



链滴

maven 中 filtering 的使用, 打包时排除资源目录, 以及 SpringBoot 下失效的一种情况处理

作者: [zhaozhizheng](#)

原文链接: <https://ld246.com/article/1631761064940>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p></p>

第一种用法</h3>

<p>SpringEL 表达式取值一般是{var}方式取值, 见于application.p</p>

<p>maven 的 pom.xml 文件中也有类似的取值表达式, 也是通过 \${var}的方式取值</p>

<p>然而: 它们并不是一个东西!</p>

<p>EL 表达式适用于配置文件及代码中的注解</p>

<p>maven 的占位符取值表达式默认仅仅适用于 pom.xml 文件中</p>

<p>我们的需求大多是想打通二者的交流,如何做?</p>

<p>此时 filtering 就派上用场了</p>

<p>maven 的占位符解析表达式的使用场合默认只在 pom 文件范围内活动</p>

<p>如果想扩大它的活动范围,就必须指定需要扩大到哪些文件,然后指定 filtering=true.然后 maven 占位符解析表达式就可以用于它里面的表达式解析了.</p>

```
<code class="language-xml highlight-chroma"><span class="highlight-line"><span class="highlight-cl"><span class="highlight-nt">&lt;build&gt;</span></span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-nt">&lt;resources&gt;</span></span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-nt">&lt;resource&gt;</span></span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-nt">&lt;directory&gt;</span>${project.basedir}/src/main/resources<span class="highlight-nt">&lt;/directory&gt;</span></span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-nt">&lt;filtering&gt;</span>true<span class="highlight-nt">&lt;/filtering&gt;</span></span></span></span></code></pre>
```

<p>如果你的项目继承了 spring-boot-starter-parent,filtering 大概率无效</p>

<p>filtering 无效的原因是, pom.xml 继承了 spring boot 的依赖</p>

```
<code class="language-xml highlight-chroma"><span class="highlight-line"><span class="highlight-cl"><span class="highlight-nt">&lt;parent&gt;</span></span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-nt">&lt;groupId&gt;</span>org.springframework.boot<span class="highlight-nt">&lt;/groupId&gt;</span></span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-nt">&lt;artifactId&gt;</span>spring-boot-starter-parent<span class="highlight-nt">&lt;/artifactId&gt;</span></span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-nt">&lt;version&gt;</span>1.5.9.RELEASE<span class="highlight-nt">&lt;/version&gt;</span></span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-nt">&lt;/parent&gt;</span></span></span></span></code></pre>
```

<p>点开这个依赖的 pom.xml, 我们发现,spring boot 把默认的占位符号 \${}改成了 @</p>

<p>那么如果想在 application.properties 中使用 maven 的占位符以获取 maven 的参数,此时就不用 \${},因为可能会与 EL 表达式冲突。解决方法如下: </p>

- 此时 parent 依赖将 maven 占位符表达式默认改为 @var@ 的形式. 所以只能用 @var@ 替代 \${

ar}

在 pom.xml 里面添加如下内容


```
<pre> <code class="language-xml highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> &lt;properties&gt;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> &lt;/resource.delimiter&gt;</span> ${</span> <span class="highlight-nt"> &lt;/resource.d
limiter&gt;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> &lt;/properties&gt;</span>
</span> </span> </code> </pre>
```

<ol start="3">

application.properties 里面不用 \${}, 改成 @ 例如:


```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> jdbc.driverClassName=@jdbc.driverClassName@
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> jdbc.url=@jdbc.
url@
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> jdbc.username
@jdbc.username@
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> jdbc.password
@jdbc.password@
</span> </span> </code> </pre>
```

<h3 id="第二种用法">第二种用法</h3>

<p>在 springboot 中整合了 Vue.js。 </p>

<p>我把 Vue.js 的前端项目放在了 resources/webapp 目录下, 当 Vue.js build 出结果到 resource /statics 下以后, 这个 webapp 作为源码目录是不应该被打包到最后部署的 jar 包中的。于是想把这个 resource/webapp 目录给屏蔽掉。因为一直保持着最小改动的原则, 这里的方法我并没有具体的测试, 因为它不仅增加了一个打包插件, 而且还对 springboot 自带的打包插件增加了设置。
文章开头提到的这种基于 resources 标签的方法非常符合我心中的方式, 经过尝试, 终于用这种方式成功的排除掉了 resources/webapp。 </p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> &lt;build&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> &lt;plugins&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> &lt;plugin&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> &lt;groupId&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> &lt;org.springframework.boot&lt;/groupId&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> &lt;artifactId&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> &lt;spring-boot-maven-plugin&lt;/artifactId&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> &lt;/plugin&
t;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> &lt;/plugins&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> &lt;resources&
t;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> &lt;resource
t;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> &lt;filterin
&gt;true&lt;/filtering&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> &lt;directo
ry&gt;src/main/resources&lt;/directory&gt;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-nt"> &lt;exclud
s&gt;
</span> </span> </code>
```

```

</span></span><span class="highlight-line"><span class="highlight-cl">      &lt;/excl
de&gt;webapp/**&lt;/exclude&gt;
</span></span><span class="highlight-line"><span class="highlight-cl">      &lt;/exclu
es&gt;
</span></span><span class="highlight-line"><span class="highlight-cl">      &lt;/resourc
&gt;
</span></span><span class="highlight-line"><span class="highlight-cl">      &lt;/resources
&gt;
</span></span><span class="highlight-line"><span class="highlight-cl">&lt;/build&gt;
</span></span></code></pre>
<p>首先一个不一样的地方就是，增加了一个 filtering 节点并且设置值为 true</p>
<p>然后在 excludes 中用/**匹配了 webapp 下面的所有内容</p>
<p>根据这种原则，我们在打包时只想包含指定的资源文件，可以用标签来实现。</p>
<pre><code class="language-xml highlight-chroma"><span class="highlight-line"><span cl
ss="highlight-cl"><span class="highlight-nt">&lt;/resources&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">      <span class=
highlight-nt">&lt;/resource&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">      <span clas
="highlight-nt">&lt;/filtering&gt;</span>true<span class="highlight-nt">&lt;/filtering&gt;</
span>
</span></span><span class="highlight-line"><span class="highlight-cl">      <span clas
="highlight-nt">&lt;/directory&gt;</span>src/main/resources<span class="highlight-nt">&lt;
directory&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">      <span clas
="highlight-nt">&lt;/includes&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">      <span c
ass="highlight-nt">&lt;/include&gt;</span>webapp/**<span class="highlight-nt">&lt;/inclu
e&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">      <span clas
="highlight-nt">&lt;/includes&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">      <span class=
highlight-nt">&lt;/resource&gt;</span>
</span></span><span class="highlight-line"><span class="highlight-cl">      <span class="h
ghlight-nt">&lt;/resources&gt;</span>
</span></span></code></pre>
<p>如此，只有 webapp 下面的资源文件才能被打包到目标 jar 包中。</p>
<p>这种方式来排除资源，即简洁，又能够很容易就明白其中的原因。</p>
<p>总结:</p>
<ol>
<li>Spring EL 表达式和 MAVEN 的占位符表达式长得一样,但两者默认进水不犯河水,不能再 Spring
范围内取 maven 的参数</li>
<li>filtering 的作用就是打通两者的连接, 让井水犯河水, 具体来说是让 Spring 的范围内能取到 mave
的参数</li>
<li>filtering 的使用要配合 resource 一起使用: 前者开启打通连接,后者指定打通的范围.</li>
</ol>

```