### 1.request 和 response

requests库是一个常用于http请求的模块，性质是和urllib，urllib2是一样的，作用就是向指定目标网站的后台服务器发起请求，并接收服务器返回的响应内容。

### 2.requests.get()方法使用

所谓的get方法，便是利用程序使用HTTP协议中的GET请求方式对目标网站发起请求，同样的还有POST，PUT等请求方式，其中GET是我们最常用的，通过这个方法我们可以了解到一个请求发起到接收响应的过程。

import requests

start\_url = 'https://www.baidu.com'

response = requests.get(url=start\_url)

print(response) # 返回值：<Response [200]>

这是一个最简单实现请求的方式，最后返回一个响应对象，响应对象中携带的数值便是HTTP状态码，你可以根据这些状态码的值来判定请求的成功以及推测失败的原因。

这并不是一个完整的请求，因为服务器在接收到你的程序的请求信息时，它可以明确的从Request headers中看到你是在用程序发起请求接收响应，为了完善我们的请求，可以自定义请求头信息，利用get()方法的headers参数。  
我们可以从浏览器的F12开发者工具中Network下找到请求网址的headers信息，保存下来作为我们自定义参数。如：

import requests

# 目标url

start\_url = 'https://www.baidu.com'

# 自定义headers

headers = {"Host": "www.baidu.com",

"Referer": "https://www.baidu.com",

"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36",

"X-Requested-With": "XMLHttpRequest"

}

response = requests.get(url=start\_url, headers=headers)

print(response) # 返回值：<Response [200]>

不同的网站对于请求头中的字段信息有着不同的要求，某些关键字段需要在html源码中去寻找，然后自己组织定义到headers中完成请求。  
对于一些特定的网站会对某一时间段内你所在ip的请求次数坐监测，从而判断此请求是否为非人类发出的，get()方法同样为我们提供了在请求时更换ip的操作方式，只需要像自定义headers一样定义我们的可用ip即可，如下：

import requests

start\_url = 'https://www.baidu.com'

headers = {"Host": "www.baidu.com",

"Referer": "https://www.baidu.com",

"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36",

"X-Requested-With": "XMLHttpRequest"

}

# 自定义代理ip,此处的ip需要自行更换，只需要将ip和port按格式拼接即可，可以去网上免费代理中寻找：http://www.xicidaili.com/nn

proxies = {"https": "https://127.0.0.1:1080", "http": "http://127.0.0.1:1080"}

response = requests.get(url=start\_url, headers=headers, proxies=proxies)

print(response) # 返回值：<Response [200]>

### 3.requests.post()方法使用—构造formdata表单

post请求方式的使用和get方式并没有很大的区别，本质的区别在于它传递参数的方式并不像get方式一样，通过在url中拼接字段来发送给服务器，他采取了一种相较之下更为安全的操作方式，通过form表单的方式来向服务器传递查询条件。我们同样可以通过浏览器的F12开发者工具或者fiddler抓包工具来看到formdata这个字段，从中获取form表单中的字段信息，很多登录操作就是基于此。操作方式如下：

import requests

url = 'https://en.artprecium.com/catalogue/vente\_309\_more-than-unique-multiples-estampes/resultat'

headers = {

"Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8",

"Accept-Language": "zh",

"Cache-Control": "max-age=0",

"Connection": "keep-alive",

"Cookie": "PHPSESSID=m2hbrvp548cg6v4ssp0l35kcj7; \_ga=GA1.2.2052701472.1532920469; \_gid=GA1.2.1351314954.1532920469; \_\_atuvc=3%7C31; \_\_atuvs=5b5e9a0418f6420c001",

"Host": "en.artprecium.com",

"Upgrade-Insecure-Requests": "1",

"User-Agent": "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36",

}

# 构造form表单

data = {"IdEpoque": "",

"MotCle": "",

"Order": "",

"LotParPage": "All",

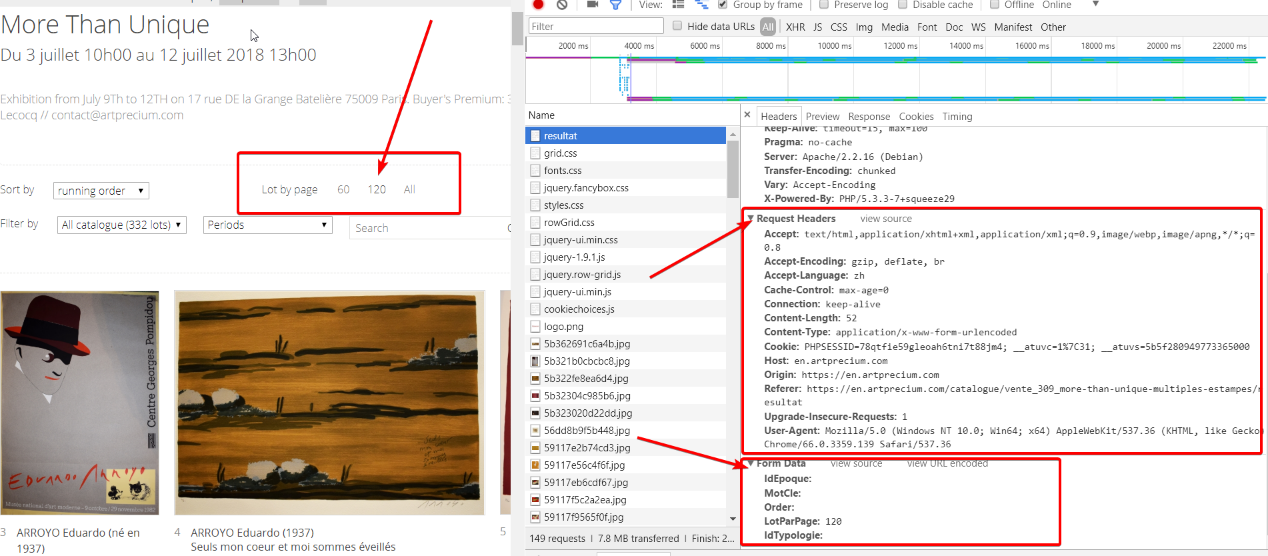
"IdTypologie": ""}

response = requests.post(url=url, data=data, headers=headers, timeout=10)

print(response) # 返回值：<Response [200]>

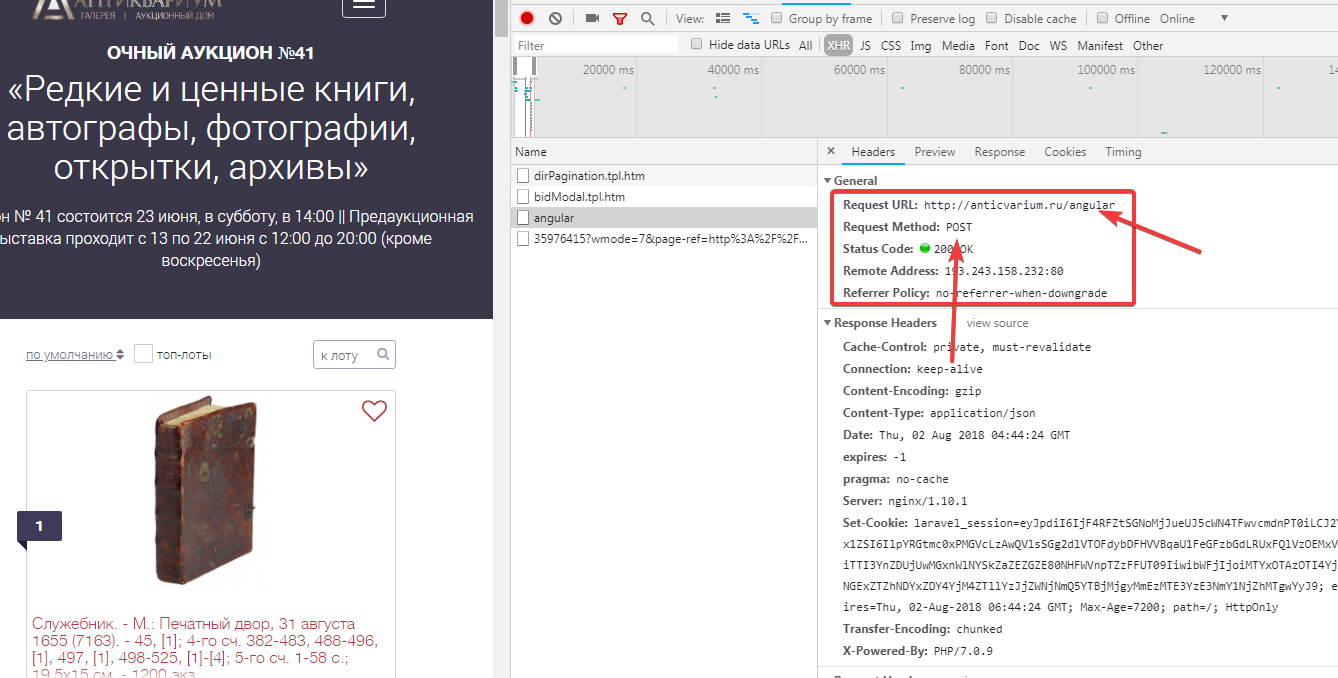
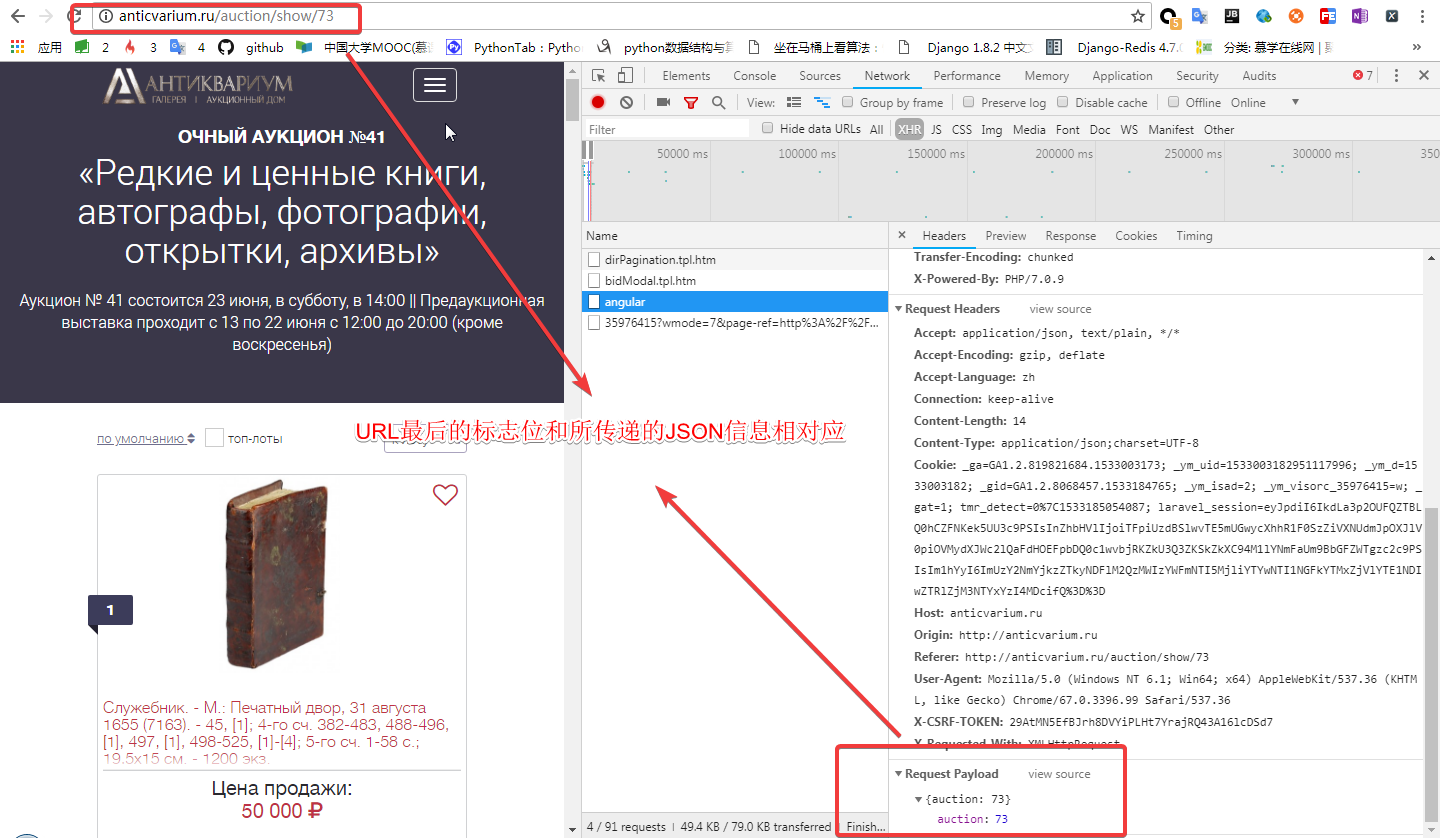
上面是一个电商网站的post请求查询的案例，当我们对页面显示商品数量进行更改时发现我们的url并没有发生改变，此时，我们便可以分析此动作是由ajax异步加载或者是通过post的请求方式来更改，我们可以通过开发者工具来获取我们想要的信息。

目标网址测试



我们可以很容易的看到请求头信息和form表单信息，通过修改LotParPage字段信息可以获得不同数量商品的响应，在实际操作过程中要通过自己的测试确定headers中所必需的字段和自定义我们的form表单从而完成请求，对于post的其他参数和get大致相同，基本通用，有兴趣的话可以去官方查看requests的api文档。

4.requests.post()方法使用—发送json数据

post除了构造表单以外还可以像通过像服务器发送json信息的方式获取正确的请求，利用的便是中post(json={"key":"value"})的方式，操作流程大致与post发送form表单相似，案例是一个国外网站的demo，虽然全是一些看不懂的文字，但是无伤大雅，看个流程就好，目标网址：[http://anticvarium.ru/auction/archive](http://anticvarium.ru/auction/archive" \t "_blank)  
从这个页面中我们可以得到一个专场列表页，每个专场中有许多商品，当我们直接去请求每个专场的URL时不会得到任何结果，我们可以从开发者工具中找到他的真实接口，如下图所示  
  
我们在看它像服务器传递的信息：  
  
通过以上的分析，我们可以

url = 'http://anticvarium.ru/angular'

headers = {

"Host": "anticvarium.ru",

"Connection": "keep-alive",

"Content-Length": "14",

"Origin": "http://anticvarium.ru",

"X-CSRF-TOKEN": "u1InfvhE23slcmReIJlgvI7IGzb3xQEvXHQbP3Bc",

"User-Agent": "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36",

"Content-Type": "application/json;charset=UTF-8",

"Accept": "application/json, text/plain, \*/\*",

"X-Requested-With": "XMLHttpRequest",

"Referer": "http://anticvarium.ru/auction/show/66",

"Accept-Language": "zh",

"Cookie": "\_ga=GA1.2.819821684.1533003173; \_gid=GA1.2.258681840.1533003173; \_ym\_uid=1533003182951117996; \_ym\_d=1533003182; \_ym\_isad=2; \_ym\_visorc\_35976415=w; tmr\_detect=0%7C1533004777867; laravel\_session=eyJpdiI6InZNSTlWSlYyUW1BaitUWTNBSjJKckE9PSIsInZhbHVlIjoibWprTDFxeTZZc0lOS2VTcmxIdzZJSkwwTVFXOHphN2FwTysxbXNHSDkybXE0VUZYakpTUlZ5Ykh6U2pFTTVJTWdWUHBUaVJwVGl1Q3RlTjdRQUhKNEE9PSIsIm1hYyI6IjQ5NmIwMGYxNDllZDFiNTY2ZmFjYWY5NjEyOTQ4MTZjYzQzNmFiY2EyMzFiOTY1Mzg3ODUyZTllN2U0MTQwYTEifQ%3D%3D",

}

response = requests.post(url, headers=headers, json={"auction": 66})

print(response) # 返回值：<Response [200]>