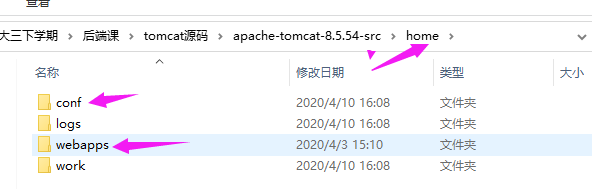
Tomcat源码

# 1 前期步骤

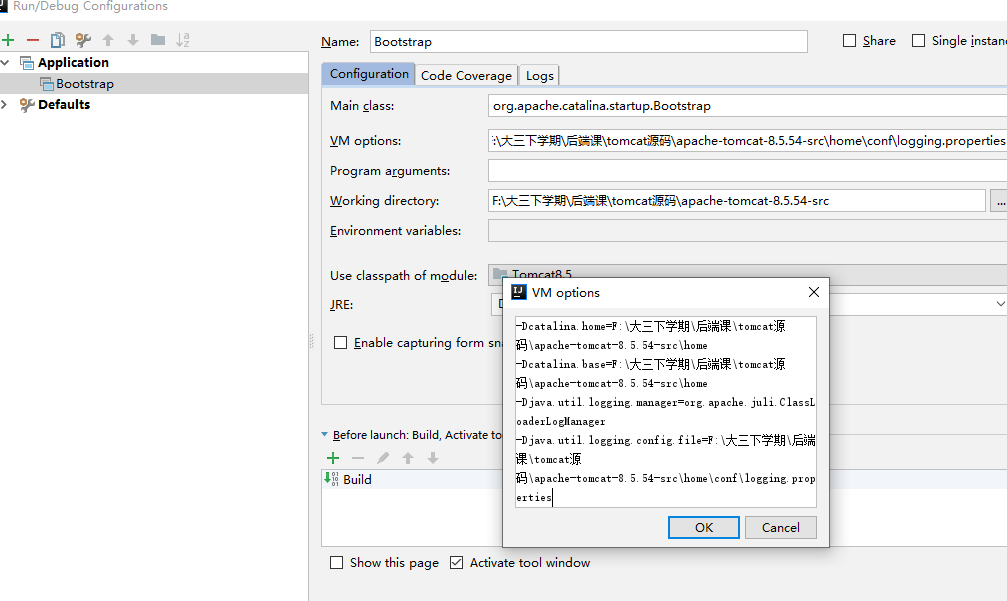
先下载tomcat源码（<https://tomcat.apache.org/download-80.cgi>），在tomcat的目录中创建一个home目录，然后再把conf目录和webapp目录移动到hone目录中，再创建一个pom.xml文件，

pom.xml内容如下：

*<?***xml version="1.0" encoding="utf-8"***?>*<**project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"**>  
 <**modelVersion**>4.0.0</**modelVersion**>  
 <**groupId**>org.apache.tomcat</**groupId**>  
 <**artifactId**>Tomcat8.5</**artifactId**>  
 <**name**>Tomcat8.5</**name**>  
 <**version**>8.5</**version**>  
 <**build**>  
 <**finalName**>Tomcat8.0</**finalName**>  
 ***<!-- 指定源文件为java 、test -->*** <**sourceDirectory**>java</**sourceDirectory**>  
 <**testSourceDirectory**>test</**testSourceDirectory**>  
 <**resources**>  
 <**resource**>  
 <**directory**>java</**directory**>  
 </**resource**>  
 </**resources**>  
 <**testResources**>  
 <**testResource**>  
 <**directory**>test</**directory**>  
 </**testResource**>  
 </**testResources**>  
 <**plugins**>  
 <**plugin**>  
 <**groupId**>org.apache.maven.plugins</**groupId**>  
 <**artifactId**>maven-compiler-plugin</**artifactId**>  
 <**version**>2.3</**version**>  
 <**configuration**>  
 <**encoding**>UTF-8</**encoding**>  
 ***<!-- 指定jdk 编译 版本 ，没装jdk 1.7的可以变更为1.6 -->*** <**source**>1.8</**source**>  
 <**target**>1.8</**target**>  
 </**configuration**>  
 </**plugin**>  
 </**plugins**>  
 </**build**>  
 ***<!-- 添加tomcat8 所需jar包依赖 -->*** <**dependencies**>  
 <**dependency**>  
 <**groupId**>junit</**groupId**>  
 <**artifactId**>junit</**artifactId**>  
 <**version**>4.12</**version**>  
 <**scope**>test</**scope**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>ant</**groupId**>  
 <**artifactId**>ant</**artifactId**>  
 <**version**>1.7.0</**version**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>wsdl4j</**groupId**>  
 <**artifactId**>wsdl4j</**artifactId**>  
 <**version**>1.6.2</**version**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>javax.xml</**groupId**>  
 <**artifactId**>jaxrpc</**artifactId**>  
 <**version**>1.1</**version**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>org.easymock</**groupId**>  
 <**artifactId**>easymock</**artifactId**>  
 <**version**>3.3</**version**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>org.eclipse.jdt.core.compiler</**groupId**>  
 <**artifactId**>ecj</**artifactId**>  
 <**version**>4.6.1</**version**>  
 </**dependency**>  
 </**dependencies**>  
</**project**>

# 2 配置VM

新建一个Application，名字叫BootStrap

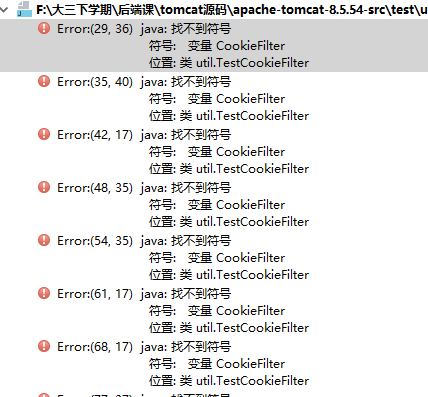


1. 选择Main class：选择Bootstrap所在的目录
2. 添加VM options：

-Dcatalina.home=F:\大三下学期\后端课\tomcat源码\apache-tomcat-8.5.54-src\home

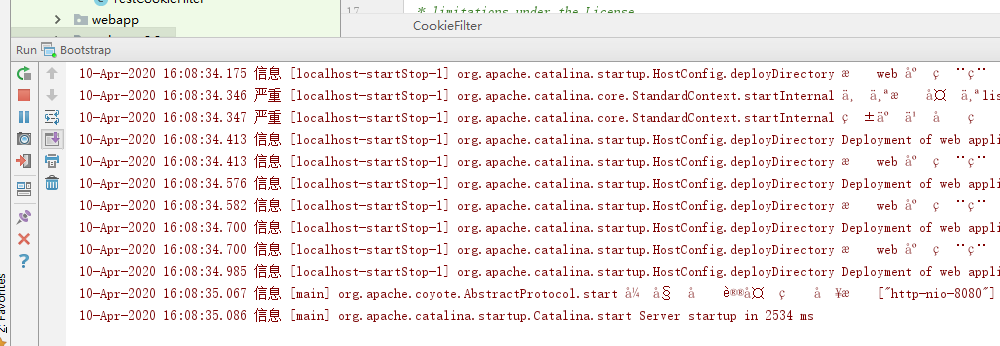
-Dcatalina.base=F:\大三下学期\后端课\tomcat源码\apache-tomcat-8.5.54-src\home -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.util.logging.config.file=F:\大三下学期\后端课\tomcat源码\apache-tomcat-8.5.54-src\home\conf\logging.properties

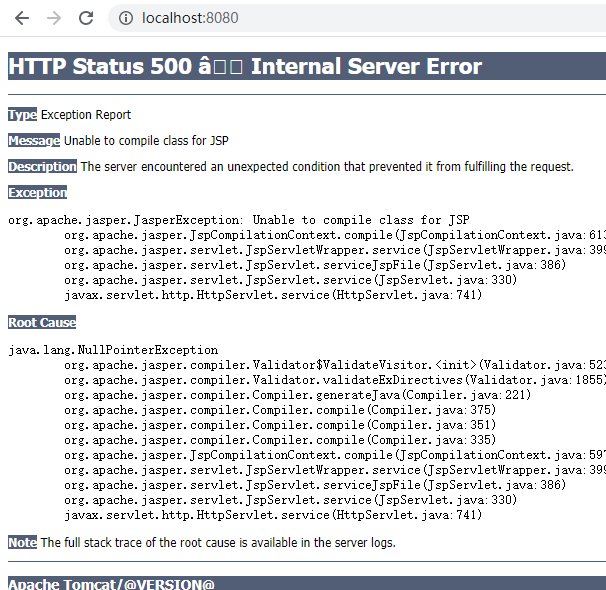
# 3 启动报错



解决方法：在网上自己找一个CookieFilter类（放在test\util下）

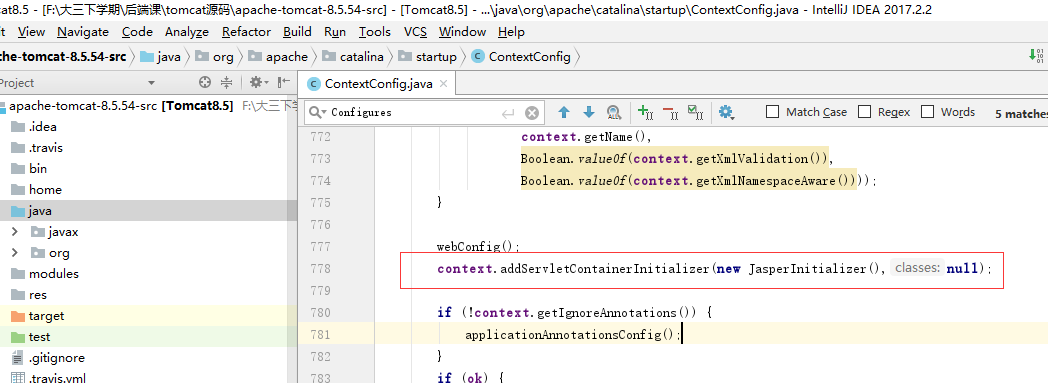
# 4 启动成功（访问不成功）



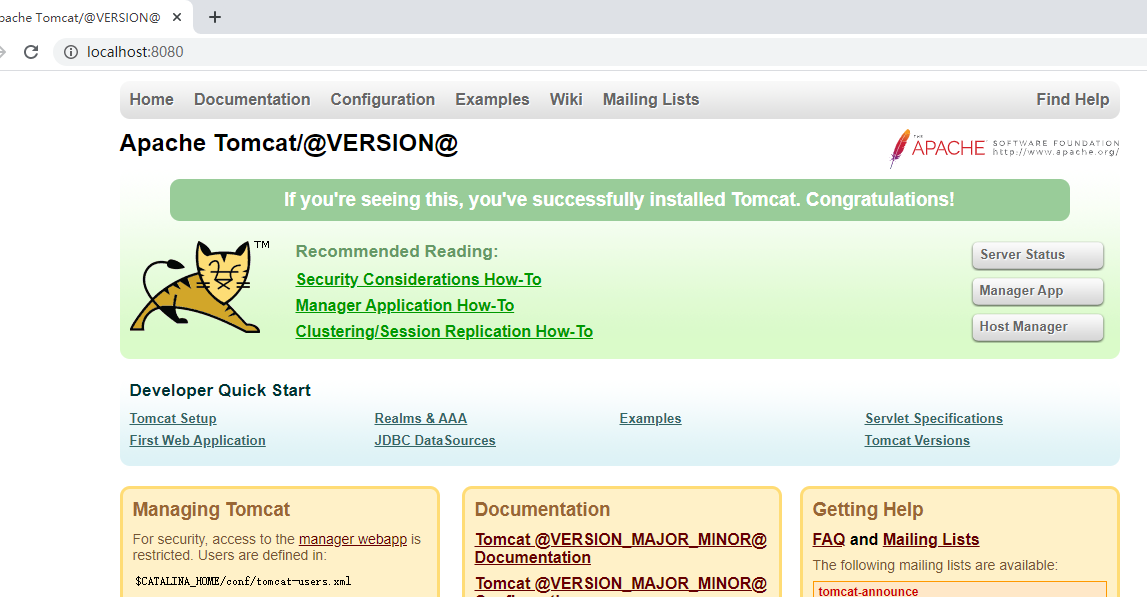


访问不成功的原因，缺少对jsp的处理，解决方法：在ContextConfig类中的ConfigureStart方法中加上一行代码：

**context**.addServletContainerInitializer(**new** JasperInitializer(),**null**);



# 5 运行成功



# 6 源码理解

Tomcat其实就是分析一个Servlet容器，从 servlet容器的角度来看看 Tomcat。一个功能健全的 servlet容器对于每个 servlet 的HTTP请求会完成以下事情：

（1）当servlet 第一次被调用的时候，加载了 servlet类并调用它的init方法（仅调用一次）。

（2）响应每次请求的时候 ，构建一 个javax.servlet.ServletRequest 和 javax.servlet.ServletResponse实例。

（3）调用servlet的service方法，将ServletRequest对象 和 ServletResponse 对象当作参数传入。

（4）当servlet类关闭的时候，调用servlet的destroy方法，并卸载servlet类。