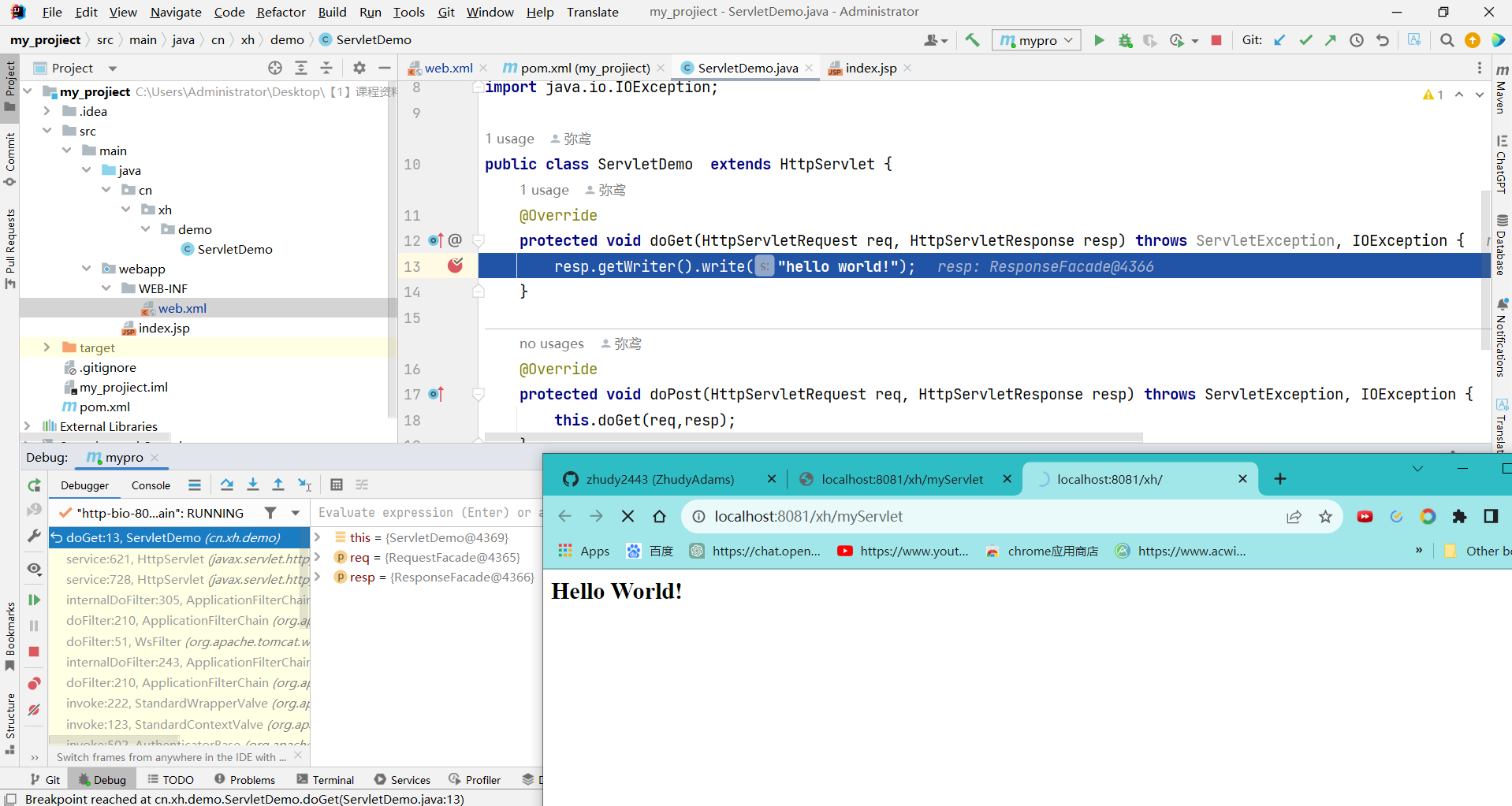
Maven配置插件的方式

<**build**>  
 <**finalName**>idea-maven</**finalName**>  
 <**pluginManagement**>*<!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->* <**plugins**>  
 <**plugin**>  
 <**artifactId**>maven-clean-plugin</**artifactId**>  
 <**version**>3.1.0</**version**>  
 </**plugin**>  
 *<!-- see http://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_war\_packaging -->* <**plugin**>  
 <**artifactId**>maven-resources-plugin</**artifactId**>  
 <**version**>3.0.2</**version**>  
 </**plugin**>  
 <**plugin**>  
 <**artifactId**>maven-compiler-plugin</**artifactId**>  
 <**version**>3.8.0</**version**>  
 </**plugin**>  
 <**plugin**>  
 <**artifactId**>maven-surefire-plugin</**artifactId**>  
 <**version**>2.22.1</**version**>  
 </**plugin**>  
 <**plugin**>  
 <**artifactId**>maven-war-plugin</**artifactId**>  
 <**version**>3.2.2</**version**>  
 </**plugin**>  
 <**plugin**>  
 <**artifactId**>maven-install-plugin</**artifactId**>  
 <**version**>2.5.2</**version**>  
 </**plugin**>  
 <**plugin**>  
 <**artifactId**>maven-deploy-plugin</**artifactId**>  
 <**version**>2.8.2</**version**>  
 </**plugin**>  
 <**plugin**>  
 <**groupId**>org.apache.maven.plugins</**groupId**>  
 <**artifactId**>maven-compiler-plugin</**artifactId**>  
 <**configuration**>  
 <**source**>1.8</**source**>  
 <**target**>1.8</**target**>  
 <**encoding**>utf-8</**encoding**>  
 </**configuration**>  
 </**plugin**>  
 <**plugin**>  
 <**groupId**>org.apache.tomcat.maven</**groupId**>  
 <**artifactId**>tomcat7-maven-plugin</**artifactId**>  
 <**configuration**>  
 <**path**>/idea-maven</**path**>  
 <**port**>8081</**port**>  
 </**configuration**>  
 </**plugin**>  
 </**plugins**>  
 </**pluginManagement**>  
</**build**>

关于maven项目的断点调试



依赖传递和依赖冲突

什么是依赖传递

加入项目中的依赖所依赖的jar包也会被加入项目中

关于依赖冲突

多个不同版本的依赖同时加入到了项目中

maven自动按照下边的原则调解

1. 第一声明者优先
2. 路径近者优先原则

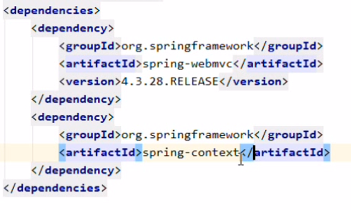
锁定版本

面对众多的依赖，有一种方法不用考虑依赖的路径，声明优化等因素可以采用直接锁定版本的方法确定依赖构件的版本



锁定版本号

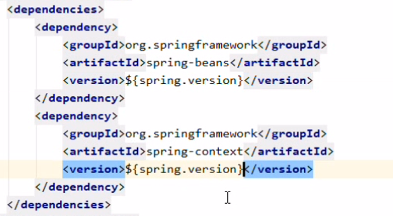
真正的引入依赖



就可以不用去声明版本号了

版本号也可以提取出来放到





分模块构建工程

**继承：创建一个parent工程将所需要的依赖都配置在pom中**

**聚合：聚合多个模块运行**

开发一个多模块的工程

