《零基础学编程——Python》期末小组作业实验报告

**作品名称：python爬虫之火车票查询**

**成员介绍**：

**周宇烁、商东庆**：负责代码

**胡健龙**：界面制作

**吴旭、王欣悦、李梦真、张凡硕**：ppt制作和提出创意

**一、作品简介：**该作品利用python中比较方便的某些网页解析和获取库，爬取网站，分析网页，从而解析出我们需要的某些特定的信息。而且同学们还可以下载相关城市的图片作为参考

1. **应用场景：**同学们寒假回家或者外出旅游时需要查询火车票时，就可以利用该作品简单的查询一下回家的火车票情况，所有的火车票信息都是基于12306这个官网上的信息。
2. **总体设计**

**（1）基本思路**：借鉴老师在课堂上讲的天气爬虫这个实例，我们利用bs4库和json库来对我们需要爬取的网页进行数据分析和爬取，最后在利用一个爬虫图片下载程序，让用户可以下载查看相关城市的图片。

**（2）主要技术难点和解决方案：**

**主要技术难点：**

1、因为我们发现该网站上的车票信息是通过动态网页的方式生成的，在我们爬取信息时造成了一定的技术困难。

2、网页所生成的url和动态网页里的数据存储不规律，无法批量解析

3、下载相关城市的图片时，不能做到精确下载，可能会有一些与该城市无关的图片出现

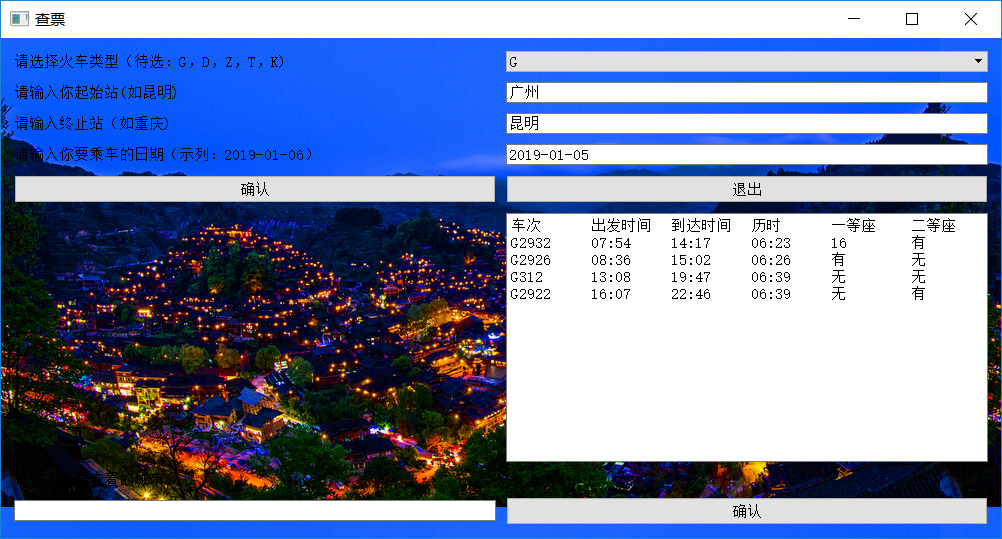
**解决方案：**

1、找到相关动态网页生成的url，利用json库返回js脚本中的数据从而进行解析

1. 分批量解析，通过仔细寻找，我们发现某些数据中仍然存在联系，我们把存在联系的数据分批解析出来，存入同一个列表中，最后同时显示出来
2. 暂无解决方案
3. **特色和创新点**

特色和创新：用户可以下载相关城市的图片，作为到达该城市之前的了解。用户还可以选择需要乘坐的列车类型，我们会进行筛选并显示出来

1. **运行截图**

****

1. **项目组成员的工作心得**

通过参与项目，请每一个成员从学习能力、独立思考的能力、发现问题的能力以及其他有参考价值的方面分别独立撰写心得，格式如下：

**成员一：周宇烁，2017级，项目负责人**

**学习能力方面：提高了自我学习能力。**

**独立思考能力方面：通过本次作业，不仅锻炼了自我思考的能力，也培养了合作的意识**

**发现问题能力方面：发现问题能力更加提高了。**

**其它方面：无**

**成员二：商东庆，2017级，组员**

**学习能力:提高了自主学习能力，更深入的了解了Python的语法和功能**

**独立思考方面:当遇到麻烦和问题时，学会了依靠自己独立思考解决问题**

**发现问题能力方面:在Debug时提高了发现问题的能力**

**成员三：胡建龙，2017级，组员**

**学习能力:提高了查资料的能力**

**独立思考能力:提高了独立思考发现方法的能力**

**发现问题能力方面:提高了自己发现问题的能力**

**成员四：王欣悦，2017级，组员**

**提高了团队合作的能力，另外学习了Python的语法和操作**

**成员五：吴旭，2017级，组员**

**在大家一起进行的活动中学习到了很多东西，包括团队合作能力和发现及解决问题的能力**

**成员六：张凡硕，2016级，组员**

**第一次接触到了编程，感觉有一定的难度，但是收获也很多，学习到了一些Python基础的语法**

**成员七：李梦真，2016级，组员**

**收获到了很多新知识，另外对思维方式也有一定的启发和转变。**

1. 存在的问题、建议及其他需要说明的情况

我们在优化方面还存在很多问题，比如说我们只能爬取出来车票的信息，无法实现用户的购票服务，另外，我们在爬取图片的时候会有些卡顿，不是很流畅，可能还需要加入一些进度条来提高用户的体验感，还可以加入一些短信服务来提醒乘客的乘车时间。

1. 附件：代码

import sys, requests, json, re, urllib

from bs4 import BeautifulSoup

from PyQt5.QtWidgets import QWidget, QLabel, QTextEdit, QPushButton, QComboBox,\

QLineEdit, QGridLayout, QDesktopWidget, QApplication

from PyQt5.QtGui import QPalette, QBrush, QPixmap

class Example(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.train\_type = "G"

self.start\_place = ""

self.end\_place = ""

self.date = ""

self.picture = ""

self.initUI()

def initUI(self):

# 网格布局

grid = QGridLayout()

# 设置组件间隔10

grid.setSpacing(10)

# 放到屏幕中央

self.resize(1000, 500)

self.center()

# 设置背景

palette1 = QPalette()

palette1.setBrush(self.backgroundRole(), QBrush(QPixmap('3.jpg'))) # 设置背景图片

self.setPalette(palette1)

# 文本显示

self.QLtrain\_type = QLabel("请选择火车类型（待选：G，D，Z，T，K)")

self.QLstart = QLabel("请输入你起始站(如昆明)")

self.QLend = QLabel("请输入终止站（如重庆)")

self.QLdate = QLabel("请输入你要乘车的日期（示列：2019-01-06）")

self.QLpicture = QLabel("请输入你要查看的城市")

# 文本编辑

#下拉选单组件

self.QCtrain\_type = QComboBox(self)

self.QCtrain\_type.addItem("G")

self.QCtrain\_type.addItem("D")

self.QCtrain\_type.addItem("Z")

self.QCtrain\_type.addItem("T")

self.QCtrain\_type.addItem("K")

# QLineEdit是单行文本编辑框组件

self.QEstart = QLineEdit()

self.QEend = QLineEdit()

self.QEdate = QLineEdit()

self.QEpicture = QLineEdit()

# QTextEdit是多行文本编辑框组件

self.textEdit = QTextEdit()

# QPushButton是按钮组件

self.btn1 = QPushButton("确认", self)

self.btn2 = QPushButton("退出", self)

self.btn3 = QPushButton("确认", self)

# 增加组件到widget

grid.addWidget(self.QLtrain\_type, 1, 0)

grid.addWidget(self.QCtrain\_type, 1, 1)

grid.addWidget(self.QLstart, 2, 0)

grid.addWidget(self.QEstart, 2, 1)

grid.addWidget(self.QLend, 3, 0)

grid.addWidget(self.QEend, 3, 1)

grid.addWidget(self.QLdate, 4, 0)

grid.addWidget(self.QEdate, 4, 1)

grid.addWidget(self.btn1, 5, 0)

grid.addWidget(self.btn2, 5, 1)

grid.addWidget(self.textEdit, 6, 1)

grid.addWidget(self.QLpicture, 7, 0)

grid.addWidget(self.QEpicture, 8, 0)

grid.addWidget(self.btn3, 8, 1)

self.setLayout(grid)

# 窗口的标题

self.setWindowTitle('查票')

# 如果各个框内容变化，更新

self.QCtrain\_type.activated[str].connect(self.onActivated1)

self.QEstart.textChanged[str].connect(self.onActivated2)

self.QEend.textChanged[str].connect(self.onActivated3)

self.QEdate.textChanged[str].connect(self.onActivated4)

self.QEpicture.textChanged[str].connect(self.onActivated5)

# 三个按钮的事件和槽绑定

self.btn1.clicked.connect(self.btnPress1\_clicked)

self.btn2.clicked.connect(self.btnPress2\_clicked)

self.btn3.clicked.connect(self.btnPress3\_clicked)

self.show()

def onActivated1(self, text):

self.train\_type = text

# 起始地输入框变化

def onActivated2(self, text):

self.start\_place = text

# 终点地输入框变化

def onActivated3(self, text):

self.end\_place = text

# 乘车日期输入框变化

def onActivated4(self, text):

self.date = text

# 城市图片信息变化

def onActivated5(self, text):

self.picture = text

# 将窗口移动到中间

def center(self):

qr = self.frameGeometry()

cp = QDesktopWidget().availableGeometry().center()

qr.moveCenter(cp)

self.move(qr.topLeft())

# 图片下载

def get\_onepage\_urls(self, onepageurl):

"""获取单个翻页的所有图片的urls+当前翻页的下一翻页的url"""

# if not onepageurl:

# print('已到最后一页, 结束')

# return [], ''

try:

html = requests.get(onepageurl)

html.encoding = 'utf-8'

html = html.text

except Exception as e:

print(e)

pic\_urls = []

fanye\_url = ''

return pic\_urls, fanye\_url

pic\_urls = re.findall('"objURL":"(.\*?)",', html, re.S)

fanye\_urls = re.findall(re.compile(r'<a href="(.\*)" class="n">下一页</a>'), html, flags=0)

fanye\_url = 'http://image.baidu.com' + fanye\_urls[0] if fanye\_urls else ''

return pic\_urls, fanye\_url

def down\_pic(self, pic\_urls):

"""给出图片链接列表, 下载所有图片"""

ct = 0

for i, pic\_url in enumerate(pic\_urls):

try:

pic = requests.get(pic\_url, timeout=15)

string = str(i + 1) + '.jpg'

with open(string, 'wb') as f:

f.write(pic.content)

print('成功下载第%s张图片: %s' % (str(i + 1), str(pic\_url)))

ct = ct + 1

if (ct > 10): # 下载11张图片

break

except Exception as e:

print('下载第%s张图片时失败: %s' % (str(i + 1), str(pic\_url)))

print(e)

sys.exit()

def btnPress3\_clicked(self):

# 下载图片

I = self.picture

keyword = I

# keyword = '\*\*\*' # 关键词, 改为你想输入的词即可, 相当于在百度图片里搜索一样

url\_init\_first = r'http://image.baidu.com/search/flip?tn=baiduimage&ipn=r&ct=201326592&cl=2&lm=-1&st=-1&fm=result&fr=&sf=1&fmq=1497491098685\_R&pv=&ic=0&nc=1&z=&se=1&showtab=0&fb=0&width=&height=&face=0&istype=2&ie=utf-8&ctd=1497491098685%5E00\_1519X735&word='

url\_init = url\_init\_first + urllib.parse.quote(keyword, safe='/')

all\_pic\_urls = []

onepage\_urls, fanye\_url = self.get\_onepage\_urls(url\_init)

all\_pic\_urls.extend(onepage\_urls)

fanye\_count = 0 # 累计翻页数

while 1:

onepage\_urls, fanye\_url = self.get\_onepage\_urls(fanye\_url)

fanye\_count += 1

# print('第页' % str(fanye\_count))

if fanye\_url == '' and onepage\_urls == []:

break

all\_pic\_urls.extend(onepage\_urls)

self.down\_pic(list(set(all\_pic\_urls)))

def btnPress2\_clicked(self):

sys.exit()

def btnPress1\_clicked(self):

info\_list = []

city\_dic = {"昆明": "KMM", "成都": "CDW", "贵阳": "GIW", "上海": "SHH", "天津": "TJP", "长沙": "CSQ", "福州": "FZS",

"广州": "GZQ", "哈尔滨": "HBB", "合肥": "HFH", "呼和浩特": "HHC", "海口": "VUQ", "杭州": "HZH", "济南": "JNK",

"兰州": "LZJ", "拉萨": "LSO", "南昌": "NCG", "南京": "NJH", "南宁": "NCG", "石家庄": "SJP", "沈阳": "SYT",

"太原": "TYV", "武汉": "WHN", "乌鲁木齐": "WMR", "西安": "XAY", "西宁": "XNO", "银川": "YIJ", "郑州": "ZZF",

"重庆": "CQW"}

test = []

# 将下拉选择框的信息接收

train\_type = self.train\_type

start\_place = self.start\_place

end\_place = self.end\_place

date = self.date

news\_place = city\_dic[start\_place]

newe\_place = city\_dic[end\_place]

I = train\_type

if (I != "G" and I != "K"):

self.textEdit.setPlainText("无法查找")

sys.exit()

raw1 = "https://kyfw.12306.cn/otn/leftTicket/query?leftTicketDTO.train\_date=" + date

raw2 = "&leftTicketDTO.from\_station=" + news\_place

raw3 = "&leftTicketDTO.to\_station=" + newe\_place

raw4 = "&purpose\_codes=ADULT"

url = raw1 + raw2 + raw3 + raw4

r = requests.get(url)

html = r.text

json\_dict = json.loads(html)

for count in range(len(json\_dict["data"]["result"])):

time = json\_dict["data"]["result"][count].split("|")

if (I == "G"):

info\_list.append(time[3])

for info in range(8, 11):

info\_list.append(time[info])

pass

# info\_list.append(time[25])

for i in range(31, 29, -1):

info\_list.append(time[i])

pass

test.append(info\_list)

info\_list = []

pass

elif (I == "K"):

info\_list.append(time[3]) # 1

for info in range(8, 11): # 3

info\_list.append(time[info])

pass

info\_list.append(time[23]) # 1

for i in range(25, 30): # 5

info\_list.append(time[i])

pass

test.append(info\_list)

info\_list = []

pass

pass

if (I == "G"):

tplt = "{:6}\t{:4}\t{:4}\t{:4}\t{:4}\t{:4}"

s1 = tplt.format("车次", "出发时间", "到达时间", "历时", "一等座", "二等座") + "\n"

for i in range(len(test)):

# print(infom[0])

infom = test[i]

for i in range(4, 6):

if (infom[i] == ""):

infom[i] = "--"

if (I in infom[0]):

temp1 = tplt.format(infom[0], infom[1], infom[2], infom[3], infom[4], infom[5])

s1 = s1 + temp1 + "\n"

pass

else:

continue

self.textEdit.setPlainText(s1)

elif (I == "K"):

tplt2 = "{:6}\t{:4}\t{:4}\t{:4}\t{:2}\t{:2}\t{:2}\t{:2}\t{:2}\t{:2}"

s2 = tplt2.format("车次", "出发时间", "到达时间", "历时", "软卧", "动卧", "硬卧", "软座", "硬座", "无座") + "\n"

for i in range(len(test)):

# print(infom[0])

infom = test[i]

for i in range(4, 7):

if (infom[i] == ""):

infom[i] = "--"

if (I in infom[0]):

temp2 = tplt2.format(infom[0], infom[1], infom[2], infom[3], infom[4], infom[5], infom[6], infom[7],

infom[8], infom[9])

s2 = s2 + temp2 + "\n"

pass

else:

continue

pass

self.textEdit.setPlainText(s2)

if \_\_name\_\_ == '\_\_main\_\_':

app = QApplication(sys.argv)

# 界面

ex = Example()

sys.exit(app.exec\_())