正常页面。

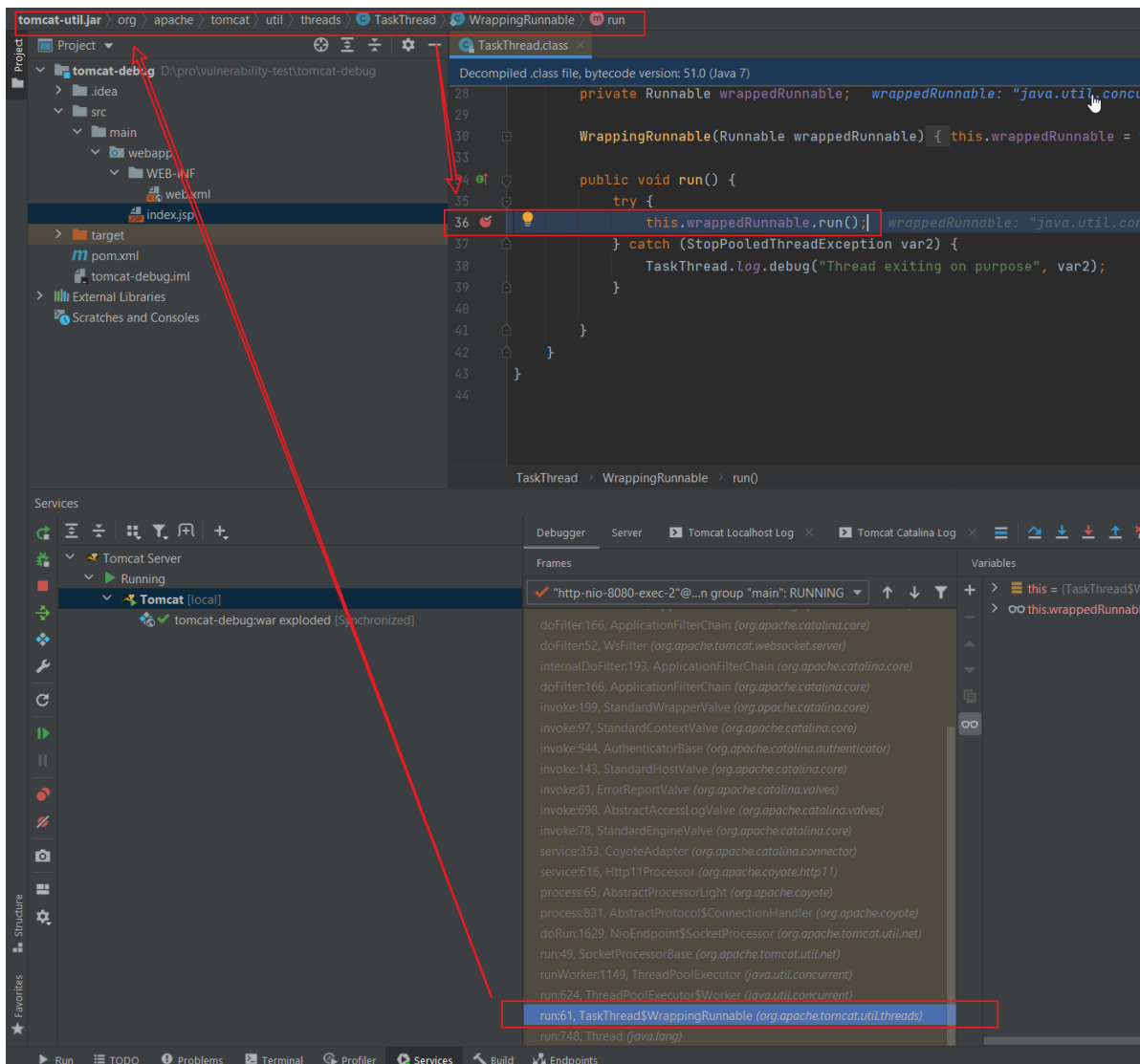
我们可以在index.jsp中下个断点，再刷新下浏览器，即可在此处断下来。此时，我们可以观察到当前的堆栈信息：



我们随便点击堆栈中的某一行，IDEA会帮我们打开并反编译对应的类，这个类 `org.apache.coyote.http11.Http11Processor` 就是Tomcat中的类了：



然后，我们找到堆栈中倒数第二行，这就是Tomcat的入口点了。我们在这里下个断点：



然后，我们以调试的方式重启Tomcat，就会先断在这个入口点处，然后按照正常的IDEA调试方法进行调试就行了。要调试其他的类也一样，先找到这个类的位置，然后下断点调试即可。

0x05 优化调试流程

理论上到这里我们就可以正常调试了，但是上述的过程其实有一个重要问题：**Tomcat是开源软件，我们可以拿到源码，但调试的时候看到的却是反编译的代码，遗失了不少信息，怎么办？**

也很简单，我们回到Tomcat官网，下载**对应版本的源码压缩包**并解压：

8.5.64

Please see the [README](#) file for packaging information. It explains what every distribution contains.

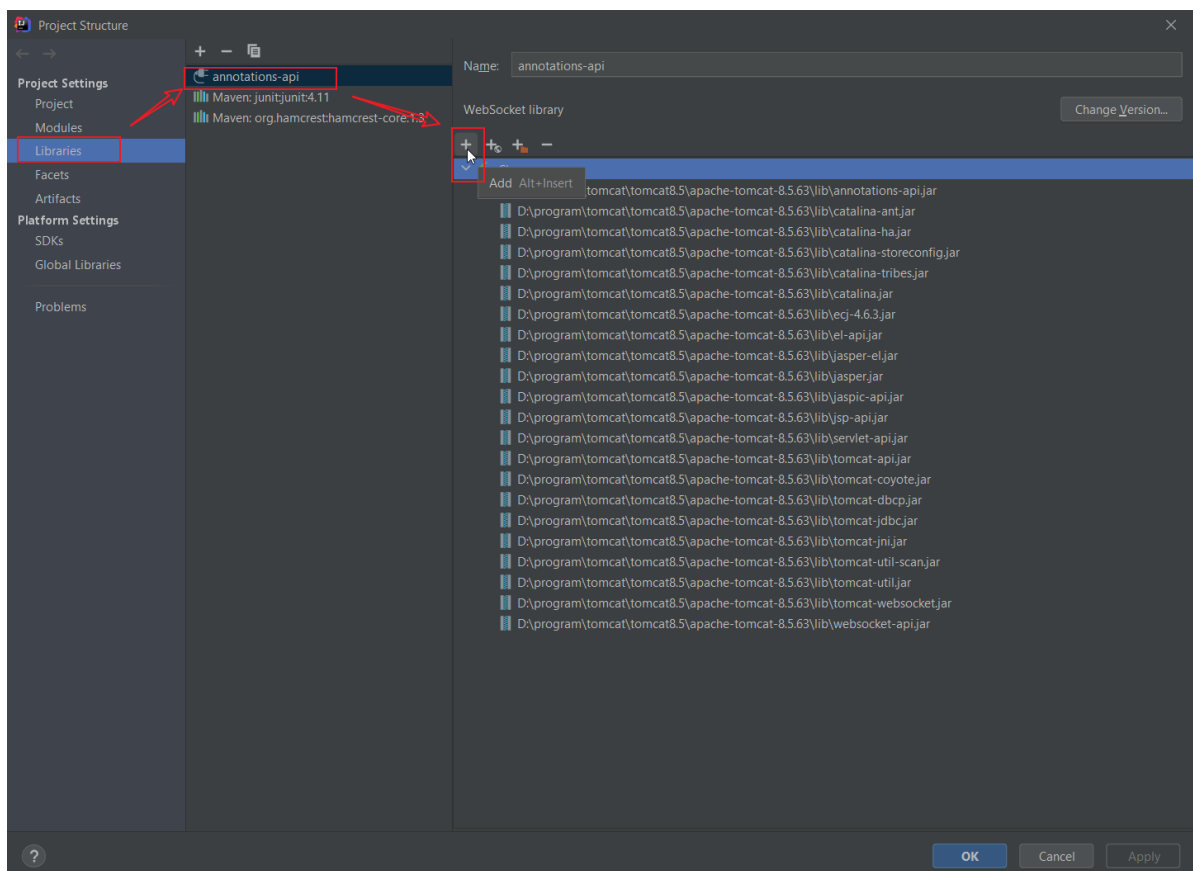
Binary Distributions

- Core:
 - [zip](#) (pgp, sha512)
 - [tar.gz](#) (pgp, sha512)
 - [32-bit Windows zip](#) (pgp, sha512)
 - [64-bit Windows zip](#) (pgp, sha512)
 - [32-bit/64-bit Windows Service Installer](#) (pgp, sha512)
- Full documentation:
 - [tar.gz](#) (pgp, sha512)
- Deployer:
 - [zip](#) (pgp, sha512)
 - [tar.gz](#) (pgp, sha512)
- Extras:
 - [JMX Remote jar](#) (pgp, sha512)
 - [Web services jar](#) (pgp, sha512)
- Embedded:
 - [tar.gz](#) (pgp, sha512)
 - [zip](#) (pgp, sha512)

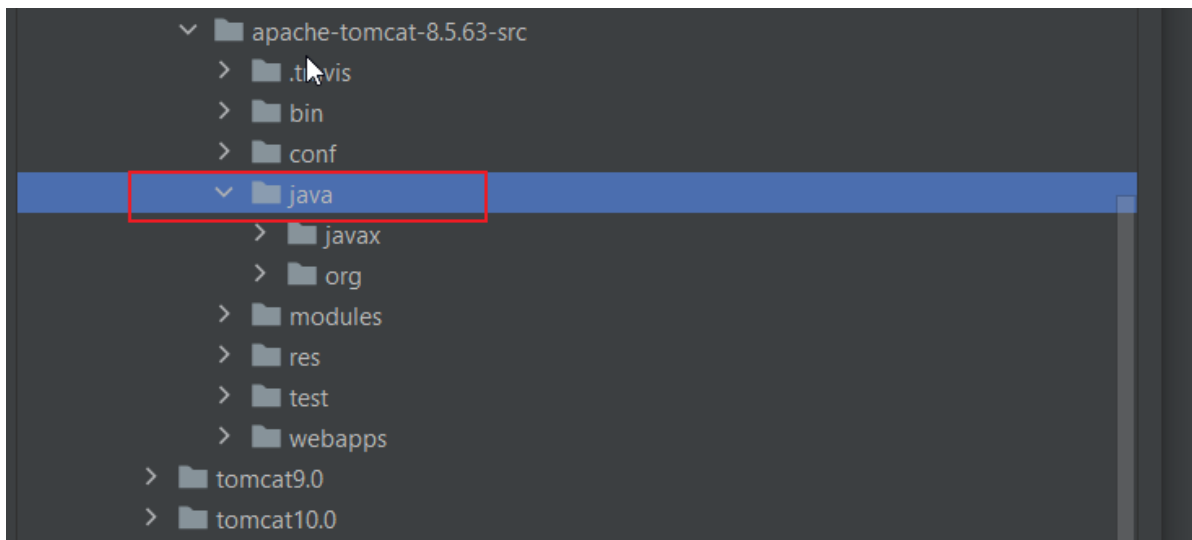
Source Code Distributions

- [tar.gz](#) (pgp, sha512)
- [zip](#) (pgp, sha512)

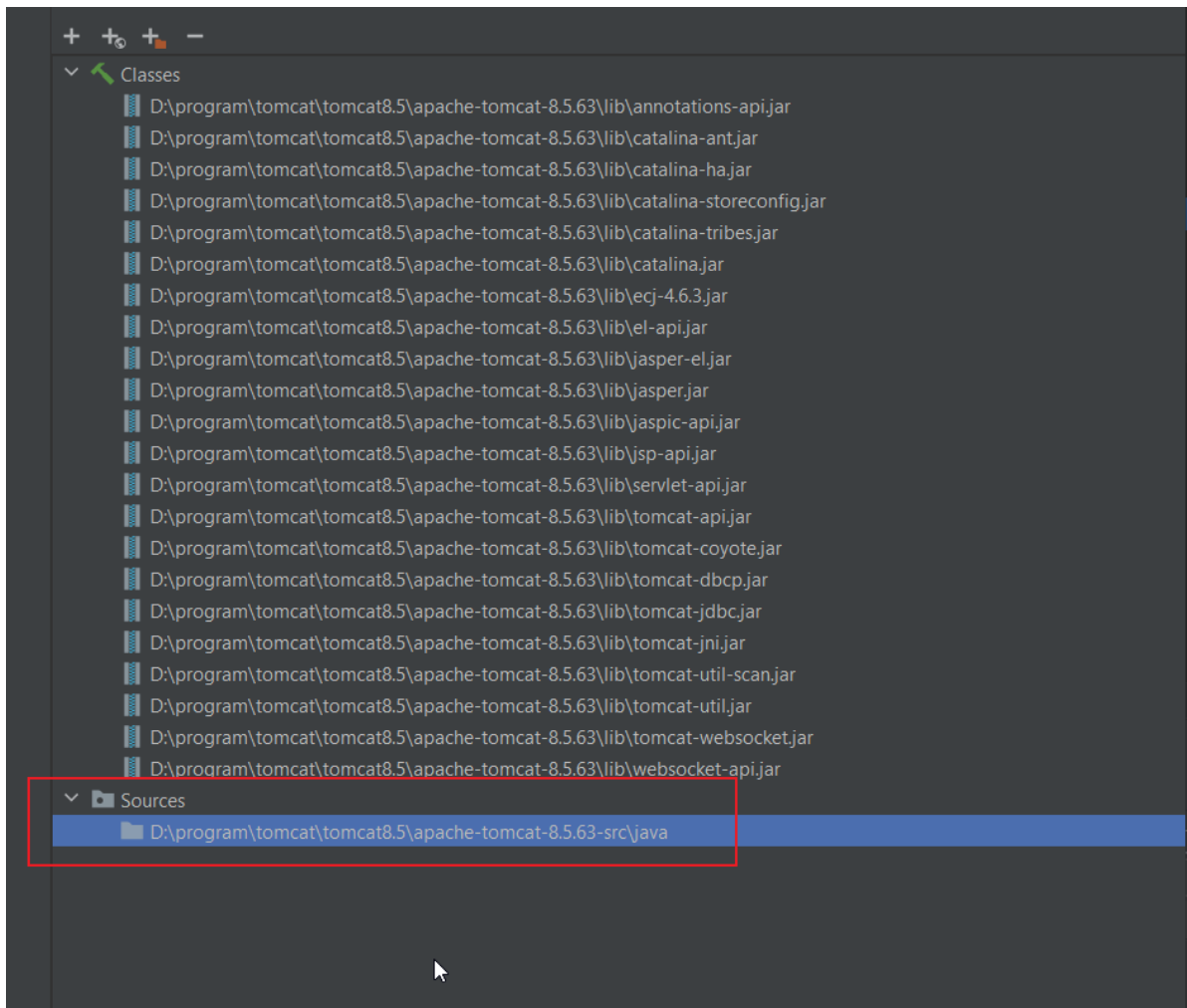
回到IDEA，仍然来到**Project Structure**的**Libraries**页面，找到刚才添加的依赖，点击依赖这一侧的加号：



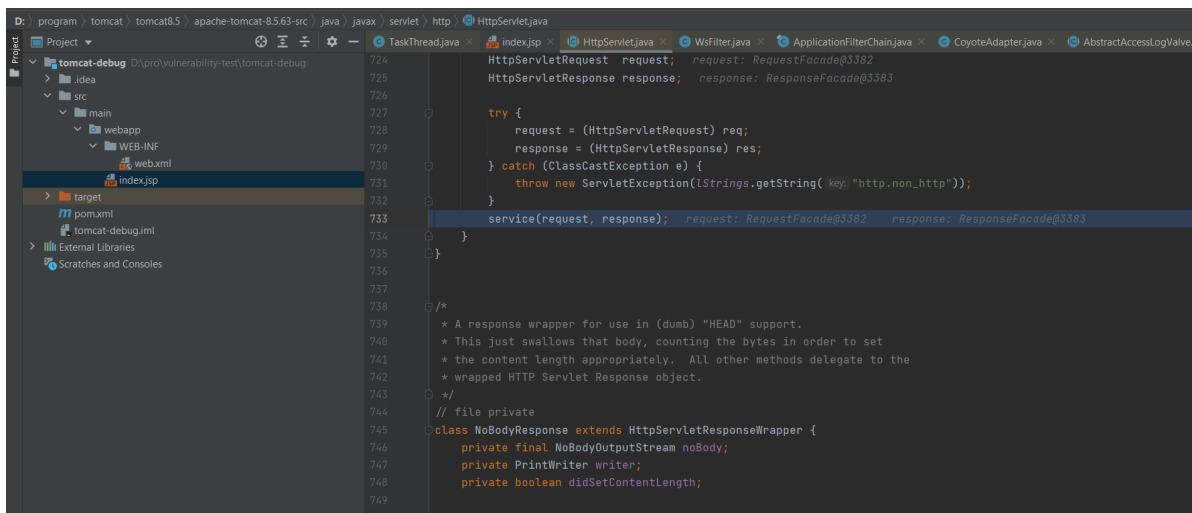
选择刚才下载的源码包中的java目录，添加：



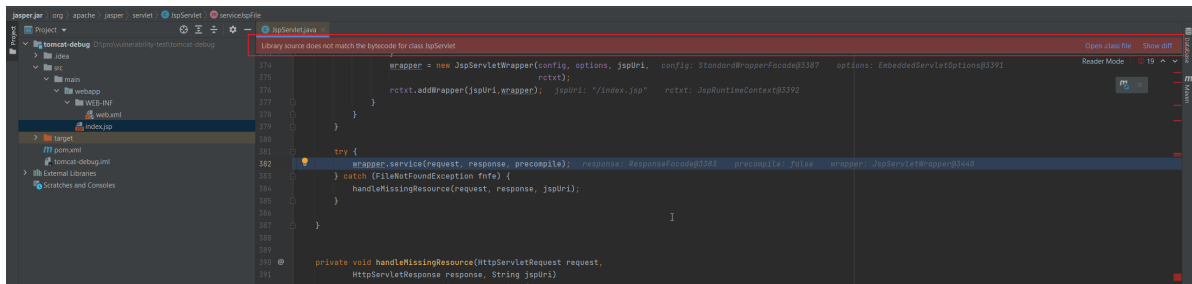
此时你会发现多了个Sources的项目：



此时，再调试看到的就是有注释的源码页面了：

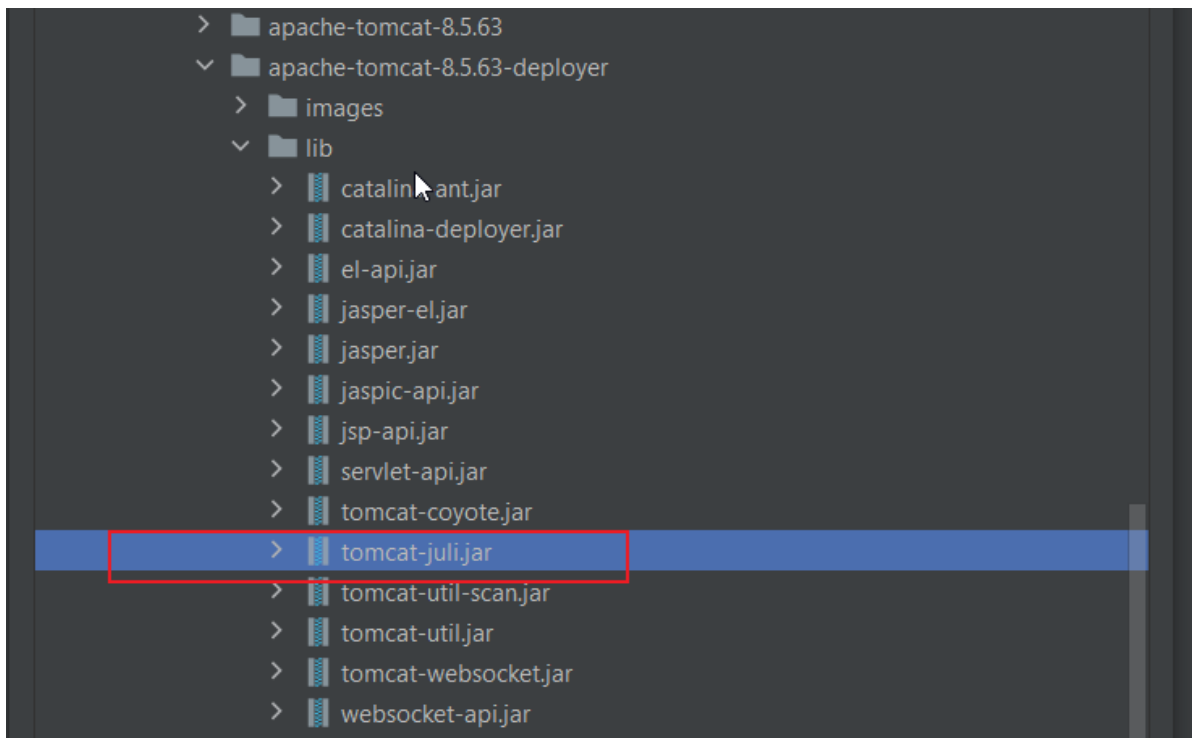


但是，调试的过程中，我们会发现另一个问题。有些页面能够正常看到源码，但是有的页面会出现这样一个错误，说是源码和反编译的字节码不能对应上：

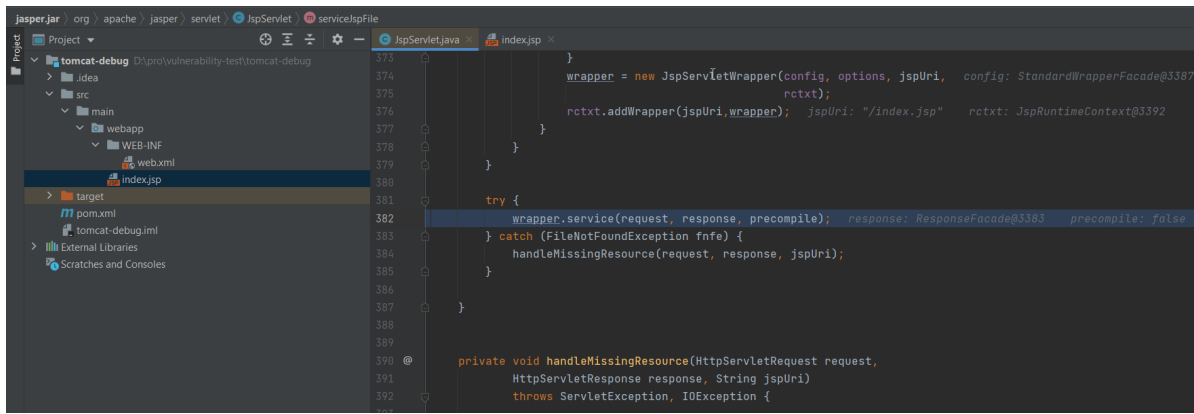


难道是源码下载错了吗？

实际上是我们少加了一个依赖。还记得第一章时我们下载了两个压缩包，一个**Core**，一个**Deployer**，我们按照添加依赖的步骤，将**Deployer**中的**tomcat-juli.jar**也添加到依赖中：



之后报错就消失了，解决问题：



0x06 总结

总结一下。首先，调试Tomcat的方法很多，我这个只是其中之一。我这个方法的优点就是本地调试，而且不需要编译Tomcat，环境快速搭建，十分方便。

其次，熟练掌握IDEA内的各个功能，自行搭配一下，大部分问题就迎刃而解了。