链滴

qrc 资源文件供外部使用的方法

作者：taujiong
原文链接：https://ld246.com/article/1577769180029
来源网站：链滴

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/bing/20190417.jpg?imageView2/1/w/960/h/540/interlace/1/q/100">
从入学之初就跟着师兄一起做一个 PyQt5 的项目，学到了很多东西。自从上次项目打包遇到资
文件处理难题之后，就开始慢慢在项目中引入 Qt 提供的 qrc 资源文件方案。
最近在系统地学 PySide2（Qt for python 的官方绑定），在其官方教程中有一个 <a href="htt
s://ld246.com/forward?goto=https%3A%2F%2Fdoc.qt.io%2Fqtforpython%2Ftutorials%2Fdata
isualize%2Findex.html" target="_blank" rel="nofollow ugc">Data Visualization Tool Tutorial<
a> 的数据可视化教程。在该教程中，用到了 pandas 库来读取一个本地 CSV 文件。教程本身提供的
相对路径读取文件方法，但本人想使用 qrc 资源文件的方式来引入该 CSV 文件，以培养习惯。不曾
，遇到了 pandas 无法读取 CSV 文件的错误，在解决过程中对 qrc 资源文件的运作方式有了更深的
解。

<h2 id="问题重现">问题重现</h2>
首先我的项目目录结构为
<pre><code class="language-bash highlight-chroma"><span c
ass="highlight-cl">D:.
└─DataVisualizati
n
    │ main.py
    │
    ├─layouts
    │   main.ui
    │
    └─resources
        │ all_day.csv
        │ data.qrc
        │ data_rc.py
</code></pre>
data.qrc 代码为
<pre><code class="language-xml highlight-chroma"><span cl
ss="highlight-cl"><RCC>
  <span class="hi
hlight-nt"><qresource prefix=<span class="h
ghlight-s">"data">
    <span class="h
ghlight-nt"><file>all_day.csv</file>
  <span class="hi
hlight-nt"></qresource>
<span class="high
ight-nt"></RCC>
</code></pre>
在 main.py 中，按照正常的相对路径写法，能够正确打印 CSV 文件内容，而使用 qrc 资源文件
法则不可行，具体代码为
<pre><code class="language-python highlight-chroma"><spa
 class="highlight-cl">import <span class="highlight-nn
>pandas as pd</sp
n>
<span class="high
ight-kn">from PySide2.QtCore <span class="h
ghlight-kn">import QFile<span class="highlight
p">, QIODevice

<span class="high

ight-kn">import DataVisualization.resources.data_rc

if __name__ == "__main__":
    data = pd.read_csv(":/data/all_day.csv")
    # data = pd.read_csv(r"DataVisualization\resources\all_day.csv")
    print(data)
</code></pre>
运行此 main.py 报错 <code>FileNotFoundError</code> ，而使用注释内的语句则能够正常示 CSV 文件内容。
<pre><code class="language-bash highlight-chroma">FileNotFoundError: [Errno 2] File b':/data/all_day.csv' does not exist: b':/data/all_day.csv'
</code></pre>
<h2 id="解决方案">解决方案</h2>
经过长时间的谷歌，终于从 stack overflow 找到了解决方案及原因。
总的来说，就是因为作为 Qt 内部使用的资源管理方案，只有 Qt 本身知道如何正确地从该资源件中得到所需文件正确的路径并读取，而外界库甚至 python 自身也不能得到文件的具体路径。
既然外界读不到信息是因为找不到文件的具体路径，那么解决方案也就呼之欲出了：跳过路径，接让 Qt 程序告诉第三方库所需文件的内容。
<pre><code class="language-python highlight-chroma">import io
import pandas as pd
from PySide2.QtCore import QFile, QIODevice

import DataVisualization.resources.data_rc

if __name__ == "__main__":
    file = QFi

e(":/data/all_day.csv
)
    <span class="h
ghlight-k">if file.</s
an>open(<span clas
="highlight-n">QIODevice.<span class="highlig
t-n">ReadOnly):
        <span class=
highlight-n">f = io<
span>.BytesIO<span
class="highlight-p">(file<span class="highlight
o">.readAll()</span
.data<span class="h
ghlight-p">())
        <span class=
highlight-n">data = <span class="highlight-n"
pd.read_csv
span class="highlight-p">(f<span class="highli
ht-p">)
        <span class=
highlight-nb">print(
ata)
</code></pre>
值得注意的是，这里 pandas.read_csv()通过此方案可行的原因是，read_csv()支持传入的参数可以是文件路径也可以是 buffer。对于其他某些接口可能必须要文件路径的则此路不通。
<h2 id="深入思考">深入思考</h2>
<h3 id="qrc-的原理">qrc 的原理</h3>
直接上结论：qrc 文件本身类似于一个 XML 格式的文本，其记录了所包含的每一个文件的路径关键在于，使用 rcc/pyrcc5/pyside2-rcc 对该文件进行编译时，程序遍历 qrc 文件里面的每一个文路径，将该文件的二进制内容写入到相应的输出文件里。
例如，本文使用的 all_day.csv 文件是一个纯文本文件，其文件大小为 56.4KB；包含该文件路径息的 data.qrc 文件大小为 85 字节；通过 pyside2-rcc.exe 编译生成的 data_rc.py 文件大小为 70.7K。此外，当我编译完成了 data_rc.py 之后，即使我删除 all_day.csv 文件，程序照样能够正常运行并取 all_day.csv 里面的数据。除了 pyside2-rcc.exe 将 all_day.csv 的所有信息统统写进了编译生成的 ata_rc.py 这种解释外，我实在想不到为什么一个 py 文件能够达到 70.7KB。
有了这样的结论，也就很好解释 qrc 文件的一些相关特性了，例如：

目标程序里所有内嵌的资源文件都是只读的，在程序运行时不能修改资源里的文件，只能读取使。
当你修改了 qrc 资源文件包含的任意文件之后，你必须重复添加-编译的操才能使更改生效。
除了 Qt 之外的程序想要直接从编译后的文件里面获取信息几乎是不可能的，必须要借助 Qt 内接口读取资源文件信息。

<h3 id="qrc-使用的其他注意事项">qrc 使用的其他注意事项</h3>

为了管理方便，需要编译进 qrc 资源文件的那些文件最好放在 qrc 的所在文件夹或者其子文件夹。
在不加前缀（prefix）的前提下，在程序中调用同级资源文件可以使用 <code>(":file_name")</ode> 如 <code>file = QFile(":/all_day.csv")</code> ；调用子目录资源文件可以用 <code>(":pat/to/file")</code> 如 <code>file = QFile(":/resources/all_day.csv")</code> 。
在加前缀的情况下，在程序中调用同级资源文件可以使用 <code>(":/prefix/file_name")</code 如 <code>file = QFile(":/data/all_day.csv")</code> ；调用子目录资源文件可以用 <code>(":/prfix/path/to/file")</code> 如 <code>file = QFile(":/data/resources/all_day.csv")</code> 。</l

>
对于文件大小超过 4M 的文件，不建议直接编译，而是使用选项 <code>-binary</code> ，具文档看这里。另，在 Qt 的 python 绑定中不支持 <code>-binary</code> 。

<h2 id="总结">总结</h2>
作为一个 PySide2 的入门者，我还有很多东西要学习，包括 qrc 也还有很多方面是一知半解的如果对于 qrc 有什么好的见解，或者文章中有什么错误，希望能够多多交流。