王伟超

wangweichao@tedu.cn

SPIDER-DAY01

1. 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
```

2. 爬虫请求模块

2.1 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
```

2.2 常用方法

- **requests.get()**

```
【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**

```
"""
向京东官网（https://www.jd.com/）发请求,获取响应内容
"""
import requests

resp = requests.get(url='https://www.jd.com/')
# 1.text属性：获取响应内容-字符串
html = resp.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
```

- **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！！！

```
# 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 - headers参数**

```
"""
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
# 在html中确认User-Agent
print(html)
```

3. 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
```

3.1 urlencode()

- **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('请输入搜索关键字:')
params = parse.urlencode({'wd':word})
url = 'http://www.baidu.com/s?{}'.format(params)

# 2. 发请求获取响应内容
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

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

3.2 quote()

- **使用方法**

```
http://www.baidu.com/s?wd=赵丽颖

# 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('请输入搜索关键字:')
params = parse.quote(word)
url = 'http://www.baidu.com/s?wd={}'.format(params)

# 2. 发请求获取响应内容
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).content.decode('utf-8')

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

3.3 unquote()

```
# 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
```

4. 百度贴吧爬虫

4.1 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
```

4.2 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
```

4.3 代码实现

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

class TiebaSpider:
    def __init__(self):
        """定义常用变量"""
        self.url = 'http://tieba.baidu.com/f?kw={}&pn={}'
        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'}

    def get_html(self, url):
        """请求功能函数"""
        html = requests.get(url=url, headers=self.headers).content.decode('utf-8')
```

```
21
22          return html
23
24      def parse_html(self):
25          """解析功能函数"""
26          pass
27
28      def save_html(self, filename, html):
29          """数据处理函数"""
30          with open(filename, 'w') as f:
31              f.write(html)
32
33      def run(self):
34          """程序入口函数"""
35          name = input('请输入贴吧名:')
36          start = int(input('请输入起始页:'))
37          end = int(input('请输入终止页:'))
38          # 编码
39          params = parse.quote(name)
40          # 拼接多页的URL地址
41          for page in range(start, end + 1):
42              pn = (page - 1) * 50
43              page_url = self.url.format(params, pn)
44              # 请求 + 解析 + 数据处理
45              html = self.get_html(url=page_url)
46              filename = '{}_第{}页.html'.format(name, page)
47              self.save_html(filename, html)
48              # 终端提示
49              print('第%d页抓取完成' % page)
50              # 控制数据抓取的频率
51              time.sleep(random.randint(1, 2))
52
53  if __name__ == '__main__':
54      spider = TiebaSpider()
55      spider.run()
```

5. 正则解析模块re

5.1 使用流程

```
1  r_list=re.findall('正则表达式',html,re.S)
```

5.2 元字符

元字符	含义
.	任意一个字符（不包括\n）

元字符	含义
\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 - 匹配任意一个字符的正则表达式?**

```
r_list = re.findall('.', html, re.S)
```

5.3 贪婪与非贪婪

- **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**

```
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)
```

5.4 正则分组

- **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？？？
# 结果：['A B', 'C D']

p2 = re.compile('(\w+)\s+\w+')
print(p2.findall(s))
# 第一步：['A B', 'C D']
# 第二步：['A', 'C']

p3 = re.compile('(\w+)\s+(\w+)')
print(p3.findall(s))
# 第一步：['A B', 'C D']
# 第二步：[('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 [(),(),()]
3、最终结果有3种情况
   情况1：[]
   情况2：['', '', '']  -- 正则中1个分组时
   情况3：[(), (), ()]  -- 正则中多个分组时
```

- **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">.*?<a title="(.*?)".*?

<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)
```

6. 笔趣阁小说爬虫

6.1 项目需求

```
【1】官网地址：https://www.biqukan.cc/list/
    选择一个类别，比如：'玄幻小说'

【2】爬取目标
    '玄幻小说'类别下前20页的
    2.1》小说名称
    2.2》小说链接
    2.3》小说作者
    2.4》小说描述
```

6.2 思路流程

```
【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

【2】翻页寻找URL地址规律
    第1页：https://www.biqukan.cc/fenlei1/1.html
    第2页：https://www.biqukan.cc/fenlei1/2.html
    第n页：https://www.biqukan.cc/fenlei1/n.html

【3】编写正则表达式
    '<div class="caption">.*?<a href="(.*?)" title="(.*?)">.*?<small.*?>(.*?)</small>.*?>(.*?)
</p>'

【4】开干吧兄弟
```

6.3 代码实现

```
"""
目标：
    笔趣阁玄幻小说数据抓取
思路：
    1. 确认数据来源 - 右键 查看网页源代码,搜索关键字
    2. 确认静态,观察URL地址规律
    3. 写正则表达式
    4. 写代码
"""

import re
import requests
import time
import random

class NovelSpider:
    def __init__(self):
```

```
        self.url = 'https://www.biqukan.cc/fenlei1/{}.html'
        self.headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.193 Safari/537.36'}

    def get_html(self, url):
        html = requests.get(url=url, headers=self.headers).content.decode('gbk', 'ignore')

        self.refunc(html)

    def refunc(self, html):
        """正则解析函数"""
        regex = '<div class="caption">.*?<a href="(.*?)" title="(.*?)">.*?<small.*?>(.*?)</small>.*?>(.*?)</p>'
        novel_info_list = re.findall(regex, html, re.S)
        for one_novel_info_tuple in novel_info_list:
            item = {}
            item['title'] = one_novel_info_tuple[1].strip()
            item['href'] = one_novel_info_tuple[0].strip()
            item['author'] = one_novel_info_tuple[2].strip()
            item['comment'] = one_novel_info_tuple[3].strip()
            print(item)

    def crawl(self):
        for page in range(1, 6):
            page_url = self.url.format(page)
            self.get_html(url=page_url)
            time.sleep(random.randint(1, 2))

if __name__ == '__main__':
    spider = NovelSpider()
    spider.crawl()
```

7. MySQL数据持久化

7.1 pymysql回顾

- **MySQL建库建表**

```
create database noveldb charset utf8;
use noveldb;
create table novel_tab(
title varchar(100),
href varchar(500),
author varchar(100),
comment varchar(500)
)charset=utf8;
```

- **pymysql示例**

```python
import pymysql

db = pymysql.connect('localhost','root','123456','noveldb',charset='utf8')
cursor = db.cursor()

ins = 'insert into novel_tab values(%s,%s,%s,%s)'
novel_li = ['花千骨', 'http://zly.com', '赵丽颖', '小骨的传奇一生']
cursor.execute(ins,novel_li)

db.commit()
cursor.close()
db.close()
```

7.2 笔趣阁数据持久化

```python
"""
1. 在 __init__() 中连接数据库并创建游标对象
2. 在数据处理函数中将所抓取的数据处理成列表，使用execute()方法写入数据库
3. 数据抓取完成后关闭游标及断开数据库连接
"""
import re
import requests
import time
import random
import pymysql

class NovelSpider:
    def __init__(self):
        self.url = 'https://www.biqukan.cc/fenlei1/{}.html'
        self.headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.193 Safari/537.36'}
        # 连接数据库
        self.db = pymysql.connect('localhost', 'root', '123456', 'noveldb', charset='utf8')
        self.cur = self.db.cursor()

    def get_html(self, url):
        html = requests.get(url=url, headers=self.headers).content.decode('gbk', 'ignore')

        self.refunc(html)

    def refunc(self, html):
        """正则解析函数"""
        regex = '<div class="caption">.*?<a href="(.*?)" title="(.*?)">.*?<small.*?>(.*?)</small>.*?>(.*?)</p>'
        novel_info_list = re.findall(regex, html, re.S)
        for one_novel_info in novel_info_list:
            # 调用数据处理函数
            self.save_to_mysql(one_novel_info)

    def save_to_mysql(self, one_novel_info):
        """将数据存入MySQL数据库"""
        one_novel_li = [
```

```
            one_novel_info[1].strip(),
            one_novel_info[0].strip(),
            one_novel_info[2].strip(),
            one_novel_info[3].strip(),
        ]
        ins = 'insert into novel_tab values(%s,%s,%s,%s)'
        self.cur.execute(ins, one_novel_li)
        self.db.commit()
        # 终端打印测试
        print(one_novel_li)

    def crawl(self):
        for page in range(1, 6):
            page_url = self.url.format(page)
            self.get_html(url=page_url)
            time.sleep(random.randint(1, 2))

        # 所有数据抓取完成后断开数据库连接
        self.cur.close()
        self.db.close()

if __name__ == '__main__':
    spider = NovelSpider()
    spider.crawl()
```

8. 今日作业

```
【1】把百度贴吧案例重写一遍,不要参照课上代码
【2】笔趣阁案例重写一遍,不要参照课上代码
【3】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
```