URLLib库

- urllib库

```
1. 是内置的http库
        也是Python的爬虫基础库
2. 模块名：
        python2：urlib  urlib2
        python3：urlib.request
```

- requests库

```
1. 爬虫的入门工具，也是http请求库
        是第三方库，底层封装了urllib
        更加便于发送请求和处理响应
2. 模块名：
        requests
```

- BeautifulSoup

```
1. 强大的高效的HTTP库
        提供了多种函数用于处理导航，搜索，修改分析树等功能
        可以用于发送请求
2. 是第三方库
3. 具备xml解析功能
        可以用于解析html
        1. 支持xpath语法
        2. 支持自己的解析规则
        3. 支持html解析
        4. 支持原生的解析（python自带的解析器）
4. 模块名：
        bs4
```

- lxml---xpath语法

```
1. 是一门用于解析XML文本的语言
2. 模块：
        lxml模块---etree
        from lxml import etree
3. lxml 是解析库---支持xpath语法
        解析库区别于请求库，作用：解析XML文本
        XML：可扩展标记语言（Extensible  Mark Language）
        XML：是纯文本，用于数据传输，和实现信息内容
              是由父子节点构成
```

---

urllib库

```
1. 什么是urllib：
        内置的http请求库
2. 种类：
        Python2：
                urlib:主要用于对请求进行编码
                urlib2：主要用于发送请求
                默认编码：ASCII
        python3：
                urlib.request: 请求模块
                urlib.Error：异常处理
                urlib.parse： url解析
                默认编码：UTF-8
```

- 基本使用

```
发送请求的两种方式
1. urllib.request.urlopen(url)
        发送请求，获得响应
2. urllib.request.urlretrieve(url,fileName)
        将url相应的内容存储到fileName下
```

- get请求

```
import urllib.request  as u
import urllib.parse as parse

url='http://www.baidu.com'

# response=u.urlopen(url)
# print(response.read().decode('utf-8'))

#
u.urlretrieve('http://5b0988e595225.cdn.sohucs.com/images/20180324/1736819eb11340dd9aaf
fc7b6f92c318.jpeg',r'E:\AI145\第四阶段：爬虫\代码\AI145\com\baizhi\AI145\爬虫
\urllib\a.jpg')

# res=u.urlopen(url+'/s?wd=呵呵')
# print(res.read().decode('utf-8'))

# url中不能出现中文，wd=呵呵 非法的
# 解决方案：将呵呵转成ASCII码
s=input('请输入要查询的数据：')
result=parse.quote(s)
# print(result)
res=u.urlopen(url+'/s?wd=%s'%result)
print(res.read().decode('utf-8'))
```

- post请求

```
import urllib.request  as u
import urllib.parse as parse
url='http://www.iqianyue.com/mypost/'
postData=parse.urlencode({
                            'name':'呵呵',
                            'pass':'123'
                              })  # 用于解析url数据
postData=postData.encode('utf-8')  # 从数据---二进制数据
res=u.urlopen(url=url,data=postData)
print(res.read().decode('utf-8'))
```

URLlib的高级应用

- 设置headers

```
Headers的内容：是常用的反爬虫手段
User-Agent：  用于识别用户身份

爬虫：在请求时携带上User-Agent,其内容修改为用户数据（伪装成浏览器）---  伪装技术
```

```
import urllib.request as reqeust
import urllib.parse as parse

# 构建Headers --- 字典
headers={
'User-Agent': 'WebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98
Safari/537.36Mozilla/5.0 (Windows NT 6.1; Win64; x64) Apple'
}
# 构建新的Requset对象
req=reqeust.Request(url='http://www.dianping.com/',headers=headers)
# 将构建好的新Request对象传入
res2=reqeust.urlopen(req) # url---Handler对象---Request对象---url
with open('hehe.html','wb') as f:
    f.write(res2.read())
# 提示：为了避免编码问题，直接传入二进制数据（二进制数据不会直接考虑编码个数）
```

- 设置Proxy代理

```
1. 如果IP被某网站封禁，则无法访问该网站
2. 解决方案：更换IP
        1. 自己的IP不变
        2. 找到可用的其他IP
        3. 将其他IP设置成代理IP
                再次访问网站时，不是使用自己IP，而是使用代理的IP
```

```
import urllib.request  as request

url='http://www.taobao.com/'
# proxy={类型:IP+端口号}
```

```
proxy={'http':'180.169.186.155:1080'}  # 别人的IP  网上搜索
# 将代理IP绑定到Handler对象
handler=request.ProxyHandler(proxy)  # Handler 控制器
# 创建opener---绑定代理IP （绑定的是Handler--包含了代理IP）
opener=request.build_opener(handler)5
# 将自己创建的opener和系统默认的opener替换掉
# 方法1：
request.install_opener(opener)

# 方法2：
opener.open(url)  # 临时使用自己的opener，不应影响默认的opener


res=request.urlopen(url)  # opener---获取IP
with open('拉钩.html','wb') as f:
    f.write(res.read())
```

- 设置超时时间

```
import urllib.request as request

urls=['https://list.tmall.com/','https://www.baidu.com/']

for url in urls:
    try:
        res=request.urlopen(url,timeout=1)
        print(res.read().decode('utf-8'))
    except:
        pass

# 说明：如果某个url访问时间过长，会影响效率，可能是无效链接，可选择跳过该url，爬取别的有用链接
```

- 异常处理

```
import urllib.request as request
import urllib.error as error

# url='https://list.tmall.com/'
url='http://www.lagou.com/'
try:
    res=request.urlopen(url,timeout=1)
    print(res.read().decode('utf-8'))
except error.URLError as e:
    print(e)
```

- 设置cookies

1. 登录状态的保持
cookies 一般存储是认证信息
账号密码： 存在于浏览器中 浏览器在访问时，自动携带cookie（包含了账号密码）访问服务器

```
import urllib.request as request
import urllib.parse as parse
import http.cookiejar as cookiejar

# 设置登录的url
url='http://www.honestcareer.com/hr/dologin'
# 准备post数据
postdata=parse.urlencode({'type': '1',
                          'username': '18501279410',
                          'password': 'hbw123'}).encode('utf-8')
# 创建cookiejar
cj=cookiejar.CookieJar()
# 创建控制器
handler=request.HTTPCookieProcessor(cj)
# 创建opener
opener=request.build_opener(handler)
# 绑定全局opener
request.install_opener(opener)

# 发送登录请求---服务器会返回一个cookie  自动的给opener中绑定的cookieJar赋值
res=request.urlopen(url,postdata)
print(res.read().decode('utf-8'))
print(request.urlopen('http://www.honestcareer.com/hr/index').read().decode('utf-8'))
```

- 百度贴吧

```
# 1. 发送请求（urllib.request）
# 2. 获取相应(urllib.request)
# 3. 解析响应（正则表达式）
# 4. 数据入库 （使用文件存储）

# 目标：https://tieba.baidu.com/p/4785080470  美图的一个贴吧

import urllib.request as request
import re

# 发送请求，获得响应的内容
def getHtml(url):
    return request.urlopen(url).read().decode('utf-8')
# 解析响应，利用re,获得所有的图片的url
def getURLs(html):
    reObject=re.compile('class="BDE_Image" src="(https://.*?\.jpg)"')
    urls=reObject.findall(html)
    return urls

# 通过图片url下载图片，并保存
def saveImages(urlList):
    count=0
    for i in urlList:
        request.urlretrieve(i,r'E:\AI145\第四阶段：爬虫\代码\AI145\com\baizhi\AI145\爬虫
\urllib\picture\%s.jpg'%count)
```

```
        count+=1

if __name__ == '__main__':
    pn=1
    while 1:
        url='https://tieba.baidu.com/p/4785080470?pn=%s'%pn
        pn+=1
        saveImages(getURLs(getHtml(url)))
```