Atitit 让maven pom.xml不编译

1. **<build>**
2. **<defaultGoal>**install**</defaultGoal>**
3. **<directory>**${basedir}/target**</directory>**
4. **<finalName>**${artifactId}-${version}**</finalName>**
5. ...
6. **</build>**

说明：

* defaultGoal，执行构建时默认的goal或phase，如jar:jar或者package等
* directory，构建的结果所在的路径，默认为${basedir}/target目录
* finalName，构建的最终结果的名字，该名字可能在其他plugin中被改变

maven 缺省会找项目的src/main/java目录进行编译，若找不到，则不编译，除非在pom.xml文件中显示指定源文件：

## **解决：**

在pom.xml中的<build>节点添加以下内容，即可以进行源码编译。

**[html]** [view plain](https://blog.csdn.net/hongweigg/article/details/52598821" \o "view plain) [copy](https://blog.csdn.net/hongweigg/article/details/52598821" \o "copy)

1. **<sourceDirectory>**src**</sourceDirectory>**

注意：加入下面的指令则是将源文件输出到结果包中。

**[html]** [view plain](https://blog.csdn.net/hongweigg/article/details/52598821" \o "view plain) [copy](https://blog.csdn.net/hongweigg/article/details/52598821" \o "copy)

1. **<resources>**
2. **<resource>**
3. **<directory>**src**</directory>**
4. **<includes>**
5. **<include>**\*\*/\*\***</include>**
6. **</includes>**
7. **</resource>**
8. **</resources>**

<resources>

<!-- 这个元素描述了项目相关或测试相关的所有资源路径 -->

<resource>

<!-- 描述了资源的目标路径。该路径相对target/classes目录（例如${project.build.outputDirectory}）。

举个例子，如果你想资源在特定的包里(org.apache.maven.messages)，你就必须该元素设置为

org/apache/maven/messages。然而，如果你只是想把资源放到源码目录结构里，就不需要该配置。 -->

<targetPath></targetPath>

<!-- 是否使用参数值代替参数名。参数值取自properties元素或者文件里配置的属性，文件在filters元素

里列出。 -->

<filtering></filtering>

<!-- 描述存放资源的目录，该路径相对POM路径 -->

<directory>WebContent/WEB-INF\classes</directory>

<!-- 包含的模式列表，例如\*\*/\*.xml. -->

<includes>

<include>\*\*/\*</include>

</includes>

<!-- 排除的模式列表，例如\*\*/\*.xml -->

<excludes>

<exclude>\*.excfile</exclude>

</excludes>

</resource>

</resources>