Ant编译构建Javaweb项目

1. **Ant的概念**  
   Make命令是一个项目管理工具，而Ant所实现功能与此类似。像make，gnumake和nmake这些编译工具都有一定的缺陷，但是Ant却克服了这些工具的缺陷。最初Ant开发者在开发跨平台的应用时，用样也是基于这些缺陷对Ant做了更好的设计。
2. **Ant的构建文件**  
   当开始一个新的项目时，首先应该编写Ant构建文件。构建文件定义了构建过程，并被团队开发中每个人使用。Ant构建文件默认命名为 build.xml，也可以取其他的名字。只不过在运行的时候把这个命名当作参数传给Ant。构建文件可以放在任何的位置。一般做法是放在项目顶层目录中，这样可以保持项目的简洁和清晰。下面是一个典型的项目层次结构。  
   (1) src存放文件。  
   (2) class存放编译后的文件。  
   (3) lib存放第三方JAR包。  
   (4) dist存放打包，发布以后的代码。

把一个WEB项目打成了一个WAR包，其实原理很简单，就是编译所有CLASS的类，然后把它放到一个临时文件夹中。用WRA命令时，会有四个东西，web.xml，lib下的包，webroot下的所有文件，编译后的CLASS文件夹。

这是一个简单的Javaweb项目的Ant编译需要的build.xml文件，并准备在华为软件开发云中的构建服务里进行编译构建。build.xml的标签使用网上有很多教程，很好入门，大家自行学习，就不在这里详细介绍了。

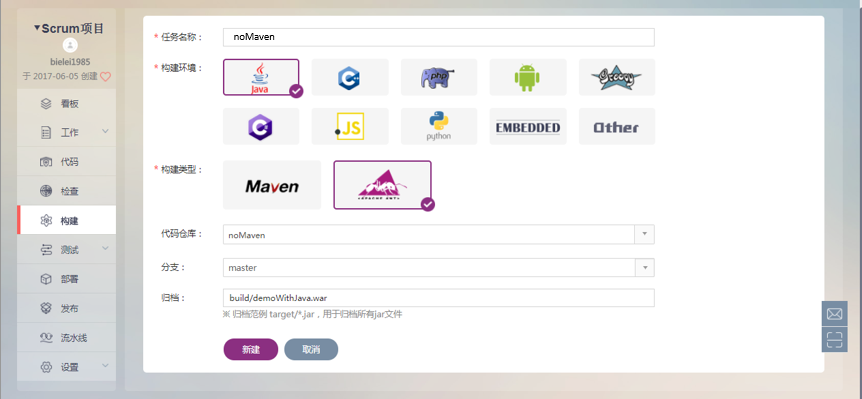
|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" standalone="no"?>  <!-- WARNING: Eclipse auto-generated file.  Any modifications will be overwritten.  To include a user specific buildfile here, simply create one in the same  directory with the processing instruction <?eclipse.ant.import?>  as the first entry and export the buildfile again. -->  <project basedir="." default="generwar" name="demoWithJava">  <property name="build" value="${basedir}/build" />  <property name="build.class" value="${build}/classes" />  <property name="src" value="${basedir}/src" />  <property name="webApp" value="${basedir}/WebContent" />  <property name="lib" value="${webApp}/WEB-INF/lib" />    <!--定义CLASSPACH,编译时用到你外部加入的包-->  <path id="classpath">  <pathelement location="${src}" />  <fileset dir="${lib}" includes="\*\*\*.jar" />  </path>  <!--删除任务-->  <target name="clean" description="delete">  <echo message="delete dir" />  <delete dir="${build}" />  </target>  <!--创建一个临时目录-->  <target name="init" description="create dir" depends="clean">  <echo message="create dir" />  <mkdir dir="${build.class}" />  </target>  <!--编译java源文件-->  <target name="compile" description="compile" depends="init">  <echo message="compile" />  <javac srcdir="${src}" destdir="${build.class}">  <classpath refid="classpath" />  </javac>  </target>  <!--打成WAR包-->  <target name="generwar" depends="compile">  <echo message="generate WAR" />  <war destfile="${build}/${ant.project.name}.war" webxml="${webApp}/WEB-INF/web.xml">  <fileset dir="WebContent">  </fileset>  <lib dir="${lib}" />  <classes dir="${build.class}" />  </war>  </target>  <!--把WAR放到一个指定目录，这里是直接布署到TOMCAT服务器-->  <!--  <target name="fabu" depends="generwar" description="deploy">  <copy file="${build}/${ant.project.name}.war" todir="${webserver}">  </copy>  </target>  -->  <!--把打好的WAR删除-->  <!--  <target name="deleteWar" depends="fabu" description="删除服务器上的war">  <delete dir="${build}" />  </target>  -->    </project> |

大致步骤：

1. 全局变量的定义
2. 定义CLASSPACH,编译时用到你外部加入的包
3. 创建“删除任务”，清空上次编译的生成目录
4. 创建临时目录，编译生成的目录
5. 编译java源文件
6. 打成WAR包
7. 其他，还可以直接部署到Tomcat服务器中等操作
8. **Ant编译构建**
9. build.xml编写好后，部署一个ant服务器
10. 环境变量ANT\_HOME: D:\apache-ant-1.8.0（具体路径根据自己的实际情况）
11. cmd命令行进入build.xml所在的目录，然后输入ant即可将打war，如下图



1. **华为软件开发云中使用Ant编译构建**
2. 新建构建任务



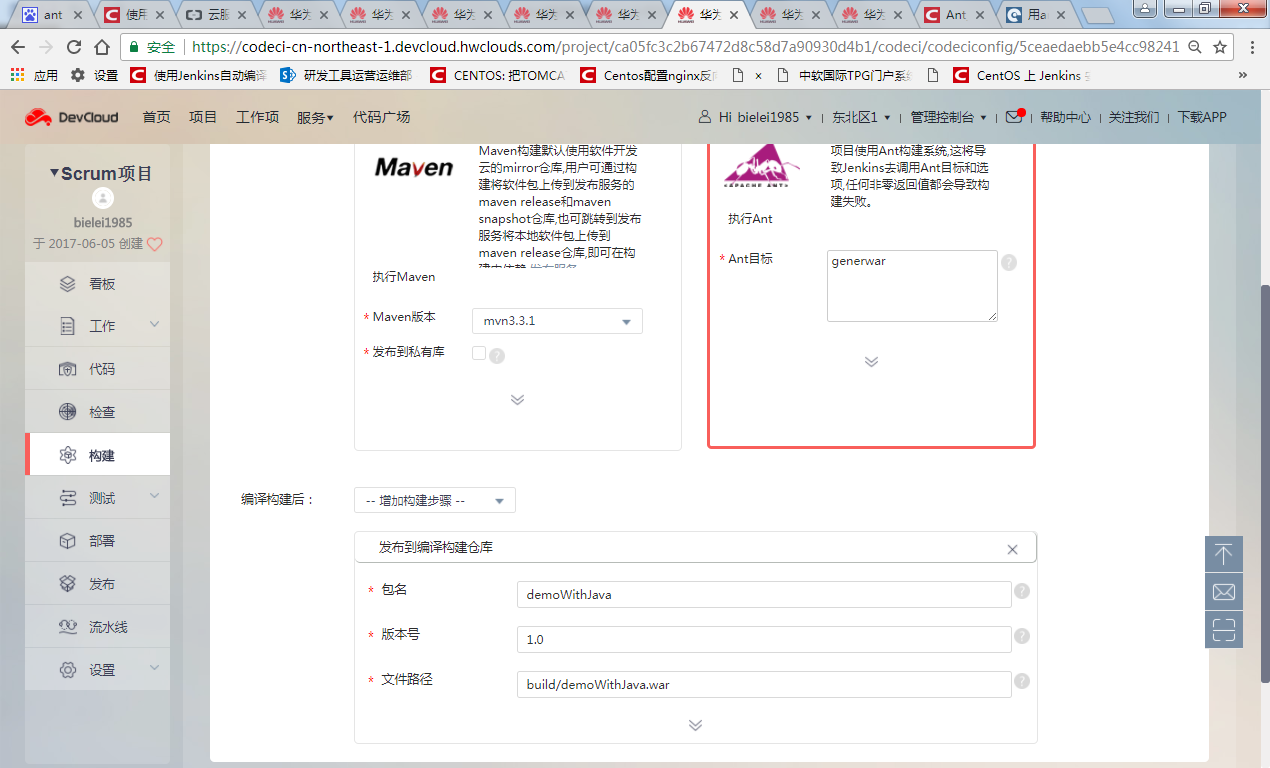
1. 配置任务
2. 切换到“构建配置”菜单。

“Ant目标”需要跟build.xml（本例中build.xml文件放在工程的根目录）文件中的project标签的“default”值一致。

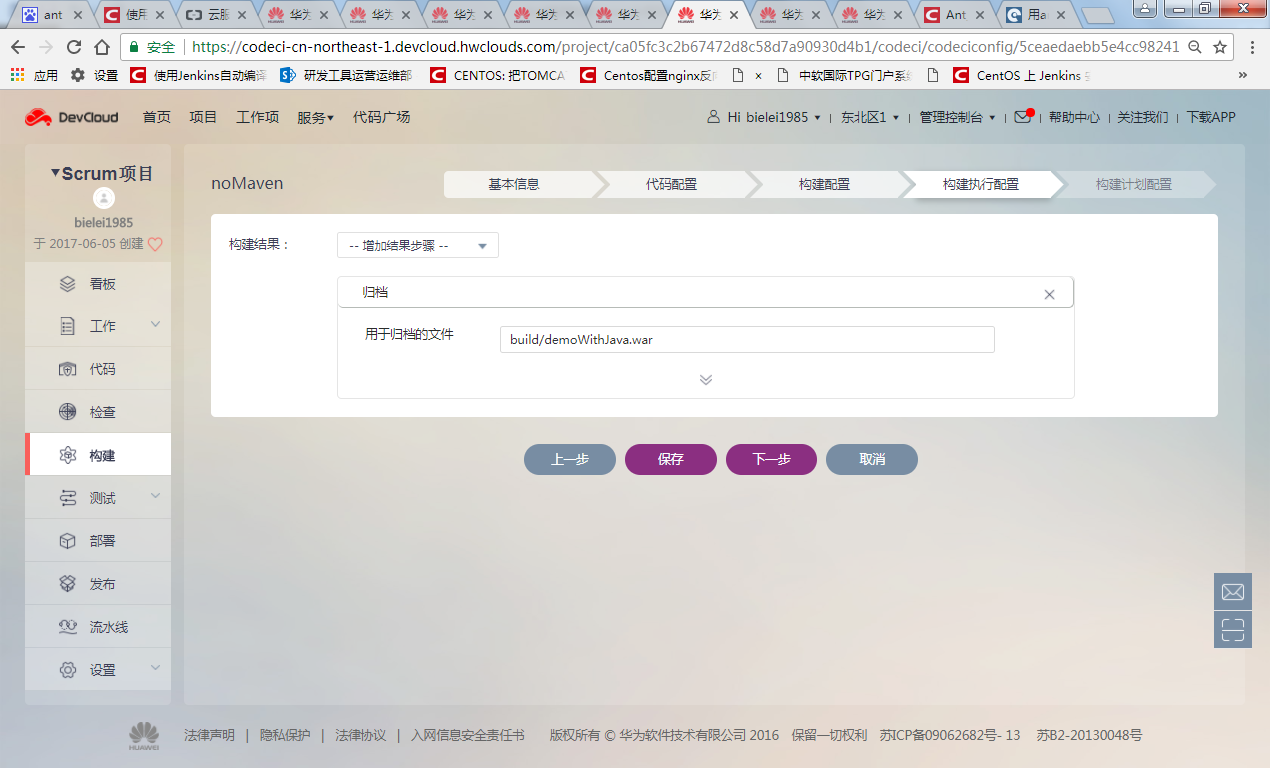


增加编译构建后步骤，“发布到编译构建仓库”，具体内容参考下图。

“文件路径”需要与build.xml中war输出的地址一致。



1. 切换到“构建配置”菜单。

“用于归档的文件”，同样需要与build.xml中war输出的地址一致。

1. 保存，并执行构建任务。

