Jinke Cao

Jkcao@iflytek.com

MMA分类爬虫文档

曹今可

目录

[1 爬虫流程简介 2](#_Toc459803029)

[2 构造有效网络请求 2](#_Toc459803030)

[2.1.1 浏览器 2](#_Toc459803031)

[2.1.2 Wireshark + Android模拟器 2](#_Toc459803032)

[2.1.3 无头浏览器 2](#_Toc459803033)

[2.2 解析请求响应 5](#_Toc459803034)

[2.2.1 常用Xpath解析 5](#_Toc459803035)

[2.2.2 常用css解析 5](#_Toc459803036)

[2.2.3 Scrapy 5](#_Toc459803037)

[2.2.4 BeautifulSoup 5](#_Toc459803038)

[2.2.5 Selenium 6](#_Toc459803039)

[2.3 反爬虫应对策略 6](#_Toc459803040)

[2.3.1 Tor代理服务器+pysocks 6](#_Toc459803041)

[2.3.2 Tesseract 训练验证码识别 6](#_Toc459803042)

[2.3.3 JS Ajax 重定向 修改请求头 修改cookie等 8](#_Toc459803043)

[3 京东网站爬虫 8](#_Toc459803044)

[4 汽车之家爬虫 9](#_Toc459803045)

[5 丝芙兰爬虫 9](#_Toc459803046)

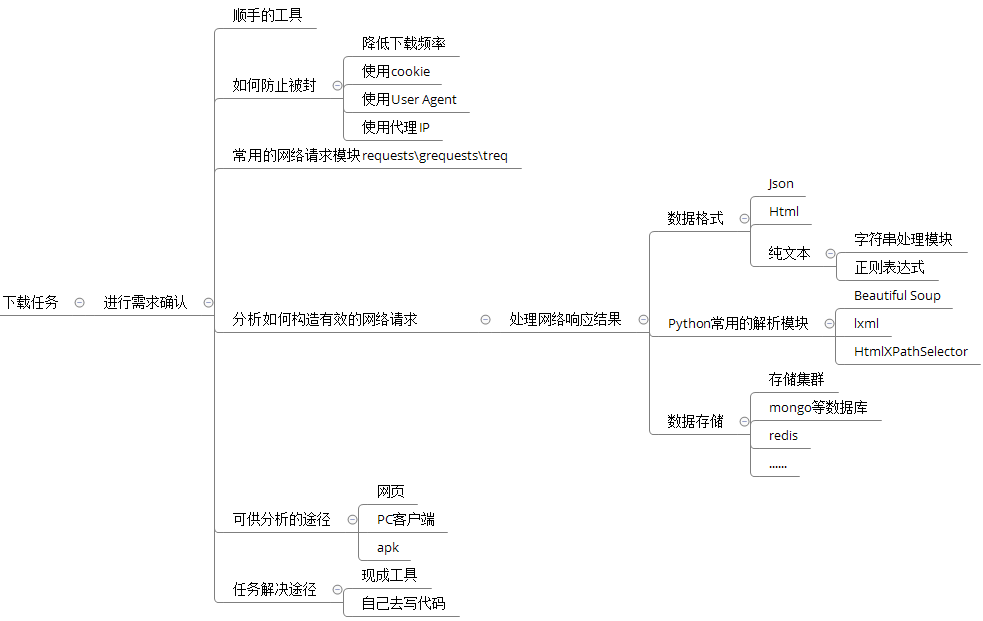
[6 百度爬虫 9](#_Toc459803047)

[7 优数广告 10](#_Toc459803048)

[8 1688爬虫 10](#_Toc459803049)

[9 中关村在线爬虫 10](#_Toc459803050)

# 爬虫流程简介



下载任务主要有上图所示的几部分，其中构造有效的网络请求和解析请求响应的结果两步是下载中的关键。

# 构造有效网络请求

### 浏览器

可作为主流网页数据的查看入口，按下F12后选择network选项，即可查看相应页面元素请求的url，为编写爬虫请求提供支持。

### Wireshark + Android模拟器

是一个常用的网络封包分析软件，可以列出尽可能详细的网络封包资料，在模拟手机端登陆或者客户端访问的时候，可以展示通过网卡的数据交换的情况，为编写爬虫请求提供依据。

### 无头浏览器

通常情况下，通过URL请求可以直接获得想要的内容；有些时候，通过URL请求有部分字段数据是需要通过分析js或者是异步加载的；有些URL请求的参数难以被构造，构造的代价比较大或者目前自己的能力解决不了；这时可以使用以下方法：从两个维度考虑,尝试能不能解决问题：模拟浏览器,常用的有Webkit和phantomjs(两者均为非图形化的无头浏览器)，以及自动化测试工具Webdriver。

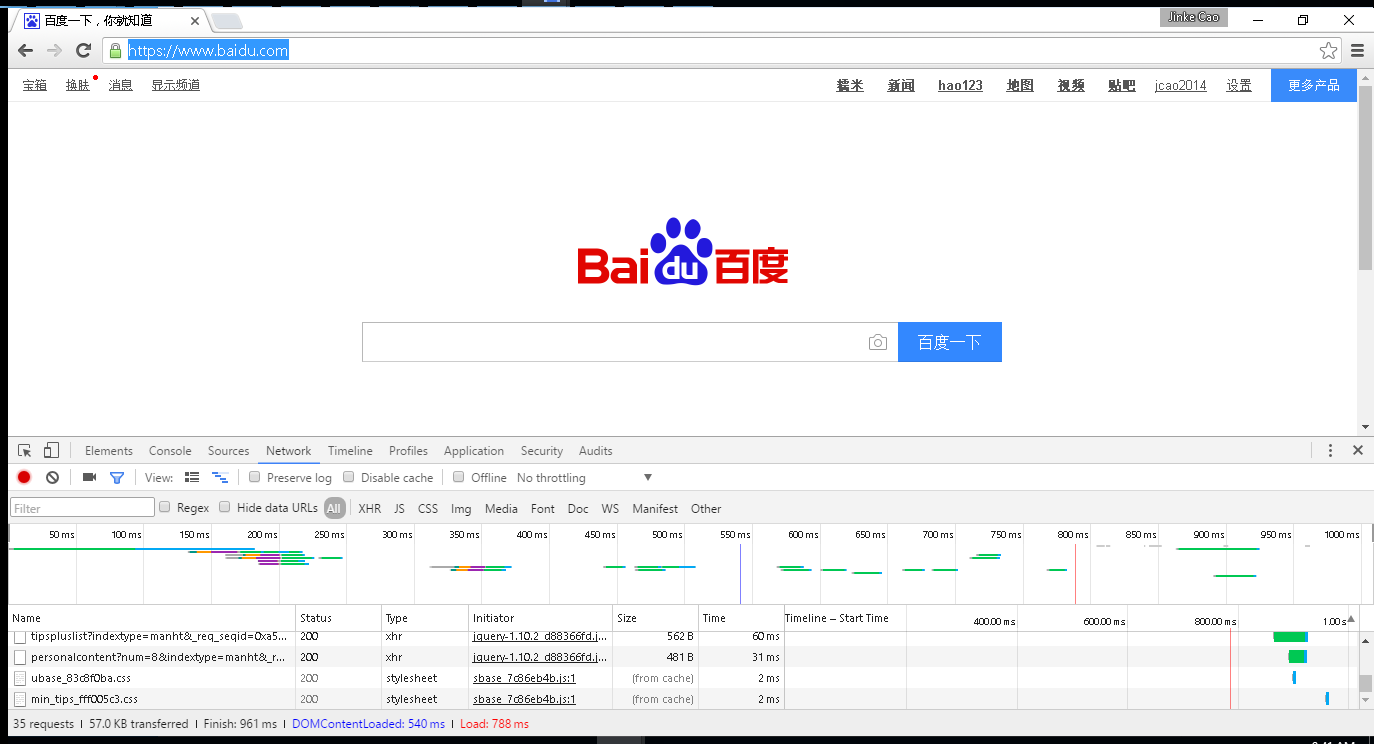


Figure 1 浏览器分析request

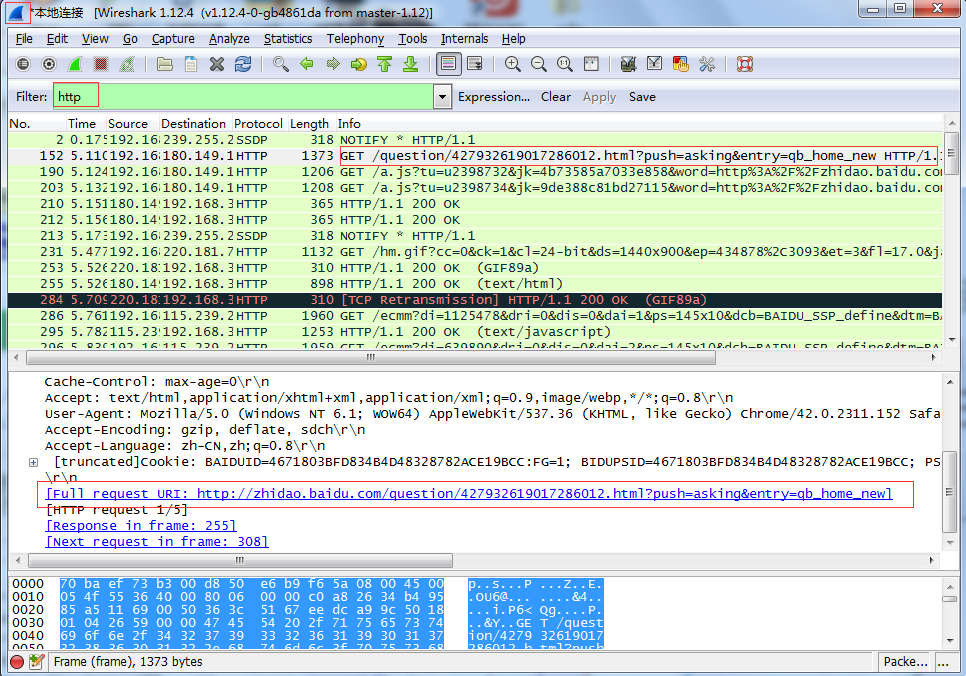


Figure 2 wireshark封包分析



Figure 3 模拟手机访问/登录

Webkit：

import webkit\_server

c = webkit\_server.Client()

c.visit(url)

l = c.css("#searchResultTotalCountRegion")

page\_num = l[0].text()

PhantomJS

from tornado\_fetcher import Fetcher

# create a fetcher

fetcher=Fetcher(

user\_agent='phantomjs', # user agent

phantomjs\_proxy='http://localhost:12306',

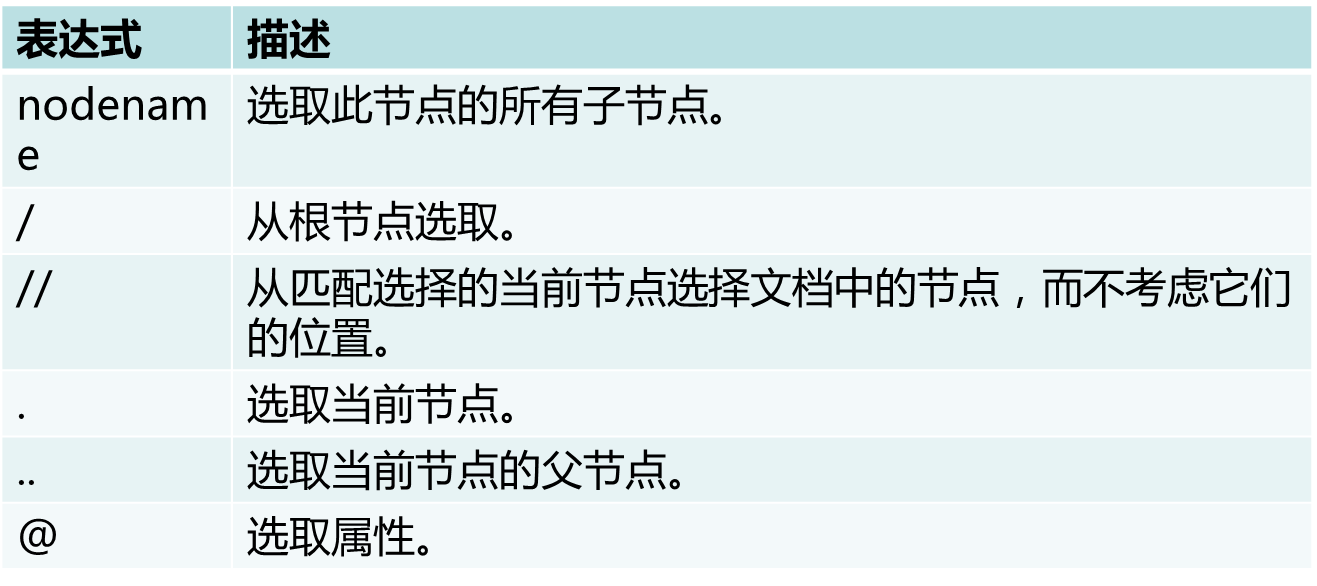
pool\_size=10, # max httpclient num

async=False )

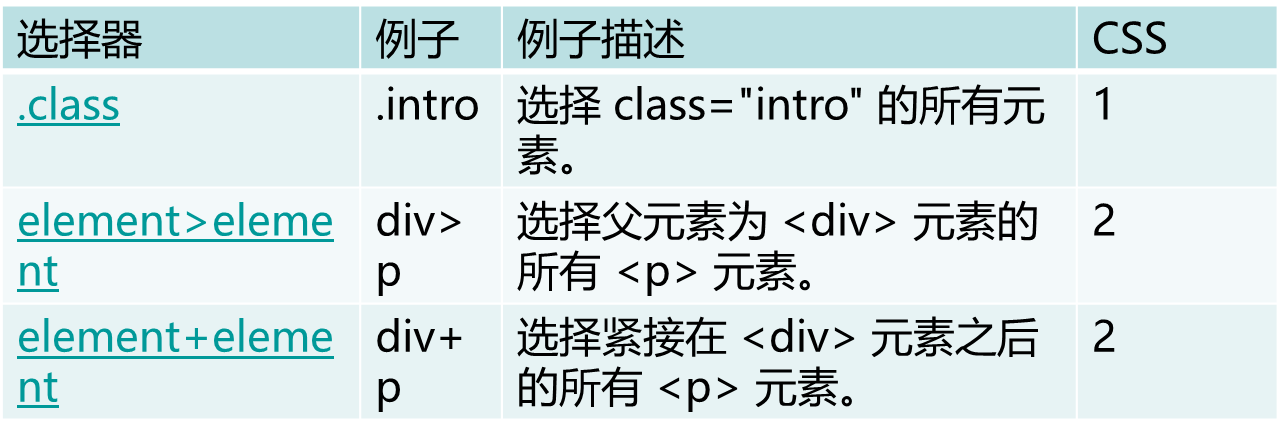
body = fetcher.phantomjs\_fetch(url)

## 解析请求响应

### 常用Xpath解析



### 常用css解析



常用解析工具：

### Scrapy

是一个为了爬取网站数据，提取结构性数据而编写的应用框架。 可以应用在包括数据挖掘，信息处理或存储历史数据等一系列的程序中。也可以应用在获取API所返回的数据(例如 Amazon Associates Web Services ) 或者通用的网络爬虫。

### BeautifulSoup

使用定位HTML标签来格式化和组织复杂网络信息，用简易的python对象来展示xml信息；相对于lxml而言，具有更高的灵活性；

### Selenium

通常和phantomJS一起使用，对于JS等动态网页提供支持。Selenium在webDriver的DOM中使用了全新的查找元素，不过它们都是非常直接的名称；

## 反爬虫应对策略

### Tor代理服务器+pysocks

洋葱路由网络，是一种IP地址匿名手段，通过不同服务器构成多个层，把客户端包在里面。数据进入网络之前会被加密，任何服务器都不能偷取通信数据。虽然每个服务器入站和出战的通信都可以被查到，但是想查清整个通信链路上所有服务器的入站和出站的通信节点，几乎是不可能的事。

Pysocks是一个非常简单的python代理服务器通信莫管，可以和tor配合使用。

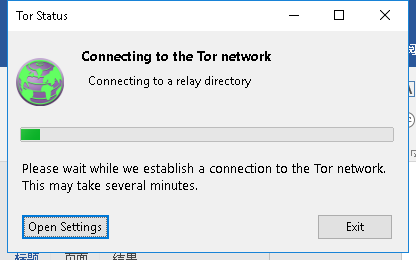


Figure 4 tor链接

Pysocks：

from selenium import webdriver

service\_args = [ '--proxy=localhost:9150', '--proxy-type=socks5', ]

driver = webdriver.PhantomJS(executable\_path='<path to PhantomJS>', service\_args=service\_args)

driver.get("http://icanhazip.com")

print(driver.page\_source)

driver.close()

### Tesseract 训练验证码识别

Tesseract提供了一套完整的验证码图片训练工具，在频繁访问某网站时，通过验证码训练 ，可以在一定概率上通过验证码测试。测试可能需要好几个小时。随后可以使用tesseract包在训练结果的基础上对图片进行解析。



from urllib.request import urlretrieve

from urllib.request import urlopen

from bs4 import BeautifulSoup

import subprocess

import requests

from PIL import Image

from PIL import ImageOps

def cleanImage(imagePath):

image = Image.open(imagePath)

image = image.point(lambda x: 0 if x<143 else 255)

borderImage = ImageOps.expand(image,border=20,fill='white')

borderImage.save(imagePath)

html = urlopen("http://y")

bsObj = BeautifulSoup(html)

#Gather prepopulated form values

imageLocation = bsObj.find("img", {"title": "Image CAPTCHA"})["src"]

formBuildId = bsObj.find("input", {"name":"form\_build\_id"})["value"]

captchaSid = bsObj.find("input", {"name":"captcha\_sid"})["value"]

captchaToken = bsObj.find("input", {"name":"captcha\_token"})["value"]

captchaUrl = "http://pythonscraping.com"+imageLocation

urlretrieve(captchaUrl, "captcha.jpg")

cleanImage("captcha.jpg")

p = subprocess.Popen(["tesseract", "captcha.jpg", "captcha"], stdout=

subprocess.PIPE,stderr=subprocess.PIPE)

p.wait()

f = open("captcha.txt", "r")

#Clean any whitespace characters

captchaResponse = f.read().replace(" ", "").replace("\n", "")

print("Captcha solution attempt: "+captchaResponse)

if len(captchaResponse) == 5:

params = {"captcha\_token":captchaToken, "captcha\_sid":captchaSid,

"form\_id":"comment\_node\_page\_form", "form\_build\_id": formBuildId,

"captcha\_response":captchaResponse, "name":"Ryan Mitchell",

"subject": "I come to seek the Grail",

"comment\_body[und][0][value]":

"...and I am definitely not a bot"}

r = requests.post("",

data=params)

responseObj = BeautifulSoup(r.text)

if responseObj.find("div", {"class":"messages"}) is not None:

print(responseObj.find("div", {"class":"messages"}).get\_text())

else:

print("There was a problem reading the CAPTCHA correctly!")

### JS Ajax 重定向 修改请求头 修改cookie等

使用selenium框架，尽量使用post而不是get请求。在填写表单时，可能会出现页面上没有显示但是html文件里出现的表单，用来诱导爬虫填写。处理方式如下：

from selenium import webdriver

from selenium.webdriver.remote.webelement import WebElement

driver = webdriver.PhantomJS(executable\_path='')

driver.get("http:// /pages/itsatrap.html")

links = driver.find\_elements\_by\_tag\_name("a")

for link in links:

if not link.is\_displayed():

print("The link "+link.get\_attribute("href")+" is a trap")

fields = driver.find\_elements\_by\_tag\_name("input")

for field in fields:

if not field.is\_displayed():

print("Do not change value of "+field.get\_attribute("name"))

针对不同网站制定不同策略，详见下文。

# 京东网站爬虫

|  |  |
| --- | --- |
| 目标网站 | 京东 |
| 抓取目标 | 商品分类信息 商品简介 |
| 输出格式 | csv |
| 反爬虫应对策略 | 使用tor实现ip滚动和ip匿名，避免ip封锁 |
| 下载框架 | scrapy |
| 解析框架 | scrapy |

对接商品分类信息索引建立需求。主要使用了ip滚动的方式。三天内下载了281mb的商品信息。

# 汽车之家爬虫

|  |  |
| --- | --- |
| 目标网站 | 汽车之家 |
| 抓取目标 | 分类信息 车型信息 |
| 输出格式 | csv |
| 反爬虫应对策略 | 更改request header和token，模拟手机登录 |
| 下载框架 | urllib |
| 解析框架 | beautifulSoup |

对接计算广告组需求。模拟安卓浏览器登录，完成全量网站车型数据下载。

# 丝芙兰爬虫

|  |  |
| --- | --- |
| 目标网站 | sephora |
| 抓取目标 | 商品分类信息 商品简介 |
| 输出格式 | json |
| 反爬虫应对策略 | 使用tor实现ip滚动和ip匿名，避免ip封锁 |
| 下载框架 | selenium |
| 解析框架 | selenium |

对接广告主分类数据需求，使用ip滚动，两小时完成42mb数据下载。

# 百度爬虫

|  |  |
| --- | --- |
| 目标网站 | 百度 |
| 抓取目标 | 搜索结果的标题，链接，网页简介 |
| 输出格式 | json |
| 反爬虫应对策略 | 使用tesserAct对验证码图片进行训练识别 |
| 下载框架 | urllib |
| 解析框架 | beauitfulSoup |

对接app兴趣分类需求，使用验证码图片训练，6天完成112w条数据搜索结果。

# 优数广告

|  |  |
| --- | --- |
| 目标网站 | 优数广告 |
| 抓取目标 | 广告视频 广告图片 |
| 输出格式 | flv，MP4，jpg， png |
| 反爬虫应对策略 | 使用代理ip翻墙 |
| 下载框架 | urllib |
| 解析框架 | urllib |

对接广告特征提取需求，使用ip代理，3小时完成8.5g数据下载。

# 1688爬虫

|  |  |
| --- | --- |
| 目标网站 | 1688批发网 |
| 抓取目标 | 商品分类信息 商品简介 |
| 输出格式 | csv |
| 反爬虫应对策略 | 使用phantomJS穿越表单，模拟人类浏览网页 |
| 下载框架 | phantomJS |
| 解析框架 | selenium |

对接MMA需求，使用selenium+phantomJS处理动态网页，2小时完成21m数据下载。

# 中关村在线爬虫

|  |  |
| --- | --- |
| 目标网站 | 中关村在线 |
| 抓取目标 | 商品分类信息 商品简介 |
| 输出格式 | csv |
| 反爬虫应对策略 | 使用百度 搜狗等引擎的缓存来代理采集 |
| 下载框架 | Requests |
| 解析框架 | beautifulSoup |

对接标签标注需求，极大减轻了人工标注的任务量，两小时完成4k条设备信心标注，准确率超过80%。