

Lecture 14: Post Web Form Data via Python

1. 主要问题：Web表单的自动提交

1. 对于普通用户来说，如果可以自动提交表单，将会为我们的生活带来很大的便利性。还记得过去三年，大家在疫情期间所经历的那些事情吧？其中，每天我们都要做的事情，就是进行健康报告。大多数日子，我们身体的健康状况是没有任何变化的，那么，在此情况下，我们是否可以通过Python编程自动提交我们的健康状况呢？
2. 对于恶意用户来说，他们的目的可能只是破坏Web服务。我们听说过一种叫作拒绝服务（Denial of service, DoS）的攻击吧？恶意用户通过向服务器发送大量数据包，使得服务器来不及一一响应，从而无法为新的用户请求提供服务。利用表单的自动提交，在某种意义上说，就可以实现DoS攻击。
3. 对于开发者来说，需要考虑系统的负载性能和安全性。会不会有人对我们Web服务进行恶意攻击呢？我们所开发的服务能经受多少人同时访问呢？2009年，12306铁路订票系统上线了。然而，在春运期间就发生了订票系统的拒绝服务，因为并发访问的人太多了，服务器无法及时响应了，从而导致整个订票系统瘫痪。所以，开发者需要一种对系统进行充分的压力测试和安全性测试。使用代码进行表单数据的自动提交，并验证响应结果，是压力测试和安全性测试中常用的手段。

每日健康申报(教职工版)

今天是2022年05月15日

1、人事工号

请输入人事工号

2、姓名

请输入姓名

3、性别

☐ 男

☐ 女

4、单位院系

请输入单位院系

5、工作地点所属校区

☐ 朝晖校区

☐ 屏峰校区

☐ 莫干山校区

☐ 朝晖校区德胜园(假山路校园)

☐ 其他

6、人员类别

2. Web表单数据提交的主要代码

```
1 import requests
2
3
4 url = '' # 对表单数据进行处理的服务器的 url
5 payload = {} # Web请求所发送的数据荷载
6
7 response = requests.post(url=url, json=payload)
```

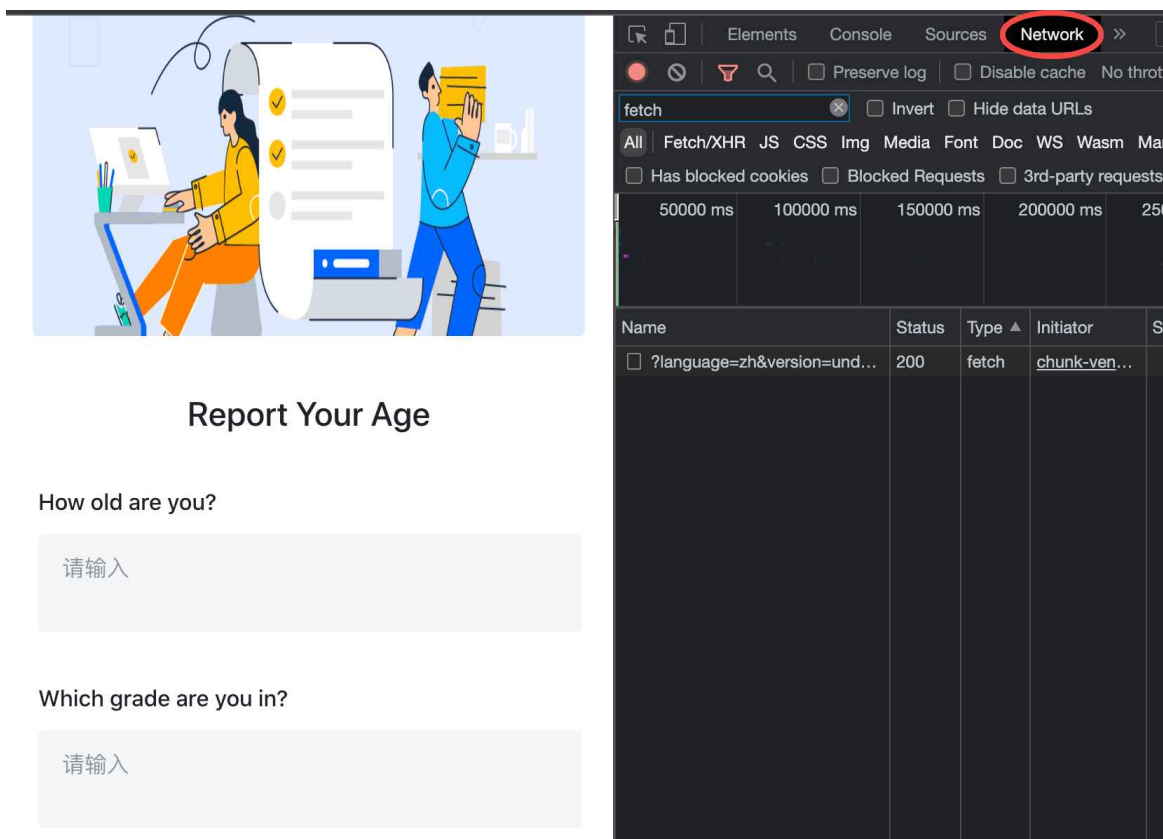
从上面的代码，可以看出，为了使用Python进行数据POST，我们需要明确其中的 `url` 和 `payload`，这需要对表单页面所发送的数据进行分析。

3. 对Web页面所发送数据的分析

这里，以一个简单的调查问卷为例，进行分析；这里假定你所使用的浏览器是：Google Chrome 或 Microsoft Edge。

3.1 主要步骤

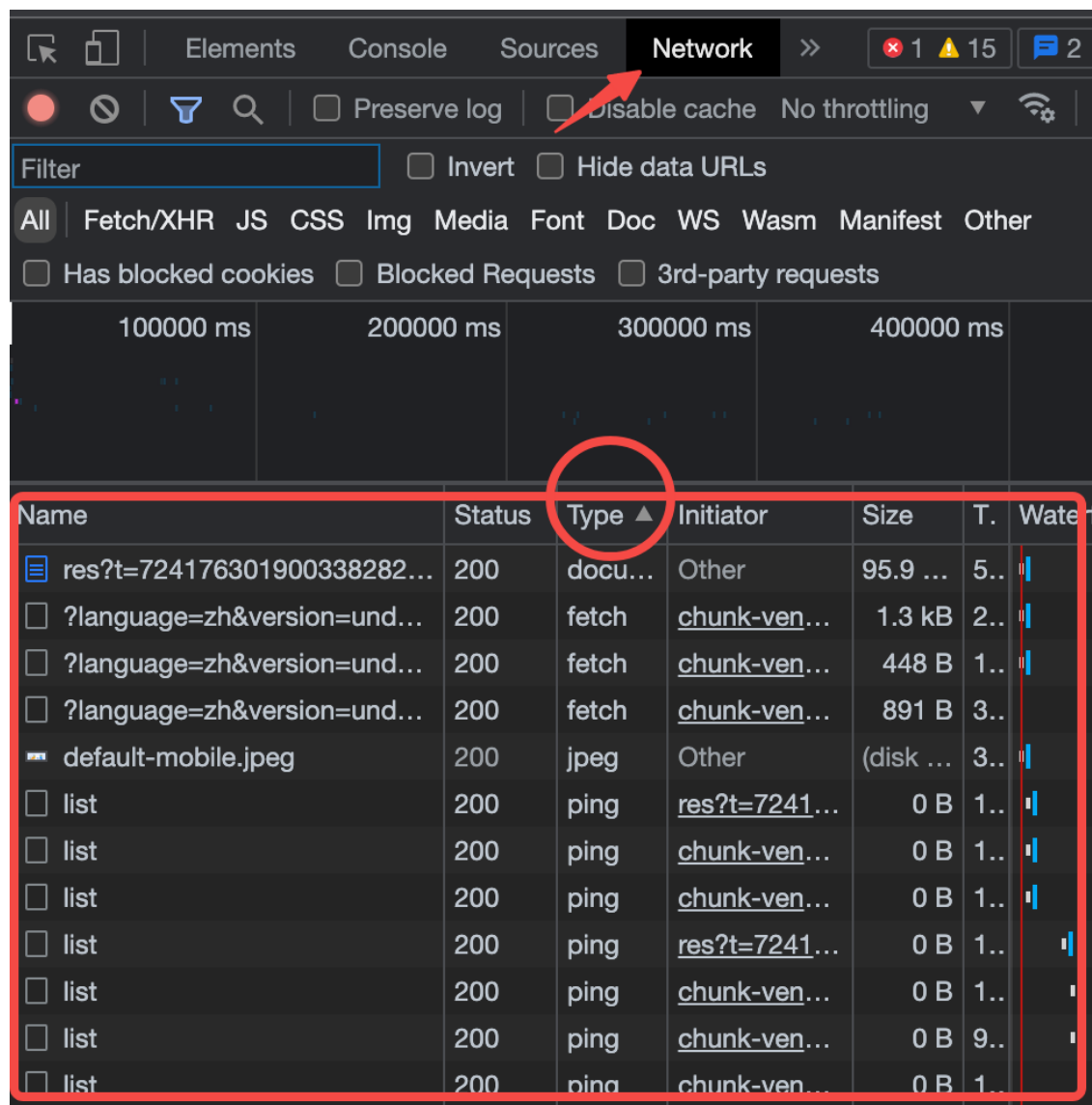
1. 打开问卷：<https://wenjuan.feishu.cn/m/cfm?t=7241763019003382821-yvmo>；
2. 按 `F12` 打开浏览器的开发者工具（关于浏览器的开发者工具，[这里](#)有个不错的教程）；



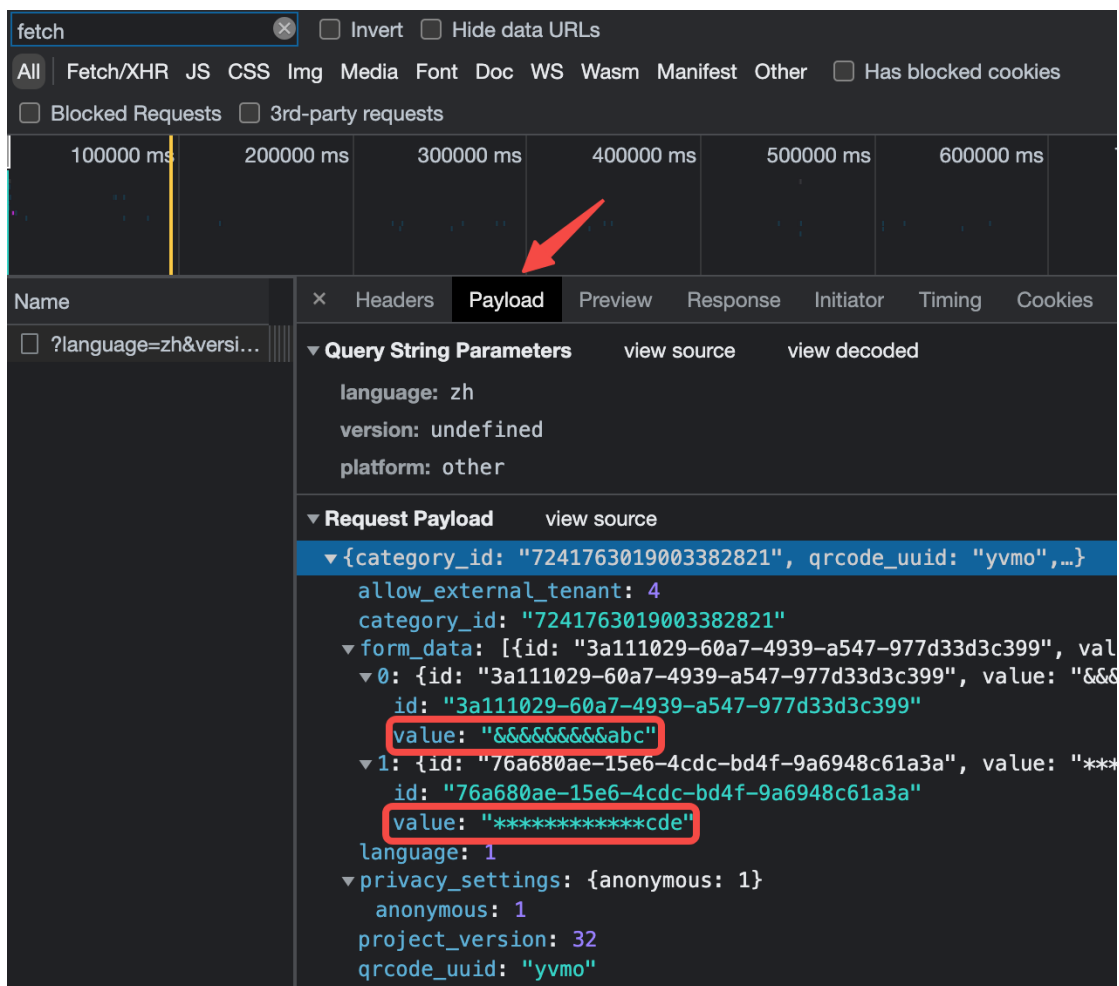
The image shows a web form titled "Report Your Age" on the left and the Chrome DevTools Network tab on the right. The form has two input fields: "How old are you?" and "Which grade are you in?", both with placeholder text "请输入". The Network tab shows a single request to "chunk-ven..." with a status of 200 and type of fetch. The "Network" tab in the DevTools interface is circled in red.

Name	Status	Type	Initiator
?language=zh&version=und...	200	fetch	chunk-ven...

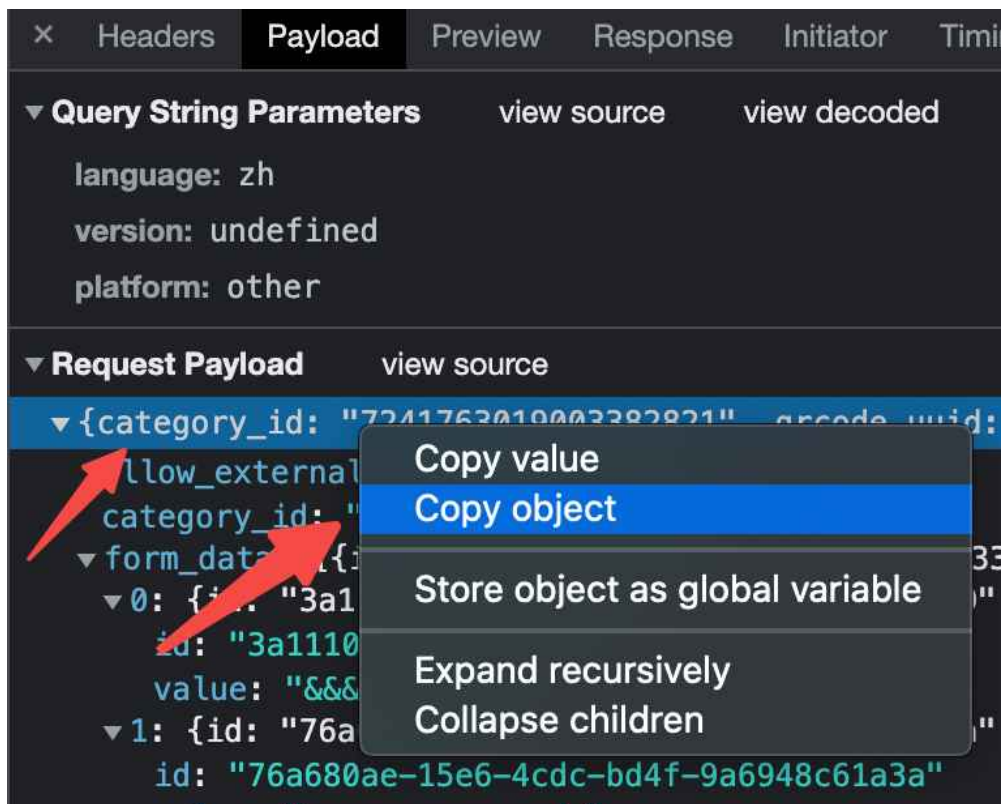
- 3. 打开开发者工具中的“Network”；
- 4. 手动填写问卷中的信息，点击“提交”；
- 5. 观察开发者工具“Network”中的输出，可以看到该页面中有很多数据；该如何分析呢？我们重点需要关注数据类型，通过数据类型可以排除大部分我们不会关心的数据。



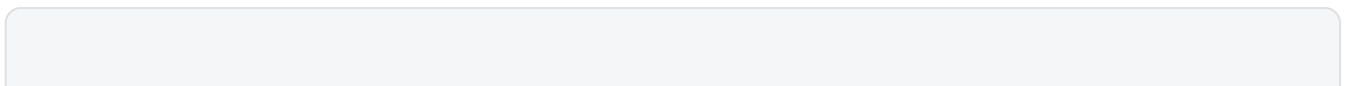
- 6. 通过分析，我们发现其中的“fetch”类型的数据是我们要关心的。
- 7. 双击打开该数据，在其“Payload”中，可以发现我们刚刚提交的问卷数据。



a. 右键点击该数据，拷贝对象：



b. 将其放入Python代码中：



```

1 import requests
2
3
4 url = '' # 对表单数据进行处理的服务器的 url
5 payload = { # Web请求所发送的数据荷载
6     "category_id": "7241763019003382821",
7     "qrcode_uuid": "yvmo",
8     "form_data": [
9         {
10             "id": "3a111029-60a7-4939-a547-977d33d3c399",
11             "value": "&&&&&&&abc"
12         },
13         {
14             "id": "76a680ae-15e6-4cdc-bd4f-9a6948c61a3a",
15             "value": "*****cde"
16         }
17     ],
18     "language": 1,
19     "privacy_settings": {
20         "anonymous": 1
21     },
22     "allow_external_tenant": 4,
23     "project_version": 32
24 }
25
26 response = requests.post(url=url, json=payload)

```

- c. 只要改变其中“黄色”高亮处的数据就可以改变我们的问卷数据。可以手动修改，当然，我们需要的是使用代码进行更改：

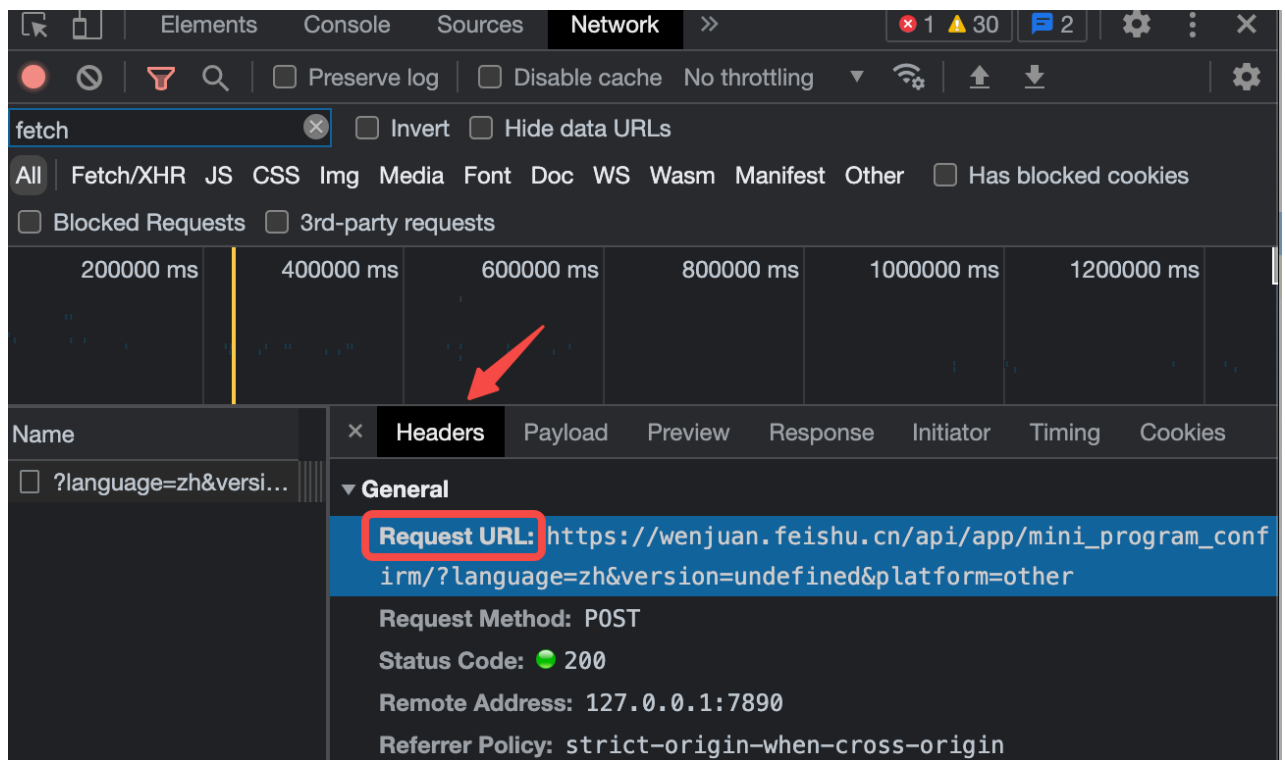
```

1 payload['form_data'][0]['value'] = 'Hello'
2 payload['form_data'][1]['value'] = 'World!'

```

8. 数据所提交到的服务器可以在Headers中获取：

- a. 在Headers中查看 Request URL：



b. 右键，Copy Value，将其放入代码中：

```
1 url = 'https://wenjuan.feishu.cn/api/app/mini_program_confirm/?  
    language=zh&version=undefined&platform=other'
```

9. 将上面的代码组织在一起，详见（3.2节）：

10. 分析 `response` 中的数据，以编写测试代码，确保我们的提交是成功的：

```
1 if '提交成功' in response.text:  
2     print('提交成功')
```

3.2 最终代码

```
1 import requests  
2  
3 payload = {  
4     "category_id": "7241763019003382821",  
5     "qrcode_uuid": "yvmo",  
6     "form_data": [  
7         {  
8             "id": "3a111029-60a7-4939-a547-977d33d3c399",  
9             "value": "&&&&&&&abc"  
10        },  
11    ],  
12 }
```

```
11         {
12             "id": "76a680ae-15e6-4cdc-bd4f-9a6948c61a3a",
13             "value": "*****cde"
14         }
15     ],
16     "language": 1,
17     "privacy_settings": {
18         "anonymous": 1
19     },
20     "allow_external_tenant": 4,
21     "project_version": 32
22 }
23
24 payload['form_data'][0]['value'] = 'Hello'
25 payload['form_data'][1]['value'] = 'World!'
26
27 url = 'https://wenjuan.feishu.cn/api/app/mini_program_confirm/?language=zh&versi
28 response = requests.post(url=url, json=payload)
29
30 if '提交成功' in response.text:
31     print('提交成功')
```