



链滴

[原创]springboot 中 resources 资源目录里 面的文件夹压缩下载

作者: [washmore](#)

原文链接: <https://ld246.com/article/1557478477841>

来源网站: [链滴](#)

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

<h2 id="前言">前言</h2>

<p>最近做个小工具需要提供一个将 resources 资源文件夹下某个目录(放了一些模板集合)打包下载能</p>

<h2 id="尝试">尝试</h2>

<p>祖传的 zip 文件夹功能代码先送上:</p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> public void zip(ZipOutputStream out, File sourceFile, String base) throws Exception {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">         //如果路径为
录 (文件夹)
</span> </span> <span class="highlight-line"> <span class="highlight-cl">         if (sourceFile.
sDirectory()) {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">             //取出文
夹中的文件 (或子文件夹)
</span> </span> <span class="highlight-line"> <span class="highlight-cl">             File[] fileLis
= sourceFile.listFiles();
</span> </span> <span class="highlight-line"> <span class="highlight-cl">             if (fileList.l
ngth == 0) {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">                 //如果
文件夹为空, 则只需在目的地zip文件中写入一个目录进入点
</span> </span> <span class="highlight-line"> <span class="highlight-cl">                 System.
ut.println(base + "/");
</span> </span> <span class="highlight-line"> <span class="highlight-cl">                 out.put
extEntry(new ZipEntry(base + "/"));
</span> </span> <span class="highlight-line"> <span class="highlight-cl">             } else {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">                 //如果
文件夹不为空, 则递归调用zip, 文件夹中的每一个文件 (或文件夹) 进行压缩
</span> </span> <span class="highlight-line"> <span class="highlight-cl">                 for (File f
le : fileList) {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">                     zip(ou
, file, base + "/" + file.getName());
</span> </span> <span class="highlight-line"> <span class="highlight-cl">                 }
</span> </span> <span class="highlight-line"> <span class="highlight-cl">             } else {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">                 //如果不
目录 (文件夹), 即为文件, 则先写入目录进入点, 之后将文件写入zip文件中
</span> </span> <span class="highlight-line"> <span class="highlight-cl">                 out.putNe
tEntry(new ZipEntry(base));
</span> </span> <span class="highlight-line"> <span class="highlight-cl">                 IOUtils.wri
e(FileUtils.readFileToByteArray(sourceFile), out);
</span> </span> <span class="highlight-line"> <span class="highlight-cl">                 out.flush();
</span> </span> <span class="highlight-line"> <span class="highlight-cl">             }
</span> </span> <span class="highlight-line"> <span class="highlight-cl">         }
</span> </span> </code> </pre>
```

<p>最开始是针对单个文件下载,很简单,通过 <code>this.getClass().getResourceAsStream("/templates/demo.xml")</code> 获取到指定文件的输入流,然后写入到 <code>response.getOutputStream()</code> 中去即可;

然后依样画葫芦针对文件夹下载,<code>this.getClass().getResourceAsStream("/templates")</code> 获取到文件夹的输入流,然鹅输出发现这个输入流拿到的信息是</p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> file1.xml
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> file2.xml
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> dictionary1
</span> </span> </code> </pre>
```

<p>这样的内容,而祖传 zip 第二个参数要求的是一个文件夹目录 File 对象,不太好整;

换个方式:</p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">
cl">      OutputStream ops = response.getOutputStream();
</span> </span> <span class="highlight-line"> <span class="highlight-cl">      ZipOutput
tream out = new ZipOutputStream(ops);
</span> </span> <span class="highlight-line"> <span class="highlight-cl">      File parent
= new File(this.getClass().getResource("/templates").getFile());
</span> </span> <span class="highlight-line"> <span class="highlight-cl">      zip(out, pare
t, "");
</span> </span> <span class="highlight-line"> <span class="highlight-cl">      out.close();
</span> </span> <span class="highlight-line"> <span class="highlight-cl">      ops.flush();
</span> </span> <span class="highlight-line"> <span class="highlight-cl">      ops.close();
</span> </span> </code> </pre>
```

<p>通过拿到资源文件目录 <code>/templates</code> 所在的 File 信息,然后基于 <code>respon
e</code> 的输出流生成 <code>ZipOutputStream</code>,调用 zip 方法压缩.搞定!</p>

<h2 id="麻烦">麻烦</h2>

<p>自测通过后打包成 jar 执行,问题出现了,会报错</p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">
cl">java.io.FileNotFoundException: File 'file:...jar!/BOOT-INF/classes!/templates' does not exist
</span> </span> </code> </pre>
```

<p>这是因为将应用打包成 jar 后,<code>File parent = new File(this.getClass().getResource("/t
emplates").getFile());</code> 这行代码不再能正确获取到 <code>/templates</code> 所在的文
目录信息,导致下载失败!</p>

<h2 id="解决">解决</h2>

<p>去 TMD 的 百度搜索,全给推荐 csdn 和 cnblogs 的文章,也不知道谁抄谁的,千篇一

<code>File parent = new File(this.getClass().getResource("/templates").getFile());</code> 换

<code>InputStream ips = this.getClass().getResourceAsStream("/templates/demo.xml")</co
e> 大法,可我他喵的要下载文件夹啊!!!已拉黑!!!</p>

<h3 id="探索">探索</h3>

<p>想着既然能通过 <code>getResourceAsStream</code> 获取到输入流,那我干脆自行遍历 <co
e>/templates</code> 资源文件夹,然后逐个转移到临时文件夹目录,然后针对临时文件夹打包下载.<
r>

说做就做!!!

将 <code>this.getClass().getResourceAsStream("/templates")</code> 获取到的输入流</p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">
cl">file1.xml
</span> </span> <span class="highlight-line"> <span class="highlight-cl">file2.xml
</span> </span> <span class="highlight-line"> <span class="highlight-cl">dictionary1
</span> </span> </code> </pre>
```

<p>进行遍历,然后又傻逼了...我倒是知道 <code>file2.xml</code> 是文件 <code>dictionary1</
ode> 是文件夹,针对文件夹还要往下层遍历,但是代码不知道啊?

千里之行死于足下...这可咋整?</p>

<h3 id="发现">发现</h3>

<p>一番上上下下左左右右 BABA 操作之后发现, <code>getResourceAsStream</code>
方法如果参数是文件夹那返回的输入流的具体类型是 <code>ByteArrayInputStream</code>,而
对文件,输入流的具体类型是 <code>BufferedInputStream</code>,

这就好办了 <code>ips instanceof ByteArrayInputStream</code> 约等于 <code>file.isDirector
()</code> 的效果嘛.</p>

<h3 id="实施">实施</h3>

<p>现在整体思路就很明朗了,先将 <code>/templates</code> 资源目录复制到临时文件夹中保存
然后针对临时文件夹进行 zip 压缩,然后输出给 <code>response</code> 完成打包下载功能;

下面是将 <code>/templates</code> 资源目录复制到临时文件夹的代码:</p>

```

<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> public void copyResourcesToTempDictionary(String sourceParentPath, String name, File
tempParent) throws Exception {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> String path =
sourceParentPath + "/" + name;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> InputStream
ps = this.getClass().getResourceAsStream(path);
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> File file = ne
File(tempParent, name);
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> if (file.exists())
{
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> file.delete()

</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> if (ips instanc
of ByteArrayInputStream) {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> //文件夹
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> file.mkdirs(
;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> List<Stri
g>&gt; children = IOUtils.readLines(ips, StandardCharsets.UTF_8);
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> if (Collecti
nUtils.isEmpty(children)) {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> return;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> for (String
child : children) {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> copyRe
sourcesToTempDictionary(path, child, file);
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> } else if (ips i
stanceof BufferedInputStream) {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> file.create
newFile();
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> FileUtils.wr
teByteArrayToFile(file, IOUtils.toByteArray(ips));
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> </code> </pre>
<p>整体流程调用代码(设置响应头/编码/文件名等操作略):</p>
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> OutputStream ops = response.getOutputStream();
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> ZipOutput
stream out = new ZipOutputStream(ops);
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> File parent
= new File(System.getProperty("java.io.tmpdir"), "~tmp");
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> if (parent.e
ists()) {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> parent.
delete();
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> parent.mk
irs();

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">      copyResou
cesToTempDictionary("", "templates", parent);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">      zip(out, pa
ent, "");
</span></span><span class="highlight-line"><span class="highlight-cl">      out.close();
</span></span><span class="highlight-line"><span class="highlight-cl">      ops.flush();
</span></span><span class="highlight-line"><span class="highlight-cl">      ops.close();
</span></span></code></pre>
<h2 id="Updates">Updates</h2>
<p>既然 zip 是从 source 写到输出流,这个 sources 既可以是 File,当然也可以来自输入流嘛,于是忍
住对祖传的 zip 方法下手了,针对这种 <code>resources</code> 文件夹的压缩新增一个 <code>zi
Resources</code> 的方法:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">      public void zipResources(ZipOutputStream out, String sourceParentPath, String name)
hrows Exception {
</span></span><span class="highlight-line"><span class="highlight-cl">      String path =
sourceParentPath + "/" + name;
</span></span><span class="highlight-line"><span class="highlight-cl">      InputStream
ps = this.getClass().getResourceAsStream(path);
</span></span><span class="highlight-line"><span class="highlight-cl">      if (ips instanc
of ByteArrayInputStream) {
</span></span><span class="highlight-line"><span class="highlight-cl">      //取出文
夹中的文件（或子文件夹）
</span></span><span class="highlight-line"><span class="highlight-cl">      List<Stri
g>&gt; children = IOUtils.readLines(ips, StandardCharsets.UTF_8);
</span></span><span class="highlight-line"><span class="highlight-cl">      if (Collecti
nUtils.isEmpty(children)) {
</span></span><span class="highlight-line"><span class="highlight-cl">      //如果
文件夹为空，则只需在目的地zip文件中写入一个目录进入点
</span></span><span class="highlight-line"><span class="highlight-cl">      out.put
extEntry(new ZipEntry(sourceParentPath));
</span></span><span class="highlight-line"><span class="highlight-cl">      } else {
</span></span><span class="highlight-line"><span class="highlight-cl">      for (Stri
g child : children) {
</span></span><span class="highlight-line"><span class="highlight-cl">      zipRe
sources(out, path, child);
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">      } else {
</span></span><span class="highlight-line"><span class="highlight-cl">      //如果不
目录（文件夹），即为文件，则先写入目录进入点，之后将文件写入zip文件中
</span></span><span class="highlight-line"><span class="highlight-cl">      out.putNe
tEntry(new ZipEntry(path));
</span></span><span class="highlight-line"><span class="highlight-cl">      IOUtils.wri
e(IOUtils.toByteArray(ips), out);
</span></span><span class="highlight-line"><span class="highlight-cl">      out.flush();
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span><span class="highlight-line"><span class="highlight-cl">      }
</span></span></code></pre>
<p>这样一来,就不需要借助临时文件夹中转了,整体流程调用可简化为:</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">      OutputStream ops = response.getOutputStream();
</span></span>

```


<pre> stream out = new ZipOutputStream(ops); es(out, "", "templates"); </code></pre> </pre>	ZipOutput
<pre> <p>真是机智的骚年!</p> <h2 id="Q-A">Q&amp;A</h2> <p>文中有用到一些 IO 操作 utils 来自 commons 系列,附 maven 地址:</p> <pre><code class="highlight-chroma"> &lt;dependency&gt; &gt;commons-io&lt;/groupId&gt; &lt;artifact d&gt;commons-io&lt;/artifactId&gt; &lt;versio &gt;2.5&lt;/version&gt; &lt;/depend ncy&gt; </code></pre> </pre>	zipResour
<pre> <p>真是机智的骚年!</p> <h2 id="Q-A">Q&amp;A</h2> <p>文中有用到一些 IO 操作 utils 来自 commons 系列,附 maven 地址:</p> <pre><code class="highlight-chroma"> &lt;dependency&gt; &gt;commons-io&lt;/groupId&gt; &lt;artifact d&gt;commons-io&lt;/artifactId&gt; &lt;versio &gt;2.5&lt;/version&gt; &lt;/depend ncy&gt; </code></pre> </pre>	out.close(); ops.flush(); ops.close();
<pre> <p>各位如果有完成过类似的 case,有更优雅或更合适的方案的话,欢迎在评论指出.</p> <h2 id="One-More-Thing">One More Thing</h2> <p>我岳母身患骨髓增生异常综合征伴骨髓纤维化,急需筹钱做骨髓移植手术,方便的话转请大家帮忙 发一下朋友圈,感谢大家!
 轻松筹地址:https://m2.qschou.com/project/love/love_v7.html?projuu d=23a9dbd5-78e3-429f-8b46-c7efce4a9443</p> </pre>	<groupI <artifact <versio </depend