王伟超

wangweichao@tedu.cn

Spider-Day01笔记

网络爬虫概述

```
【1】定义
    1.1) 网络蜘蛛、网络机器人，抓取网络数据的程序
    1.2) 其实就是用Python程序模仿人点击浏览器并访问网站，而且模仿的越逼真越好

【2】爬取数据的目的
    2.1) 公司项目的测试数据，公司业务所需数据
    2.2) 获取大量数据，用来做数据分析

【3】企业获取数据方式
    3.1) 公司自有数据
    3.2) 第三方数据平台购买(数据堂、贵阳大数据交易所)
    3.3) 爬虫爬取数据

【4】Python做爬虫优势
    4.1) Python ：请求模块、解析模块丰富成熟,强大的Scrapy网络爬虫框架
    4.2) PHP ：对多线程、异步支持不太好
    4.3) JAVA：代码笨重,代码量大
    4.4) C/C++：虽然效率高,但是代码成型慢

【5】爬虫分类
    5.1) 通用网络爬虫(搜索引擎使用,遵守robots协议)
        robots协议：网站通过robots协议告诉搜索引擎哪些页面可以抓取,哪些页面不能抓取，通用网络爬虫需要遵守robots协议（君子协议）
        示例：https://www.baidu.com/robots.txt
    5.2) 聚焦网络爬虫：自己写的爬虫程序

【6】爬取数据步骤
    6.1) 确定需要爬取的URL地址
    6.2) 由请求模块向URL地址发出请求,并得到网站的响应
    6.3) 从响应内容中提取所需数据
        a> 所需数据,保存
        b> 页面中有其他需要继续跟进的URL地址,继续第2步去发请求，如此循环
```

爬虫请求模块

requests模块

- **安装**

```
【1】Linux
    sudo pip3 install requests

【2】Windows
    方法1>  cmd命令行 -> python -m pip install requests
    方法2>  右键管理员进入cmd命令行 : pip install requests
```

pip3 freeze|grep -i requests

常用方法

- **requests.get()**

get（）方法得到的是响应对象

```
【1】作用
    向目标网站发起请求,并获取响应对象

【2】参数
    2.1> url : 需要抓取的URL地址
    2.2> headers : 请求头
    2.3> timeout : 超时时间，超过时间会抛出异常
```

- **此生第一个爬虫**

```
"""
打开浏览器，输入京东地址(https://www.jd.com/)，得到响应内容
"""
import requests

res = requests.get('https://www.jd.com/')
html = res.text
print(html)
```

- **响应对象（res）属性**

```
【1】text        : 字符串
【2】content     : 字节流
【3】status_code : HTTP响应码
【4】url         : 实际数据的URL地址
```

字节串-图片、音频、视频、文件...

- **重大问题思考**

网站如何来判定是人类正常访问还是爬虫程序访问？--检查请求头！！！

```python
# 请求头（headers）中的 User-Agent
# 测试案例：向测试网站http://httpbin.org/get发请求，查看请求头(User-Agent)
import requests

url = 'http://httpbin.org/get'
res = requests.get(url=url)
html = res.text
print(html)
# 请求头中:User-Agent为-> python-requests/2.22.0 那第一个被网站干掉的是谁？？？我们是不是需要发送请求时重构一下User-Agent？？？添加 headers 参数！！！
```

- **重大问题解决**

```python
"""
包装好请求头后,向测试网站发请求,并验证
养成好习惯，发送请求携带请求头，重构User-Agent
"""
import requests

url = 'http://httpbin.org/get'
headers = {'User-Agent':'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/14.0.835.163 Safari/535.1'}
html = requests.get(url=url,headers=headers).text
print(html)
```

爬虫编码模块

- **urllib.parse模块**

```
1、标准库模块：urllib.parse
2、导入方式：
import urllib.parse
from urllib import parse
```

- **作用**

```
给URL地址中查询参数进行编码

# 示例
编码前：https://www.baidu.com/s?wd=美女
编码后：https://www.baidu.com/s?wd=%E7%BE%8E%E5%A5%B3
```

常用方法

urlencode({ 参数为字典 })

- **作用**

```
给URL地址中查询参数进行编码，参数类型为字典
```

- **使用方法**

```
# 1、URL地址中 一 个查询参数
编码前: params = {'wd':'美女'}
编码中: params = urllib.parse.urlencode(params)
编码后: params结果:  'wd=%E7%BE%8E%E5%A5%B3'

# 2、URL地址中 多 个查询参数
编码前: params = {'wd':'美女','pn':'50'}
编码中: params = urllib.parse.urlencode(params)
编码后: params结果: 'wd=%E7%BE%8E%E5%A5%B3&pn=50'
发现编码后会自动对多个查询参数间添加 & 符号
```

- **拼接URL地址的三种方式**

```
# url = 'http://www.baidu.com/s?'
# params = {'wd':'赵丽颖'}
# 问题: 请拼接出完整的URL地址
**********************************
params = urllib.parse.urlencode(params)
【1】字符串相加
【2】字符串格式化（占位符 %s）
【3】format()方法
    'http://www.baidu.com/s?{}'.format(params)

【练习】
    进入瓜子二手车直卖网官网 - 我要买车 - 请使用3种方法拼接前20页的URL地址,从终端打印输出
    官网地址: https://www.guazi.com/langfang/
```

- **练习**

```
"""
问题：在百度中输入要搜索的内容，把响应内容保存到本地文件
编码方法使用 urlencode()
"""
import requests
from urllib import parse

# 1.拼接URL地址
word = input('请输入搜索内容:')
params = parse.urlencode({'wd':word})

url = 'http://www.baidu.com/s?{}'
url = url.format(params)

# 2.发请求获取响应内容
headers = {'User-Agent':'Mozilla/5.0'}
```

```
html = requests.get(url=url,headers=headers).content.decode('utf-8')

# 3.保存到本地文件
filename = word + '.html'
with open(filename,'w',encoding='utf-8') as f:
    f.write(html)
```

quote('参数为字符串')

- **使用方法**

```
# 对单独的字符串进行编码 - URL地址中的中文字符
word = '美女'
result = urllib.parse.quote(word)
result结果: '%E7%BE%8E%E5%A5%B3'
```

- **练习**

```
"""
问题：在百度中输入要搜索的内容，把响应内容保存到本地文件
编码方法使用 quote()
"""
import requests
from urllib import parse

# 1.拼接URL地址
word = input('请输入搜索内容:')
params = parse.quote(word)

url = 'http://www.baidu.com/s?wd={}'
url = url.format(params)

# 2.发请求获取响应内容
headers = {'User-Agent':'Mozilla/5.0'}
html = requests.get(url=url,headers=headers).content.decode('utf-8')

# 3.保存到本地文件
filename = word + '.html'
with open(filename,'w',encoding='utf-8') as f:
    f.write(html)
```

- **unquote(string)解码**

```
# 将编码后的字符串转为普通的Unicode字符串
from urllib import parse

params = '%E7%BE%8E%E5%A5%B3'
result = parse.unquote(params)

result结果: 美女
```

- **小总结**

```
【1】 什么是robots协议

【2】 requests模块使用
    res = requests.get(url=url,headers={'User-Agent':'xxx'})
    响应对象res属性:
        a> res.text
        b> res.content
        c> res.status_code
        d> res.url

【3】urllib.parse
   a> urlencode({'key1':'xxx','key2':'xxx'})
   b> quote('xxx')
   c> unquote('xxx')

【4】URL地址拼接 - urlencode()
   a> 'http://www.baidu.com/s?' + params
   b> 'http://www.baidu.com/s?%s' % params
   c> 'http://www.baidu.com/s?{}'.format(params)
```

案例 - 百度贴吧数据抓取

- **需求**

```
1、输入贴吧名称：赵丽颖吧
2、输入起始页：1
3、输入终止页：2
4、保存到本地文件：赵丽颖吧_第1页.html、赵丽颖吧_第2页.html
```

- **实现步骤**

```
【1】查看所抓数据在响应内容中是否存在
    右键 - 查看网页源码 - 搜索关键字

【2】查找并分析URL地址规律
    第1页：http://tieba.baidu.com/f?kw=???&pn=0
    第2页：http://tieba.baidu.com/f?kw=???&pn=50
    第n页：pn=(n-1)*50

【3】发请求获取响应内容

【4】保存到本地文件
```

- **代码实现**

```
import requests
from urllib import parse
import time
import random
```

```python

class BaiduSpider(object):
    def __init__(self):
        self.url='http://tieba.baidu.com/f?kw={}&pn={}'
        self.headers = { 'User-Agent':'Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; InfoPath.3)' }

    def get_html(self,url):
        """获取响应内容html"""
        html = requests.get(url=url,headers=self.headers).content.decode('utf-8')

        return html

    def parse_html(self):
        """解析提取数据"""
        pass

    def save_html(self,filename,html):
        """处理数据"""
        with open(filename,'w',encoding='utf-8') as f:
            f.write(html)

    def run(self):
        """入口函数"""
        name = input('请输入贴吧名:')
        beign_page = int(input('请输入起始页:'))
        end_page = int(input('请输入终止页:'))
        # 对name进行编码
        params = parse.quote(name)
        for page in range(beign_page,end_page+1):
            # 拼接URL地址
            pn = (page-1)*50
            url = self.url.format(params,pn)
            # 请求+保存
            html = self.get_html(url)
            filename = '{}_第{}页.html'.format(name,page)
            self.save_html(filename,html)
            # 控制爬取频率:uniform(0,1)生成0-1之间浮点数
            time.sleep(random.randint(1,2))
            # time.sleep(random.uniform(0,1))
            print('第%d页抓取完成' % page)

if __name__ == '__main__':
    spider = BaiduSpider()
    spider.run()
```

正则解析模块re

re模块使用流程

```python
# 方法一
r_list=re.findall('正则表达式',html,re.S)

# 方法二
pattern = re.compile('正则表达式',re.S)
r_list = pattern.findall(html)
```

正则表达式元字符

元字符	含义
.	任意一个字符（不包括\n）
\d	一个数字
\s	空白字符
\S	非空白字符
[]	包含[]内容
*	出现0次或多次
+	出现1次或多次

- **思考 - 请写出匹配任意一个字符的正则表达式?**

```python
import re
# 方法一
pattern = re.compile('[\s\S]')
result = pattern.findall(html)

# 方法二
pattern = re.compile('.',re.S)
result = pattern.findall(html)
```

贪婪匹配和非贪婪匹配

- **贪婪匹配(默认)**

```
1、在整个表达式匹配成功的前提下,尽可能多的匹配 * + ?
2、表示方式：.* .+ .?
```

- **非贪婪匹配**

```
1、在整个表达式匹配成功的前提下,尽可能少的匹配 * + ?
2、表示方式：.*? .+? .??
```

- **代码示例**

```python
import re

html = '''
<div><p>九霄龙吟惊天变</p></div>
<div><p>风云际会潜水游</p></div>
'''
# 贪婪匹配
p = re.compile('<div><p>.*</p></div>',re.S)
r_list = p.findall(html)
print(r_list)

# 非贪婪匹配
p = re.compile('<div><p>.*?</p></div>',re.S)
r_list = p.findall(html)
print(r_list)
```

正则表达式分组

作用

```
在完整的模式中定义子模式，将每个圆括号中子模式匹配出来的结果提取出来
```

示例代码

```python
import re

s = 'A B C D'
p1 = re.compile('\w+\s+\w+')
print(p1.findall(s))
# 分析结果是什么？？？
# ['A B','C D']

p2 = re.compile('(\w+)\s+\w+')
print(p2.findall(s))
# 第1步：['A B','C D']
# 第2步：['A','C']

p3 = re.compile('(\w+)\s+(\w+)')
print(p3.findall(s))
# 第1步：['A B','C D']
# 第2步：[('A','B'),('C','D')]
```

分组总结

```
1、在网页中，想要什么内容，就加()
2、先按整体正则匹配，然后再提取分组()中的内容
   如果有2个及以上分组()，则结果中以元组形式显示 [(),(),()]
```

课堂练习

```
# 从如下html代码结构中完成如下内容信息的提取:
问题1 : [('Tiger',' Two...'),('Rabbit','Small..')]
问题2 :
    动物名称 : Tiger
    动物描述 : Two tigers two tigers run fast
    ********************************************
    动物名称 : Rabbit
    动物描述 : Small white rabbit white and white
```

页面结构如下

```
<div class="animal">
    <p class="name">
            <a title="Tiger"></a>
    </p>
    <p class="content">
            Two tigers two tigers run fast
    </p>
</div>

<div class="animal">
    <p class="name">
            <a title="Rabbit"></a>
    </p>

    <p class="content">
            Small white rabbit white and white
    </p>
</div>
```

练习答案

```
import re

html = '''<div class="animal">
    <p class="name">
        <a title="Tiger"></a>
    </p>

    <p class="content">
        Two tigers two tigers run fast
    </p>
</div>

<div class="animal">
    <p class="name">
        <a title="Rabbit"></a>
    </p>

    <p class="content">
        Small white rabbit white and white
    </p>
</div>'''

```

```
p = re.compile('<div class="animal">.*?title="(.*?)".*?content">(.*?)</p>.*?</div>',re.S)
r_list = p.findall(html)

for rt in r_list:
    print('动物名称:',rt[0].strip())
    print('动物描述:',rt[1].strip())
    print('*' * 50)
```

猫眼电影top100抓取案例

- **爬虫需求**

```
【1】确定URL地址
    百度搜索 - 猫眼电影 - 榜单 - top100榜

【2】 爬取目标
    所有电影的 电影名称、主演、上映时间
```

- **爬虫实现**

```
【1】查看网页源码，确认数据来源
    响应内容中存在所需抓取数据 - 电影名称、主演、上映时间

【2】翻页寻找URL地址规律
    第1页：https://maoyan.com/board/4?offset=0
    第2页：https://maoyan.com/board/4?offset=10
    第n页：offset=(n-1)*10

【3】编写正则表达式
    <div class="movie-item-info">.*?title="(.*?)".*?class="star">(.*?)</p>.*?releasetime">(.*?)</p>

【4】开干吧兄弟
```

- **代码实现**

```

import requests
import re
import time
import random


class MaoyanSpider(object):
    def __init__(self):
        self.url = 'https://maoyan.com/board/4?offset={}'
        self.headers = {
            'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/14.0.835.163 Safari/535.1'}
        self.i = 0

```

```
    def get_html(self, url):
        html = requests.get(url=url, headers=self.headers).text
        # 直接调用解析函数
        self.parse_html(html)

    def parse_html(self, html):
        """正则解析函数"""
        regex = '<div class="movie-item-info">.*?title="(.*?)".*?class="star">(.*?)</p>.*?releasetime">(.*?)</p>'
        pattern = re.compile(regex, re.S)
        # dd_list: [(),(),()]
        dd_list = pattern.findall(html)
        self.save_html(dd_list)

    def save_html(self, dd_list):
        """保存数据函数"""
        item = {}
        for dd in dd_list:
            item['name'] = dd[0].strip()
            item['star'] = dd[1].strip()[3:]
            item['time'] = dd[2].strip()[5:15]
            print(item)
            self.i += 1

    def run(self):
        for offset in range(0, 91, 10):
            url = self.url.format(offset)
            self.get_html(url)
            time.sleep(random.randint(1, 3))
        print('电影数量: ', self.i)


if __name__ == '__main__':
    start_time = time.time()
    spider = MaoyanSpider()
    spider.run()
    end_time = time.time()
    print('执行时间:%.2f' % (end_time - start_time))
```

今日作业

- **把百度贴吧案例重写一遍,不要参照课上代码**
- **猫眼电影案例重写一遍,不要参照课上代码**
- **复习任务**

```
pymysql、MySQL基本命令
MySQL  ：建库建表普通查询、插入、删除等
Redis ：  python和redis交互,集合基本操作
```

- **建库建表**

```
【1】使用SQL命令完成
   1.1) 创建库 maoyandb ,字符编码为 utf8
   1.2) 在maoyandb库中创建表 maoyantab, 字段要求如下:
        name 变长,宽度为100
        star 变长,宽度为500
        time 变长,宽度为100

【2】使用pymysql模块完成
   在表maoyantab中插入1条表记录（电影名称、主演、上映时间三个字段）

【3】将猫眼电影案例中所抓取的电影信息存入到MySQL数据库的maoyantab表中
```