

# Python实现接口测试中的常见四种Post请求数据

0.953 2017.08.29 20:43:22 字数 1867 阅读 14979

前情：

在日常的接口测试工作中，模拟接口请求通常有两种方法，利用工具来模拟，比如fiddler，postman，poster，soapUI等  
利用代码来模拟，使用到一些网络模块，比如HttpClient，或者requests等

Fiddler是一个简单的http协议调试代理工具，它界面友好，易于操作，是调试网络，抓取请求的利器之一。

Requests包是趁着目前如火如荼的爬虫风潮而流行起来的Python的第三方包，使用简单，对于不需要对网络底层模块有太多了解又想实现接口测试的测试人员来说简直不要太友好。

在接口测试中，接口通常是GET请求或者POST请求。以下是对接口测试中常见的四种Post请求数据方式进行一个详细的讲解：

## post请求主体详解

一个正常的post请求主要包括请求行，请求头，请求主体

对于get请求来说没有请求主体entity-body。对于post请求而言，不会对发送请求的数据格式进行限制，理论上你可以发任意数据，但是服务器能不能处理就是另一回事了。服务器收到数据后，如何解析数据呢？它会以请求头中的Content-Type设置的内容来进行数据解析。确定好Content-Type的格式之后，请求主体的数据格式也就确定下来了。

Content-Type的格式有四种：分别是application/x-www-form-urlencoded（这也是默认格式）、application/json、text/xml以及multipart/form-data格式。

这些不同的post请求数据格式要通过HttpEntity来构造，有必要简单理一下HttpClient的HttpEntity对象，因为所有的post请求数据均需要置于HttpEntity实体中进行发送。HttpEntity是一个接口，实现这个接口的具体类有很多，比较常用的是StringEntity、UrlEncodedFormEntity（继承自StringEntity）、MultipartEntity。他们将在发送不同格式的post请求时被用到。接下来就详细地介绍每一种数据格式对应的fiddler请求模拟和Requests请求模拟（Python实现）的实现情况。

## （一）application/x-www-form-urlencoded数据格式

在W3C官网上明确对这种数据格式进行了定义：

This is the default content type. Forms submitted with this content type must be encoded as follows:Control names and values are escaped. Space characters are replaced by '+', and then reserved characters are escaped as described in [RFC1738], section 2.2: Non-alphanumeric characters are replaced by '%HH', a percent sign and two hexadecimal digits representing the ASCII code of the character. Line breaks are represented as "CR LF" pairs (i.e., '%0D%0A').The control names/values are listed in the order they appear in the document. The name is separated from the value by '=' and name/value pairs are separated from each other by '&'.

这是post请求最常见也是默认的数据提交格式。它要求数据名称（name）和数据值（value）之间以等号相连，与另一组name/value值之间用&相连。例如：



CC先生之简书

关注

拥有357钻 (约55.00元)

敏捷团队中如何度量需求时效 (Python版)

阅读 41

七日打造你的专注倾听

阅读 42

夏令时的困惑

阅读 32

### 精彩继续

好女人，都是有弹性的

阅读 4204

秦岚曝光女明星身材真相：越没有自我的人...

阅读 2151



写下你的评论...



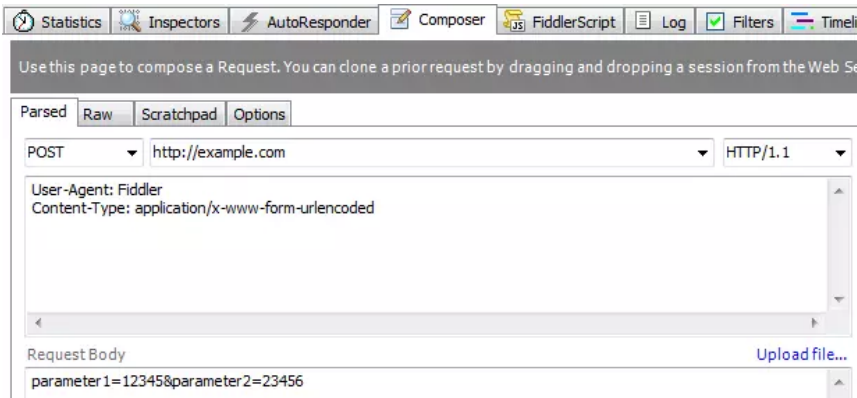
评论0



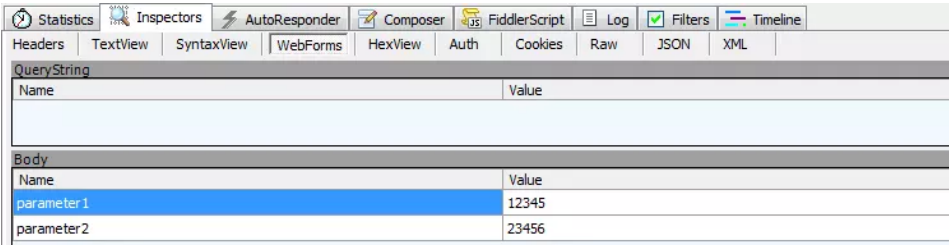
赞18



(1) 如果用fiddler模拟请求的话，请求头和请求主体的内容可以这样构造：



模拟请求之后，从返回结果可以查看到我们的请求数据：



(2) 如果用Requests模拟post请求的话，请求可以这样构造：

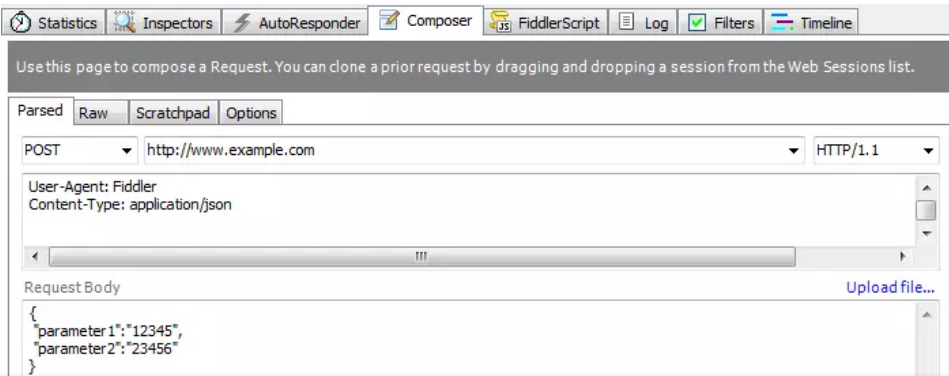
```
1 datas = {'parameter1': '12345', 'parameter2': '23456'}
2 r = requests.post('http://example.com', data=datas)
3 print(r.content)
4 print(r.status_code)
```

Reqeusts支持以form表单形式发送post请求，只需要将请求的参数构造成一个字典，然后传给requests.post()的data参数即可。

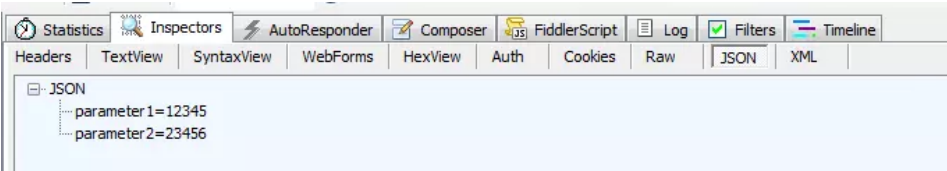
**\*\* (二) application/json 数据格式 \*\***

application/json 格式的请求头是指用来告诉服务端post过去的消息主体是序列化后的 JSON 字符串。

(1) 如果用fiddler模拟请求的话，请求头和请求主体的内容可以这样构造：



写下你的评论... 评论0 赞18 ...



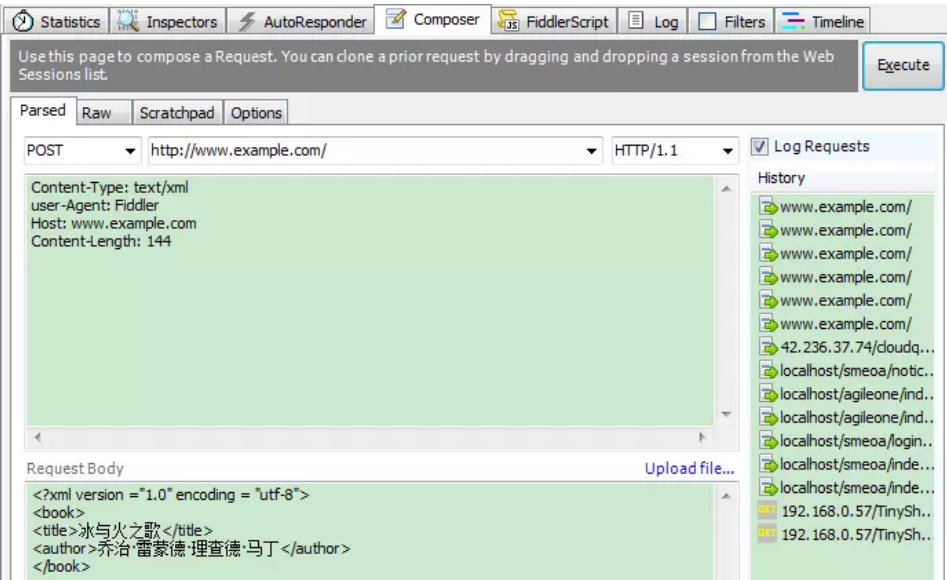
( 2 ) 如果用Requests模拟post请求的话，请求可以这样构造：

```
1 url = 'http://www.example/post'
2 s = json.dumps({'key1': 'value1', 'key2': 'value2'})
3 r = requests.post(url, data=s)
4 print (r.text)
```

这里我们可以发现Requests模拟post请求时，请求头格式为application/x-www-form-urlencoded与application/json的主要差别在于请求主体的构造格式（前者是键值对，后者是JSON串），前者直接用字典传入，后者用json.dumps()函数将字典转为JSON串即可。

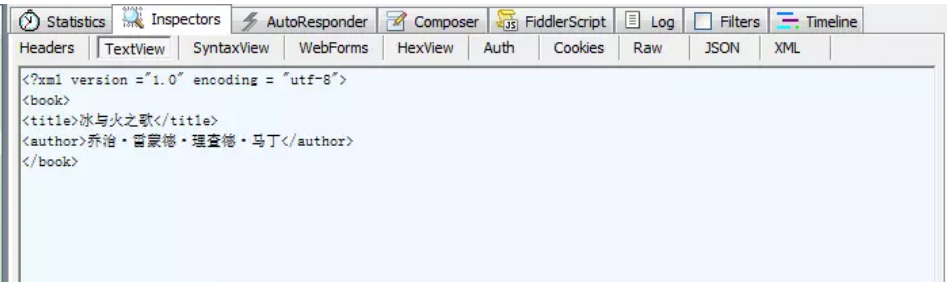
( 三 ) text/xml数据格式

( 1 ) 如果用fiddler模拟请求的话，请求头和请求主体的内容可以这样构造：



xmlrequest.png

模拟请求之后，从返回结果可以看到我们的请求数据：



xmlresponse.png

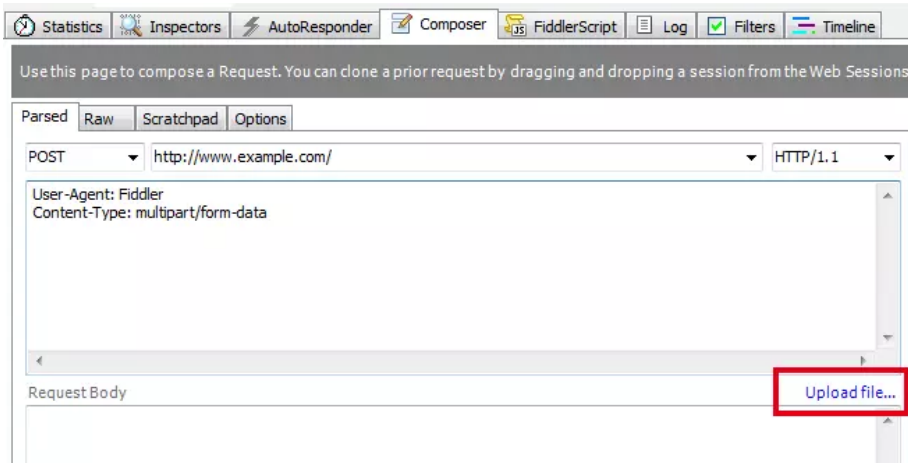
( 2 ) 如果用Requests模拟post请求的话，请求可以这样构造：

```
1 import requests
2
3 def request_ws(request):
4     with open(archivo_request,"r") as archivo:
5         request_data = archivo.read()
6
7     target_url = "http://127.0.0.1:8000/?wsdl"
8
9     headers = {'Content-type':'text/xml'}
10
11     data_response = requests.post(target_url, data=request_data, headers=headers)
```

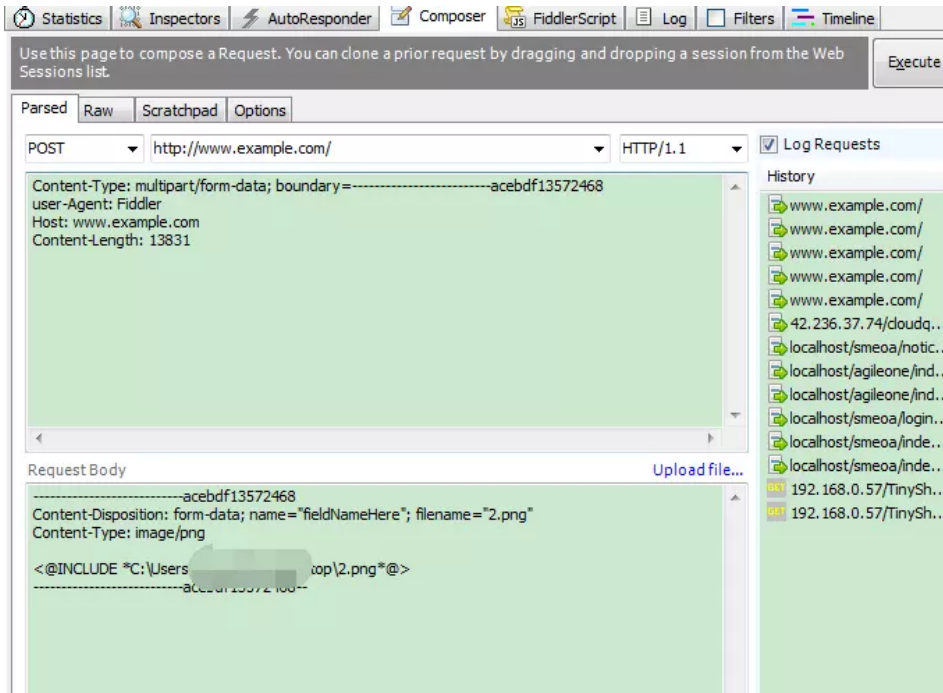
(四) multipart/form-data数据格式

除了传统的application/x-www-form-urlencoded表单，我们另一个经常用到的是上传文件用的表单，这种表单的类型为multipart/form-data。

(1) 如果用fiddler模拟请求的话，请求头和请求主体的内容可以这样构造：  
第一步，先设置好请求头格式，然后点击upload file...



第二步，上传你的文件，这里我上传一个png的图片



段从概率上说“几乎不可能”的数据即可。每次post浏览器都会生成一个随机的30-40位长度的随机字符串，浏览器一般不会遍历这次post的所有数据找到一个不可能出现在数据中的字符串，一般都是随机生成。选择了这个边界之后，浏览器便把它放在Content-Type 里面传递给服务器，服务器根据此边界解析数据。下面的数据便根据boundary划分段，每一段便是一项数据。(每个field被分成小部分，而且包含一个value是"form-data"的"Content-Disposition"的头部；一个"name"属性对应field的ID等等，文件的话包括一个filename)模拟请求之后，从返回结果可以看到我们的请求数据：

Headers

TextView

SyntaxView

WebForms

HexView

Auth

Cookies

Raw

JSON

XML

QueryString

Name	Value

Content-Type: multipart/form-data is not yet fully supported.

Name	Value
Content-Disposition: form-data; name="fieldNameHere"; filename="2.png"	<file>
Content-Type: image/png	

response.png

( 2 ) 如果用Requests模拟post请求的话，请求可以这样构造：

```
1 url = 'http://www.example.com'
2 files = {'file': open('C://Users//Someone//Desktop//2.png', 'rb')}
3 r = requests.post(url, files=files)
4 print(r.text)
```

综上，就是接口测试中fiddler与requests模拟post接口四种请求数据的构造方法，总结起来有利于在以后的接口测试工组过程中可以及时查阅。

同时，Requests库还可以处理多部分编码(Multipart-Encoded)的文件，比如xls文件，甚至改写content-Type的报头。  
比如：

```
1 url = 'http://httpbin.org/post'
2 files = {'file': ('report.xls', open('report.xls', 'rb'), 'application/vnd.ms-excel', {'Expires':
3
4 r = requests.post(url, files=files)
5
```

<

>

👍 18人点赞 >

👎

📖 接口测试

⋮

 CC先生之简书  
拥有357钻 (约55.00元)

关注

"想转载的，给杯茶钱呀~~"

赞赏

全部评论 0

只看作者

按时间倒序

按时间正序

被以下专题收入，发现更多相似内容

 Python

推荐阅读

更多精彩内容 >

### 干货！接口测试中模拟post四种请求数据

一、背景介绍 在日常的接口测试工作中，模拟接口请求通常有两种方法，fiddler模拟和HttpClient模拟。 ...

 隋胖胖LoveFat


### Fiddler模拟post四种请求数据

前言： Fiddler是一个简单的http协议调试代理工具，它界面友好，易于操作，是模拟http请求的利器之一。 ...

 小喜\_ww

### Spring Cloud

Spring Cloud为开发人员提供了快速构建分布式系统中一些常见模式的工具（例如配置管理，服务发现，断路器，智...

 卡卡罗2017

### 百战程序员V1.2——尚学堂旗下高端培训\_ Java1573题

百战程序员\_ Java1573题 QQ群：561832648489034603 掌握80%年薪20万掌握50%年薪...

 Albert陈凯

### 四种常见的 POST 提交数据方式 <http://www.aikaiyuan.com/632...>

HTTP/1.1 协议规定的 HTTP 请求方法有 OPTIONS、GET、HEAD、POST、PUT、DELET...

 seventhboy