**第三章python爬虫模拟登陆&验证码识别**

**本节所讲内容：**

**3.1、模拟登录概述**

**3.2、requests 模拟登陆**

**3.3、普通验证码识别**

**3.1、模拟登录概述**

**1、模拟登陆：**

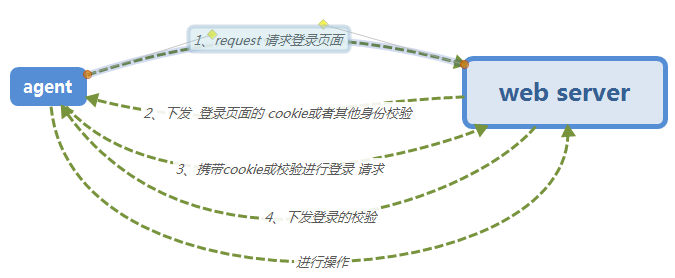
**用脚本向服务器提交登陆信息，获得登陆后的资源和权限，进行下载或者操作**

**模拟登陆的核心和我们编写网站登陆功能的核心是一体的都是对cookie和用户身份的校验和可持续访问的操作**

**最基本的模拟登陆的思路**

****

**最正常的模拟登陆思路**

****

**常见的防止模拟登陆的方法**

1. **加入hidden标签 csdn类型网站**

**爬取登陆页面，匹配hidden标签的值，然后携带提交**

1. **加入js加密，hidden提交的东西是js动态加密的，数据动态加载，qq，csdn**
2. **强大的js分析能力，分析算法，提交给python重写，来获取值**
3. **Selenium 驱动js，我们通常使用phanomjs来不打开网页驱动js登录**

**3、验证码**

**1、使用Opencv(图像处理的库)+PIL+numpy 进行图像识别**

**2、弹出验证码，保存验码，人眼识别，提交验证码**

**4、定时发送响应，隐藏登录接口**

**模拟登陆的目标：我爱我家，用户名 17611100746，密码 \*\*\*\*\*\*\***

* 1. **requests 模拟登陆**

**3.2.1 模拟登陆分析**

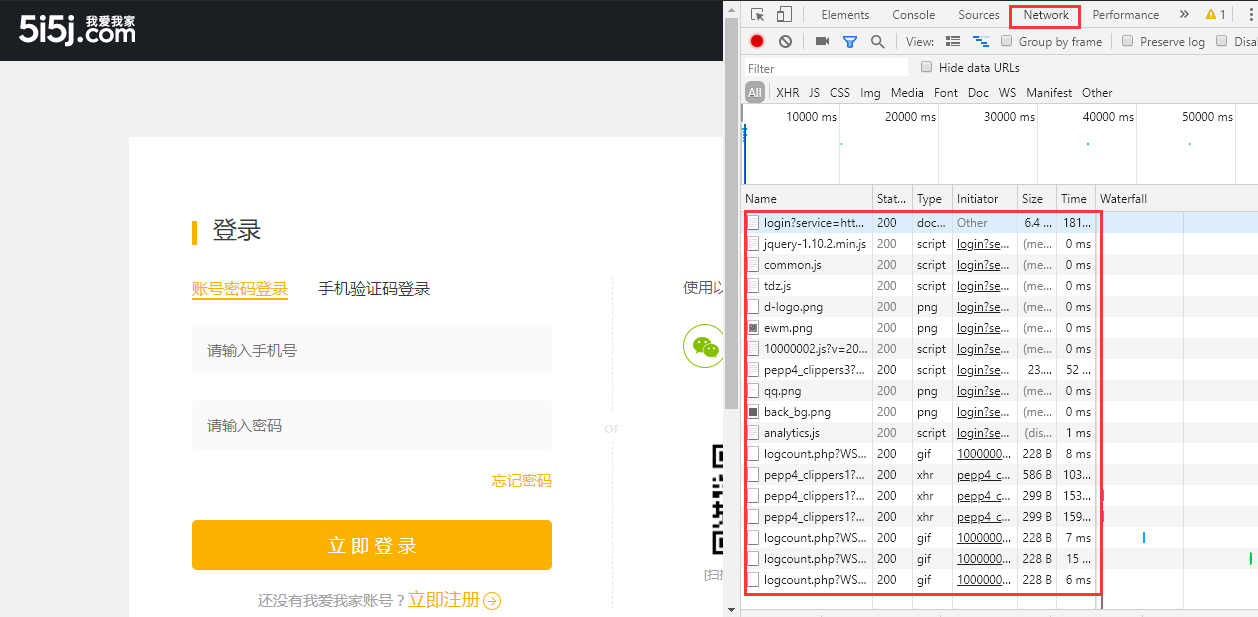
1. **获取要登陆的前置页面**



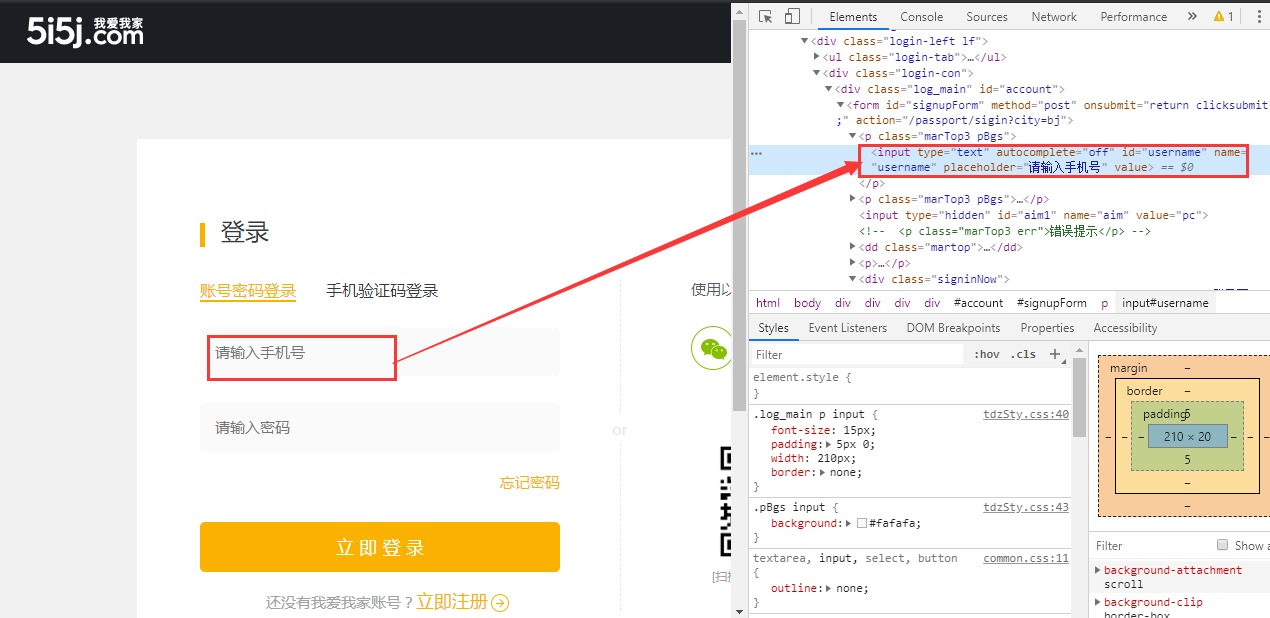
**2.打开抓包工具进行持续监听（注意：为了防止浏览器自动会话保持，我们使用无痕浏览）**



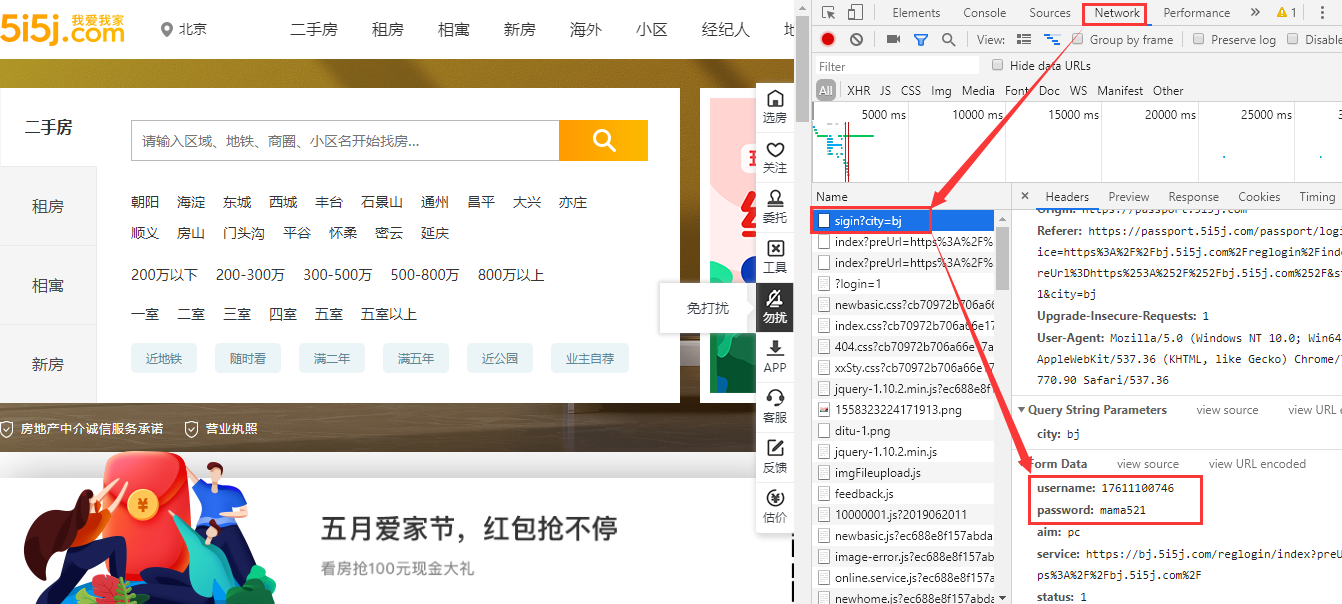
**3、请求登陆页面持续抓包**



**4、进行登陆界面结构分析**



**6、清空记录进行正确登陆一次，进行分析确认**



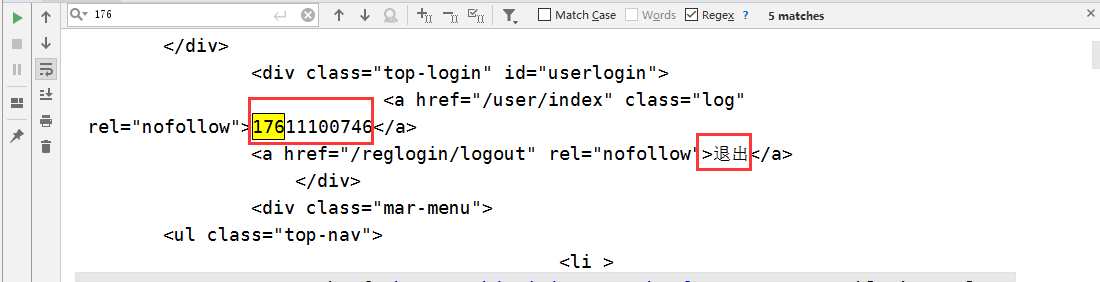
**通过以上的分析我们知道：**

**登录请求的网站为：**[**https://passport.5i5j.com/passport/sigin?city=bj**](https://passport.5i5j.com/passport/sigin?city=bj)

**该网站请求的方式为：POST （大部分的登录注册都为POST）**

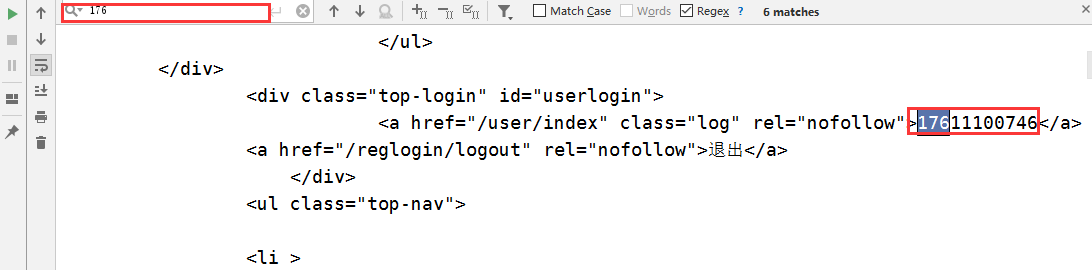
**3.2.1 第一种登录方法**

**直接在给服务器发送数据，示例如下：  
import requests  
from fake\_useragent import UserAgent  
headers = {  
'Refer':'https://passport.5i5j.com/passport/login?service=https%3A%2F%2Fbj.5i5j.com%2Freglogin%2Findex%3FpreUrl%3Dhttps%253A%252F%252Fbj.5i5j.com%252F&status=1&city=bj',  
 'User-Agent':UserAgent().random  
}  
data = {  
 'username': '17611100746',  
 'password': 'mama521',  
 'aim': 'pc',  
 'service': 'https://bj.5i5j.com/reglogin/index?preUrl=https%3A%2F%2Fbj.5i5j.com%2F',  
 'status': '1'  
}  
request = requests.session()  
base\_url ='https://passport.5i5j.com/passport/sigin?city=bj'  
res = request.post(url=base\_url,headers=headers,data=data)  
print(res.text)**



**第二种直接获取cookie进行登录**

**import requests  
from fake\_useragent import UserAgent  
  
headers = {  
 'User-Agent':UserAgent().random,  
 'Cookie':'yfx\_c\_g\_u\_id\_10000001=\_ck19062011510118832553136352305; \_ga=GA1.2.1475804441.1561002662; \_gid=GA1.2.540602414.1561002662; Hm\_lvt\_94ed3d23572054a86ed341d64b267ec6=1561002665; wiwj\_token\_ST-130792-G2rt091GxYsVKbITX45H-passport.5i5j.com=%7B%22uid%22%3A%226257157%22%7D; yfx\_s\_u\_id\_10000001=6257157; yfx\_s\_u\_name\_10000001=17611100746; webim\_token\_6257157=YWMtK52fZJMPEemdhaNJlFg0sUhiN\_DRZBHmt4eJit69xiBvJew6SvUR6ZgX0ySpLDT6AwMAAAFrcwXxKQBPGgBspLbTwce08Asl6GTjQMKzdndFDw1n-0wXFzW9P\_exAA; domain=bj; PHPSESSID=ST-131136-cTrZ1vaIp9bIQEHw7kDF-passport5i5jcom; user\_info=TBYRRFZKXlNdAUFbEANWAQIAVAMABwEOQ08VXF1RDgtRXVUSChIBBwYBAQEAAAcEBhIcEkVDVUJ5VBIKEgYCBQcBBQcSTQ; wiwj\_token\_ticket=ST-131136-cTrZ1vaIp9bIQEHw7kDF-passport.5i5j.com; wiwj\_token\_ST-131136-cTrZ1vaIp9bIQEHw7kDF-passport.5i5j.com=%7B%22uid%22%3A%226257157%22%7D; \_Jo0OQK=1C572A3F961128528960637F4806CCF6E4163736568F18B0CB397F616FA3614F09488C33CF6B443C4D7C59C245201DF4D25EE14E83ED3BB624145237EBFE03A3B47C57212F12283777C840763663251ADEB840763663251ADEB73E48B72E3F8F8C52350674422DE2517GJ1Z1Ww==; yfx\_f\_l\_v\_t\_10000001=f\_t\_1561002661865\_\_r\_t\_1561002661865\_\_v\_t\_1561011370659\_\_r\_c\_0; \_gat=1; Hm\_lpvt\_94ed3d23572054a86ed341d64b267ec6=1561011371'  
}  
url = 'https://bj.5i5j.com/user/index/'  
request = requests.session()  
response = request.get(url,headers=headers)  
print(response.text)**

**运行效果如下：**

**3.3 验证码识别**

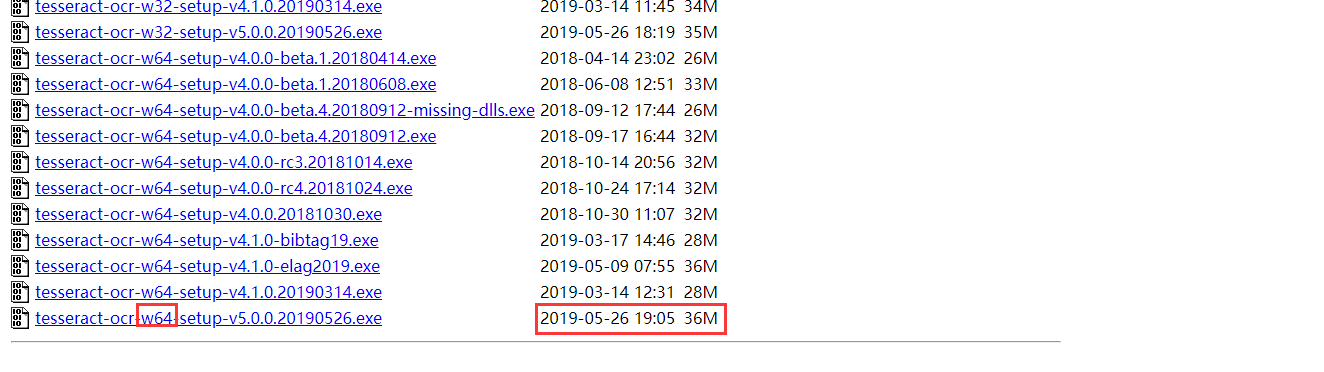
**如果你是一个数据挖掘爱好者，那么验证码是你避免不过去的一个天坑，和各种验证码斗争，必然是你成长的一条道路，但你可能听说了，在进行机器识别的过程中，你要收集样本，对图片去噪、二值化、提取字符、计算特征，甚至还要祭出神经网络去训练数据进行机器学习……**

**其实根本不想去钻研那么多高深的理论，只想要寥寥数行Python代码就搞定它，然后把主要精力投入到更重要的渗透测试中去。在这种情况下，tesseract就能帮上大忙了。**

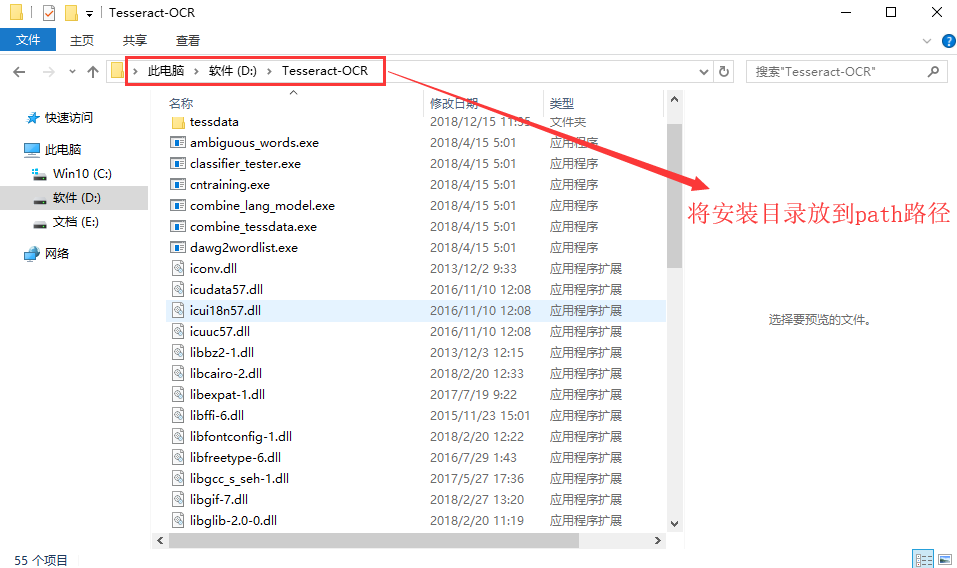
**3.3.1 安装tesseract-OCR**

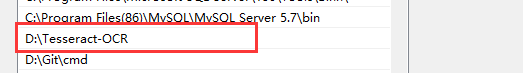
**Tesseract：开源的OCR识别引擎，初期Tesseract引擎由HP实验室研发，后来贡献给了开源软件业，后经由Google进行改进，消除bug，优化，重新发布。**

**安装目录：**[**https://digi.bib.uni-mannheim.de/tesseract/**](https://digi.bib.uni-mannheim.de/tesseract/)



**安装的时候需要自行选择安装的语言，一些其他国家的语言可以不选择安装，我之按安装了中文，英文和日语。安装过程和其他软件一样。**





**3.3.2 tesseract和python交互操作  
 1. pytesseract讲解  
 pytesseract是Python的一个OCR识别库，但其实是对tesseract做的一层Python API封装，pytesseract是Google的Tesseract-OCR引擎包装器；所以它们的核心是tesseract,因此在安装tesserocr之前，我们需要先安装tesseract**

**Pytesseract的安装: pip install pytesseract**

**顺便安装下处理图片的库：pip install pillow**

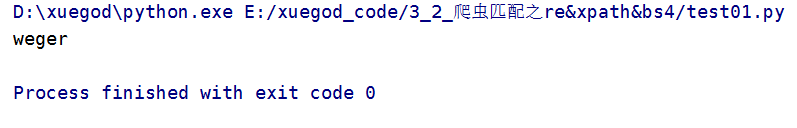
1. **Pytesseracrt 使用**

**由于该模块语法非常简洁直接让我们上实例，接下来我准备几张图片供我们识别**



***#pip install pillow ,图片处理模块*from PIL import Image  
*#pip install pytesseract ,图文识别模块*import pytesseract  
*#打开图像*img = Image.open('13.png')  
*#识别图像*result = pytesseract.image\_to\_string(img)  
*#打印结果*print(result)**

**运行结果如下：**



**经过上面两个例子来看，这个识别效果还是可以的但是效果不太好，如果想要处理更好，也要对数据图像更高深的理解以及掌握。**

**我们也可以和第三方公司合作，使用他们的产品比如：百度AI。在这里我把操作文档发给大家，可以在课下作为一个拓展。**

**总结：**

**3.1、模拟登录概述**

**3.2、requests 模拟登陆**

**3.3、普通验证码识别**