# **功能说明文档**

[环境需求 2](#_Toc9047)

[实例 2](#_Toc7993)

[获取天气数据并输出图表 2](#_Toc25567)

[API 4](#_Toc24126)

[getSource 4](#_Toc671)

[Source类 4](#_Toc21934)

[Urldata() 4](#_Toc27609)

[gzipSource() 4](#_Toc19333)

[useSelenium() 4](#_Toc5482)

[analyzeSource 4](#_Toc20145)

[getUrllist() 4](#_Toc10473)

[getTitlelist() 4](#_Toc26438)

[StandardOut 4](#_Toc26511)

[downloadImg() 4](#_Toc3401)

[writeExcel 5](#_Toc24140)

[Config 5](#_Toc17119)

[CONSTANT类 5](#_Toc29638)

[writeLog() 5](#_Toc21727)

[异常处理 5](#_Toc26389)

[STARTError 5](#_Toc7457)

[IOError 5](#_Toc19818)

[VALUEError 5](#_Toc15347)

[SOCKETError 5](#_Toc19990)

[UNICODEError 5](#_Toc10918)

[URLError 5](#_Toc6014)

[总结 5](#_Toc22286)

# 一、环境需求

操作系统：Windows7/10、Android4.5以上

语言环境：Python、Java

Python库：1、BeautifulSoup 3.8 2、linecache 3、xlwrite 1.0.2

1. Tesseract 0.1.2 5、selenium 2.0.1

# 二、实例

### 获取天气数据并输出图表

**·**用于显示中文

# -\*- coding: utf-8 -\*- #

**·**引用StandarOut模块

**from** spider.StandardOut **import**\*

**·**创建Source类

url='http://www.syuntravel.cn/PassengerTicket/SearchHistoryTicket'  
Source=Source(url)

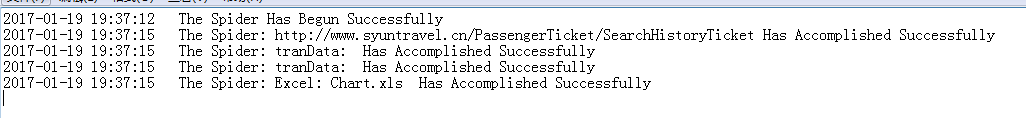
**·**启用Urldata（）方法，返回值为Html。Way=‘POST’，POST数据为字典。

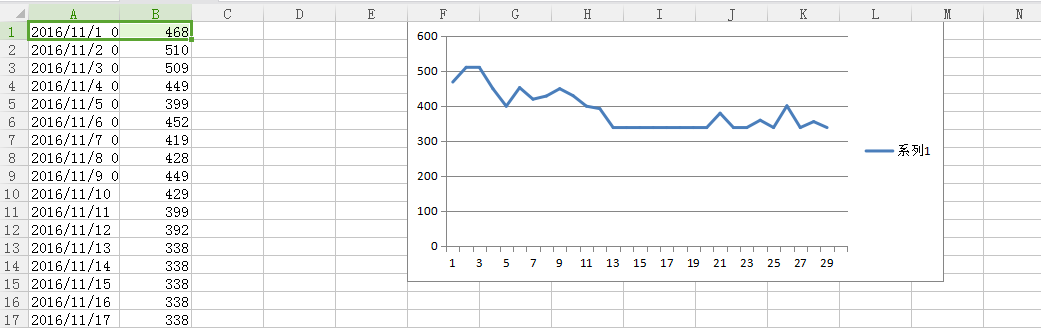
Data=Source.Urldata(Way='POST',Departure='北京

',Destination='上海',Date='2016-11')

**·**C.ExcelName为常量在config.py中定义。  
writeExcel(C.ExcelName,Data)

**·**输出日志文件log.txt、Chart..xls





## 三、API

### 1、getSource

Source类：

初始化log文件，接受Url参数

**def** \_\_init\_\_(self,Url):  
 self.Url=Url  
 writeLog(C.STARTSuccess,Way='w')

Source.Urldata()：

参数：Way：POST/GET 网页请求方法

SQL：True/False 是否启用SQL

\*\*kwargs：传入POST数据字典

用途：主函数，获取网页源码

返回值：Html文本

**def Urldata**(self,Way='GET',SQL=False,\*\*kwargs):

Source.gzipSource()：

参数：self.Url

用途：辅助函数，应对于需要gzip解码的网页

返回值：Html文本

**def gzipSource**(self):

Source.useSelenium()：

参数：self.Url

用途：辅助函数，应对于需要加载jsscript网页

返回值：Html文本

**def useSelenium**(self):

### 2、analyzeSource

getUrllist()：

参数：Data（Html文本）、Way=‘link’、‘img’

用途：分析网页获取img/link 链接

返回值：字符列表（网页）

**def getUrllist**(Data,Way='link'/’img’):

getTitlelist()：

参数：Data（Html文本）、Way=‘link’、‘img’

用途：分析网页获取img/link标题文本

返回值：字符列表（字符串）

**def getTitlelist**(Data,Way='link'/’img’):

### 3、StandardOutx

downloadImg():

参数：Imgnum（文件数字名’Imgnum.jpg’）、Urllist（链接列表）、

Thread=True/False（是否启用多线程下载）

用途：下载网页指定的图片，并保存在C.Dirname（config中定义的常量）中

返回值：多个图片(jpg/png/gif)文件

**def** downloadImg(Imgnum,Urllist,Thread=False):

writeExcel:

参数：Filename（输出Excel文件名）、Data（Html文本）、

Thread=True/False（是否启用多线程下载）

用途：获取网页指定的数据，绘制图表并保存在Filename.xls中

返回值：xls文件

**def** writeExcel(Filename,Data,Num\_row=0,Num\_col=0):

### 4、Config

CONSTANT类：

用途：定义软件中各个常量（伪造请求的头、输出文件名、错误代码）

Headers={'User-Agent' : 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36'}  
TimeLimit=3  
ExcelName='Chart.xls'  
STARTError=000  
IOError=001  
INDEXError=002

writeLog()：

参数：LogNum（错误类型）、Info（log消息）

用途：接受错误类型，记录软件运行状态。

返回值：log.txt文本

**def writeLog**(LogNum,Info=' '):

### 5、异常处理

STARTError：

抛出函数启动/接受错误，记录于log.txt中

'The Spider: %s Is Failed to complete '%Info+'\n'

IOError：

抛出文件/数据库输入/输出错误，记录于log.txt中

File.write('The %s Failed to Input '%Info+'\n')

VALUEError：

抛出传值/数据类型匹配错误，记录于log.txt中

File.write('The VALUE:%s Failed to Save '%Info+'\n')

SOCKETError：

抛出Socket请求超时错误，记录于log.txt中

File.write('The URL:%s Failed to Required ,OVERTIME '%Info+'\n')

UNICODEError：

抛出网页编码传值错误，记录于log.txt中

File.write('The URL:%s Failed to Code '%Info+'\n')

URLError：

抛出域名错误，记录于log.txt中

File.write('The URL:%s Failed to open '%Info+'\n')

## 总结

## 1、目前存在的问题

1爬虫相对性能低下，没有相应算法提升性能

2伪造功能不完善

3应用的数据库sqllite低效，无法处理大量数据

4不支持命令行操作

5验证码功能还未实现

6由于GIL锁存在,python里一个进程永远只能同时执行一个线程，浪费多核CPU的资源，没有采用多进程编程.

7 UI功能不完善，不可定制，选项较少

8代理IP,Tor匿名网络没有启用

9移植性较差

## 2、希望的改进

1能够移植到linux,android，IOS等其他系统上

2减少对于python的依赖，其他语言同样能实现

3确定需求，构建大的框架，做出接口，而不是具体方法的实现

4减少耦合度，代码易读，易于升级，添加新功能

5智能训练Tesseract，使其验证正确率提升

6软件能够帮助用户获取指定的信息，如：网页上各个商品的价格，图片

7利用os库能够支持命令行操作