

第 15 关、复习

1、爬虫总复习

(1) 爬虫四步

获取数据（包含请求和响应两个动作）、解析数据、提取数据、存储数据。



(2) 最简请求方式：requests.get()

```
1 import requests
2
3 url = ''
4 response = requests.get(url)
```



1-1、工具

1-1-1、Network

Network 能够记录浏览器的所有请求。我们最常用的是：ALL（查看全部）/XHR（仅查看XHR）/Doc（Document，第0个请求一般在这里），有时候也会看看：Img（仅查看图片）/Media（仅查看媒体文件）/Other（其他）。最后，JS 和 CSS，则是前端代码，负责发起请求和页面实现；Font 是文字的字体；而理解 WS 和 Manifest，需要网络编程的知识，倘若不是专门这个，我们不需要了解。

ALL	查看全部
XHR	一种不借助刷新网页即可传输数据的对象
Doc	Document，第0个请求一般在这里
Img	仅查看图片
Media	仅查看媒体文件
Other	其他
JS和CSS	前端代码，负责发起请求和页面实现
Font	字体
WS和Manifest	网络编程相关知识，无需了解

by 风变编程

1-1-2、XHR和Doc

我们能在 Doc 里找到一个网页的源代码，而在网页源代码里找不到的信息，通常都能在 XHR 里找到，XHR 帮我们实现了异步请求。



1-2、解析与提取（一）

BeautifulSoup，它能提供一套完整的数据解析、数据提取解决方案。



当 response.text 自动解码出问题，可使用 response.encoding=" " 来对编码进行修改。

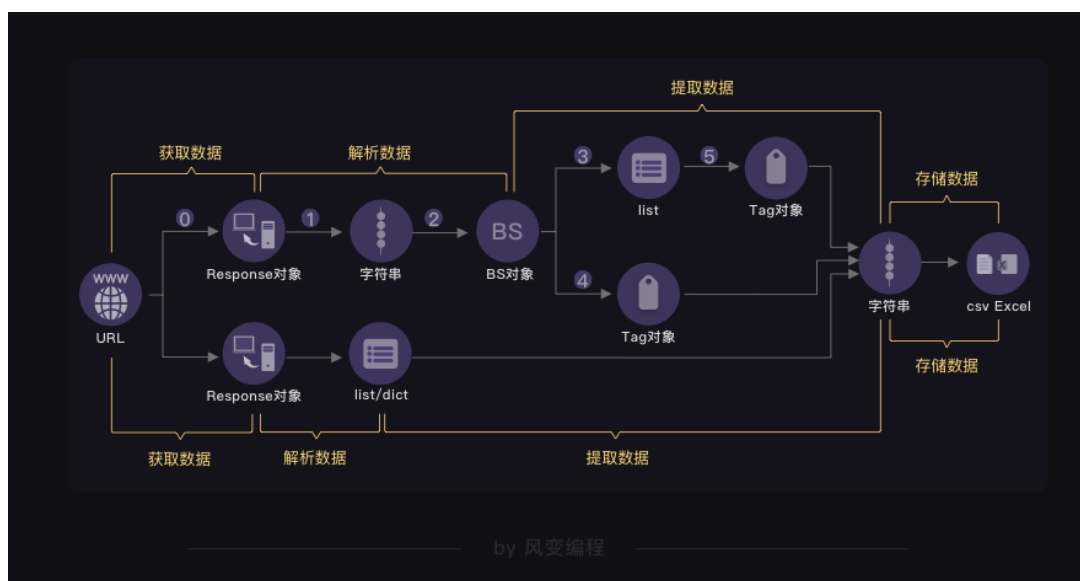
1-3、解析与提取（二）

XHR 所传输的数据，最重要的一种是用 json 格式写成的，和 html 一样，这种数据能够有组织地存储大量内容。json 的数据类型是“文本”，在 Python 语言当中，我们把它称为字符串。我们能够非常轻易地将 json 格式的数据转化为列表/字典，也能将字典/列表转为 json 格式的数据。

1-3-1、解析 json 数据



1-3-2、提取数据



1-4、更厉害的请求

1-4-1、params

可以让我们带着参数来请求数据：我想要第几页？我想要搜索的关键词？我想要多少个数据？

1-4-2、headers

请求头。它告诉服务器，我的设备/浏览器是什么？我从哪个页面而来？

1-4-3、post

post 区别于 get 的是：get 是明文显示参数，post 是非明文显示参数。

1-4-4、cookies

cookies，的作用是让服务器“记住你”，当下一次，浏览器带着cookies访问博客，服务器会知道你何人，你不需要再重复输入账号密码，就能直接访问。

1-5、存储

最常见两种存储数据的方法：csv 和 excel。



1-5-1、csv

(1) csv 写入步骤



```
1 #csv写入的代码:
2
3 import csv
4 csv_file=open('demo.csv','w',newline='')
5 writer = csv.writer(csv_file)
6 writer.writerow(['电影','豆瓣评分'])
7 csv_file.close()
```

(2) csv 读取步骤

csv读取的步骤

0 打开文件
调用open()函数

1 创建对象
借助reader()函数

2 读取内容
遍历reader对象

3 打印内容
print()

by 风变编程

```
1 #csv读取的代码:
2
3 import csv
4 csv_file=open('demo.csv','r',newline='')
5 reader=csv.reader(csv_file)
6 for row in reader:
7     print(row)
```

1-5-2、Excel

(1) Excel 写入步骤

Excel文件写入的步骤

0 创建工作簿
利用openpyxl.Workbook()创建工作book对象

1 获取工作表
借助workbook对象的active属性

2 操作单元格
单元格: sheet['A1']; 一行: append()

3 保存工作簿
save()

by 风变编程

```
1 #Excel写入的代码:
2
3 import openpyxl
4 wb=openpyxl.Workbook()
5 sheet=wb.active
6 sheet.title='new title'
7 sheet['A1'] = '漫威宇宙'
8 rows= [['美国队长','钢铁侠','蜘蛛侠','雷神'], ['是','漫威','宇宙','经典','人物']]
9 for i in rows:
10     sheet.append(i)
```

```
11 print(rows)
12 wb.save('Marvel.xlsx')
```

(2) Excel 读取步骤

Excel文件读取的步骤

0 打开工作簿

利用openpyxl.load_workbook()
创建工作簿对象

1 获取工作表

workbook对象的键，wb['sheet']

2 读取单元格

借助单元格value属性，sheet['A1'].value

3 打印单元格

print()

by 风变编程

```
1 #Excel读取的代码:
2
3 import openpyxl
4 wb = openpyxl.load_workbook('Marvel.xlsx')
5 sheet=wb['new title']
6 sheetname = wb.sheetnames
7 print(sheetname)
8 A1_value=sheet['A1'].value
9 print(A1_value)
```

1-6、更多的爬虫

多协程，是一种非抢占式的异步方式。使用多协程能让多个爬取任务用异步的方式交替执行。

用gevent实现多协程爬取的重点

0 定义爬取函数

1 用gevent.spawn()创建任务

2 用gevent.joinall()执行任务

by 风变编程

用queue模块的重点

- 0 用queue()创建队列
- 1 用put_nowait()存储数据
- 2 用get_nowait()提取数据

by 风变编程

queue对象的方法

put_nowait()	往队列里存储数据
get_nowait()	从队列里提取数据
empty()	判断队列是否为空
full()	判断队列是否为满
qsize()	判断队列还剩多少数量

by 风变编程

示例代码：

```
1 import gevent,time,requests
2 from gevent.queue import Queue
3 from gevent import monkey
4 monkey.patch_all()
5
6 start = time.time()
7
8 url_list = ['https://www.baidu.com/',
9 'https://www.sina.com.cn/',
10 'http://www.sohu.com/',
11 'https://www.qq.com/',
12 'https://www.163.com/',
13 'http://www.iqiyi.com/',
14 'https://www.tmall.com/',
15 'http://www.ifeng.com/']
16
17 work = Queue()
18 for url in url_list:
19     work.put_nowait(url)
20
21 def crawler():
```



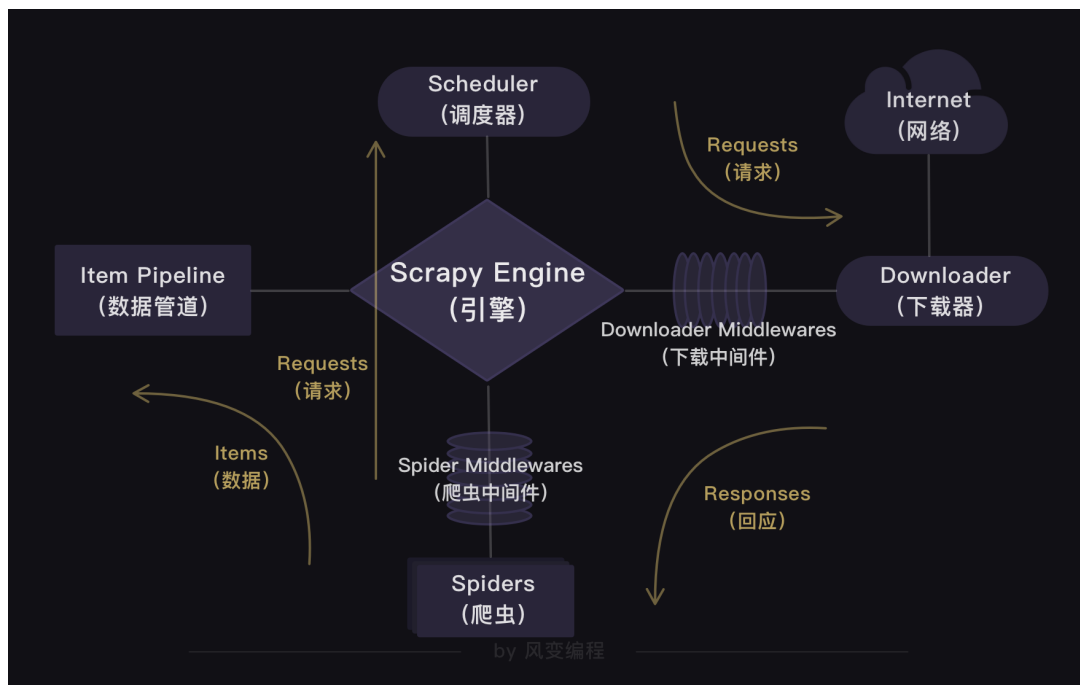
```

22     while not work.empty():
23         url = work.get_nowait()
24         r = requests.get(url)
25         print(url,work.qsize(),r.status_code)
26
27 tasks_list = [ ]
28
29 for x in range(2):
30     task = gevent.spawn(crawler)
31     tasks_list.append(task)
32 gevent.joinall(tasks_list)
33
34 end = time.time()
35 print(end-start)

```

1-7、更强大的爬虫 — 框架

1-7-1、Scrapy 结构



1-7-2、Scrapy 工作原理



1-7-3、Scrapy 用法

Scrapy的用法

- 0 创建Scrapy项目
- 1 定义item(数据)
- 2 创建和编写spiders文件
- 3 修改settings.py文件
- 4 运行Scrapy爬虫

by 风变编程

1-8、给爬虫加上翅膀

三个有力工具：selenium，邮件通知和定时。

1-8-1、selenium

(1) selenium 提取数据

Selenium提取数据的方法

方法	作用
find_element_by_tag_name	通过元素的标签名称选择
find_element_by_class_name	通过元素的class属性选择
find_element_by_id	通过元素的id选择
find_element_by_name	通过元素的name属性选择
find_element_by_link_text	通过链接文本获取超链接
find_element_by_partial_link_text	通过链接的部分文本获取超链接

by 风变编程

(2) 对象转换



(3) 搭配 BeautifulSoup 解析提取数据

前提是先获取字符串格式的网页源代码:

```
1 HTML源代码字符串 = driver.page_source
```

(4) 自动操作浏览器的方法

Selenium操作元素的常用方法	
方法	作用
.clear()	清除元素的内容
.send_keys()	模拟按键输入, 自动填写表单
.click()	点击元素

by 风变编程

1-8-2、邮件通知



所需模块：smtplib 和 email。前者负责连接服务器、登录、发送和退出的流程；后者负责填输邮件的标题与正文。



1-8-3、定时

schedule 模块：

Usage

```
$ pip install schedule
```

```
import schedule
import time

def job():
    print("I'm working...")

schedule.every(10).minutes.do(job)
schedule.every().hour.do(job)
schedule.every().day.at("10:30").do(job)
schedule.every().monday.do(job)
schedule.every().wednesday.at("13:15").do(job)

while True:
    schedule.run_pending()
    time.sleep(1)
```

by 风变编程

示例代码：

```
1 import schedule
2 import time
3 #引入schedule和time
4
5 def job():
6     print("I'm working...")
7 #定义一个叫job的函数，函数的功能是打印'I'm working...'
8
9 schedule.every(10).minutes.do(job)           #部署每10分钟执行一次job()函数的任务
10 schedule.every().hour.do(job)                #部署每x小时执行一次job()函数的任务
```

```
11 schedule.every().day.at("10:30").do(job) #部署在每天的10:30执行job()函数的任
    务
12 schedule.every().monday.do(job)          #部署每个星期一执行job()函数的任务
13 schedule.every().wednesday.at("13:15").do(job)#部署每周三的13: 15执行函数的任
    务
14
15 while True:
16     schedule.run_pending()
17     time.sleep(1)
18 #13-15都是检查部署的情况，如果任务准备就绪，就开始执行任务。
```