Python爬虫

1. 工具准备

➢第三方依赖库

- Requests: HTTP请求库
- BeautifulSoup: HTML解析库
- Pymongo: MongoDB的Python封装模块
- Selenium: Web自动化测试框架，用于模拟登陆和获取JS动态数据

2. HTTP请求

➢HTTP的基本概念

通常HTTP消息包括客户端向服务器的请求消息和服务器向客户端的响应消息。
这两种类型的消息由一个起始行，一个或者多个头域，一个指示头域结束的空行和可选的消息体组成。

➢HTTP概览

Request URL: 表示请求的URL
Request Method: 表示请求的方法，还有HEAD, POST,DELETE,PUT等，其中GET和POST最常用。
Status Code: 显示HTTP请求和状态码，表示HTTP请求的状态。

- 1xx: 请求已被服务器接收，继续处理。
- 2xx: 请求已被服务器接收、理解并接受。
- 3xx: 需要后续操作才能完成这一请求。
- 4xx: 请求含有词法错误或者无法被执行。
- 5xx: 服务器在处理某个正确请求时发生错误。

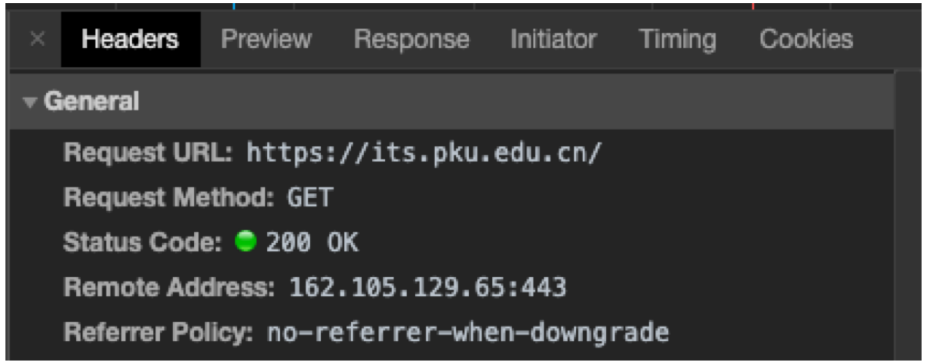

图2-1. its.pku.edu.cn的请求和响应头

3. 解析HTTP

- 使用BeautifulSoup解析处理

图3-1. 解析html实例

3. 解析HTTP

- 使用BeautifulSoup解析处理

  select函数：

  1. 通过标签名查找 soup.select(‘title’)
  2. 通过类名查找，前面加. soup.select(‘.class’)
  3. 通过id查找，前面加# soup.select(‘#id’)
  4. 组合查找 soup.select(‘title #id’) 中间有一个空格
  5. 属性查找，属性需要使用中括号括起来。

4. 使用Cookie模拟登录

- Cookie

  Cookie，可以简单认的为是在浏览器端记录包括登陆状态在内的各种属性值的容器名称，其实就是服务器为了保持浏览器与服务器之间连通状态，而在用户本地上创建的数据。只要用户再一次登陆，服务器会主动地寻找这些预存的数据，而无需再要求像第一次一样的操作。

- 使用Cookie的两种方式

  - 将Cookie写在header头部
  - 使用requests插入Cookie

    cookie = {“Cookie”: “xxxxxx”}

    html = requests.get(url, cookies=cookie)

4. 使用Cookie模拟登录

- Cookie保持登录机制

  前一次登录时，服务器发送了包含登录凭据(用户名和密码的某种加密形式)的Cookie到用户的硬盘中。再次登录时，如果Cookie尚未到期，则浏览器会发送该cookie，服务器验证凭据，于是不必输入用户名和密码就可以让用户登录了。

图4-1. 微博登录实例

5. 使用多线程爬虫

- 使用multiprocessing模块

```
from multiprocessing.dummy import Pool
results = pool.map(爬取函数，网址列表)
```

6.使用Selenium

- Selenium简介

  Selenium可以用来模拟浏览器的运行，直接运行在浏览器中，可以让浏览器自动加载页面，获取需要的数据，甚至页面截图，或者判断页面上某些动作是否发生。

7. 使用数据库存储爬虫数据

- MySQL数据库

  MySQL为关系型数据库(Relational Database Management System)，这种所谓的"关系型"可以理解为"表格"的概念，一个关系型数据库由一个或数个表格组成。

- MongoDB数据库

  MongoDB数据库是基于k-v形式保存数据的非关系型数据库，存储格式类似于python的字典。

```
MongoDB shell version v4.2.5
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("e65841e9-2997-4ace-bf56-328420569e50") }
MongoDB server version: 4.2.5
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        http://docs.mongodb.org/
Questions? Try the support group
        http://groups.google.com/group/mongodb-user
Server has startup warnings:
2020-03-29T14:13:20.329+0800 I  CONTROL  [initandlisten]
2020-03-29T14:13:20.329+0800 I  CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-03-29T14:13:20.329+0800 I  CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2020-03-29T14:13:20.329+0800 I  CONTROL  [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.
2020-03-29T14:13:20.329+0800 I  CONTROL  [initandlisten]
2020-03-29T14:13:20.329+0800 I  CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2020-03-29T14:13:20.329+0800 I  CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2020-03-29T14:13:20.329+0800 I  CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2020-03-29T14:13:20.329+0800 I  CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2020-03-29T14:13:20.329+0800 I  CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2020-03-29T14:13:20.329+0800 I  CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2020-03-29T14:13:20.329+0800 I  CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

> 1+1
2
>
```

图7-1. MongoDB成功运行实例

8. AJAX数据爬取

- AJAX介绍

  AJAX，即Asynchronous Javascript and XML，是指一种创建交互式网页应用的网页开发技术。通过在后台与服务器进行少量数据交换，AJAX可以实现网页异步更新。AJAX特殊的请求类型，xhr。

- AJAX网页特点

  1. 页面加载速度快
  2. 不刷新网页就可以更新信息
  3. 源代码内容与网页内容不同