# 爬虫手册

# 分析

1. 确定网页内容是静态还是动态加载（查看网页源码）
2. 确定动态内容是否通过AJAX获取（查看网络请求XHD)
3. 确定动态内容是保存在js文件中（查看js文件）

通过网路请求分析网页/js文件=>性能高，结构清晰，| 复杂、多变、对前端要求高

selenium分析->简单 | 性能低

性能高，不缺钱选择1

# robots文件

位置：[www.baidu.com/robots.txt](http://www.baidu.com/robots.txt)

语法：

User-agent:Baiduspider //爬虫名

Allow:/article

Allow:/$ //首页

Disallow:/

User-Agent:\*

Disallow:/

# HTTP请求报文

GET / HTTP/1.1 #协议

User-Agent: #必填，用户认证

Accept-Encoding: gzip, deflate, br #压缩方式，不能填写

Cookie: BAIDUID=AAAAAAAAA; #获取登陆后状态，

referer: 代表你从哪一个网站过来的

Host: www.baidu.com #选填

Connection: keep-alive #连接方式、选填

Accept:#返回数据

Accept-Language: zh-CN,zh;q=0.9 #语言

“User-Agent”:” Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36”

# 解析语法

## 正则

查看手册

## beautifulsoup

from bs4 import BeautifulSoup

# html是爬虫获取的html页面代码

# html.parser是参数

bs = BeautifulSoup(html, “html.parser”)

**解析方法**

html\_tag = bs.title # 获取title

html\_tag = bs.header #获取header

print(html\_tag.string ) #打印解析变量

**find方法**

bs.find(“div”) #返回第一条符合要求

bs.find(id=”info”) # 查找id为info的节点

bs.find(“div”, id=”info”) # 在div节点下查找id为info的节点

bs.find\_all(“div”) #获取所有元素

bs.find(“div”, {“class”:input}).parent

bs.find(“div”, {“class”:input}).parents #迭代器

bs.find(“div”, {“class”:input}).next\_siblings #兄弟节点

***#使用正则查找***

bs.find(“div”, id=re.compile(“info-\d”))

bs.find(string=re.compile(“\d”)

param[‘class’]/param\_get(“class”)

**# 获取指定属性节点值**

bs.find(“input”,attrs={“name”:”\_fxsrf}).get(“value”)

input：标签

attrs：属性{“value: value”}

get()：要获取的值

## xpath和css选择器

pip install lxml

**xpath选择器**

from scrapy import selector # scrapy内置xpath解析库

# 将文本转换为xml对象

sel = Selector(text=html)

# 使用xpath语法选择

tag = sel.xpath(“//\*[@id=”info”]/text()”)

tag.extract() # 获取字符串

通过google游览器的copy xpath可直接生成xpath语法

**css选择器-查看css语法**

\* 所有节点

#container id为container的节点

.container 选取所有class包含container的节点

li a 选取li下所有container的节点

ul + p 选取ul下第一个p元素

div#container > ul 选取id为container的div的第一个ul子元素

from scrapy import selector # scrapy内置xpath解析库

# 将文本转换为xml对象

sel = Selector(text=html)

sel.css(“.info::text”).extract() //通过css选择器

# 六、selenium/phantomJS

selenium:

用于web应用程序的测试工具，selenium可以直接操作游览器，可进行脚本测试，支持多个平台

phantomJS无界面游览器（需要从网站下载）

## 安装

pip install selenium

## 简单使用

#控制游览器的核心

# 需要填写GoogleDriver位置，GoogleDriver需要下载，百度搜索

# browser = webdirver.Firefox() #火狐核心

# browser = webdirver.Ie() # IE核心

from selenium import webdirver

browser = webdirver.Chrome(executable\_path=”..”) #谷歌游览器核心

browser.get(“http://www.baidu.com”) # 发送请求

browser.page\_source # 网站源码（html）

browser.close() #关闭游览器

# TODO：转xml对象，使用xpath解析

# 无界面selenium

## 无界面

from selenium.webdriver.chrome.options import Options

chrome\_option = Options() #初始化设置

chrome\_option.add\_argument(“--headless”) # 无界面

# google文档中提到加上这个参数，规避bug

chrome\_option.add\_argument(“--disable-gpu”)

## 不请求图片

chrome\_option.add\_argument(“blink-settings=imagesEnabled=false”)

webdirver.Chrome(“executable\_path=”, **chrome\_options**=chrome\_option)

## selenium查找一个元素

click\_ele = browser.find\_element\_by\_xpath()

click\_ele = browser.find\_element\_by\_name(“name”).send\_keys(“value”)

click\_ele = browser.find\_element\_by\_id(“xxx”).click()

**获取iframe中元素**

#切换到iframe中，iframe中元素取不到

browser.switch\_to.frame(browser.find\_element\_by\_tag\_name(“iframe”))

## selenium点击事件

1. 获取这个元素、click\_ele
2. 调用点击方法,每次点击后都应调用browser.page\_source方法

click\_ele.send\_keys(“value”)

click\_ele.click()

## 获取cookie

browser.get\_cookie() #获取cookie

cookies = browser.get\_cookie()

for item in cookies:

cookie\_dict[item[‘name’]] = item[‘value’]

request.get(url, cookies=cookie\_dict)

## 滑动验证码

1. **窗口最大化截图**

browser.maxmize\_window() # 重要

1. **输入用户名密码**
2. **移动鼠标到元素上，显示出没有缺口的图片下载**

#找到滑动元素

ele\_A = browser.find\_element\_by\_xpath(“//..”)

# 导入类

from seleium.webdriver import ActionChains

# 将鼠标移动到指定元素上

ActionChains(browser).move\_to\_element(ele\_A).perform()

# 截取图片

driver.save\_screenshot(“xxx.png”) #截屏

超级鹰/云打码

auth2.0 jwt登录模式

# 七：机器视觉/Tesseract

## 安装

## 使用

import pytesseract

from PIL import Image

image = Image.open(“”)

str = image\_to\_string(image) #将图片转换为文本

# 反爬策略

User-agent

IP访问频率

账号访问频率

动态网页-加大分析难度

前端JS加密混淆

机器学习分析爬虫新闻

只请求html, 不请求js和CSS

CSS下毒

## nginx user-agent反扒

# 禁止Scrapy 等工具抓取

if($http\_user\_agent ~\* (Scrapy|Curl|HttpClient|python)){

return 403;

}

# 禁止非GET|HEAD|POST方式抓取

if($request\_method !~ ^(GET|HEAD|POST)$){

return 403;

}

生成随机的user-agent

pip install fake\_useragent

from fake\_useragent import UserAgent

ua = UserAgent()

ua.random

## 通过IP设置

阿布云：自动代理IP

ip.zdaye.com