



链滴

【袁人学爬虫刷题平台】 (二) 入门级 js 参 数

作者: [gitsilence](#)

原文链接: <https://ld246.com/article/1606555216113>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



```
top
└─ match.yuanrenxue.com
   └─ match
      └─ 12
         └─ static
            └─ apps.bdimg.com
               └─ cdn.bootcss.com
                  └─ cdn.jsdelivr.net

782     });
783     if ($.cookie('no-alert') !== 'true'){
784         (function (){$('alert_ban').click()})();
785     }
786
787 </script>
788 </div>
789 <script src="/static/match/dist/js/main.min.js"></script>
790 <script>
791     window.url = '/api/match/12';
792     request = function(){
793         // 点击换页后的操作, 先得到翻到了几页
794         if (window.page){}else {window.page=1}
795         var list = {
796             "page": window.page,
797             "m": btoa('yuanrenxue' + window.page)
798         };
799         $.ajax({
800             url: window.url,
801             dataType: "json",
802             async: false,
803             data: list,
804             type: "GET",
805             beforeSend: function(request) {
806             },
807             success: function(data) {
808
809                 data = data.data;
810                 let html = '';
811                 $.each(data, function(index, val) {
812                     html += '<tds' + val.value + '</tds'
813                 });

```

yuanrenxue + 页数, 然后对其进行base64编码
用篇python访问的时候, 需要对其进行url编码

```
# -*- coding: utf-8 -*-
# @Time : 2020/11/28 16:51
# @Author : gitsilence
"""
入门级js加密
"""
```

```
import requests
import base64
import urllib.parse
```

```
url = "http://match.yuanrenxue.com/api/match/12?page=%s&m=%s"
```

```
headers = {
```

```
"User-Agent": "yuanrenxue.project",  
}
```

```
sum = 0  
for i in range(1, 6):  
    m = base64.b64encode(("yuanrenxue" + str(i)).encode("utf-8")) # 被编码的参数必须是二进制数据  
    m = urllib.parse.quote(m) # 对m的值进行url编码  
    print(m)  
    datas = requests.get(url % (i, m), headers=headers).json()  
    print(datas)  
    for data in datas['data']:  
        sum += data['value']
```

```
print("最终求和结果: ", sum)
```

```
eXVhbnJlbnh1ZTE%3D  
{'status': '1', 'state': 'success', 'data': [{'value': 62}, {'value': 4633}, {'value': 2177}, {'value': 6424}, {'value': 6567}, {'value': 956}, {'value': 2474}, {'value': 7846}, {'value': 7787}, {'value': 2789}]}
```

```
eXVhbnJlbnh1ZTI%3D  
{'status': '1', 'state': 'success', 'data': [{'value': 4542}, {'value': 9173}, {'value': 8246}, {'value': 4449}, {'value': 8026}, {'value': 374}, {'value': 4610}, {'value': 2916}, {'value': 9066}, {'value': 2618}]}
```

```
eXVhbnJlbnh1ZTM%3D  
{'status': '1', 'state': 'success', 'data': [{'value': 5150}, {'value': 2325}, {'value': 5170}, {'value': 986}, {'value': 6206}, {'value': 6412}, {'value': 5985}, {'value': 9476}, {'value': 4302}, {'value': 8450}]}
```

```
eXVhbnJlbnh1ZTQ%3D  
{'status': '1', 'state': 'success', 'data': [{'value': 8955}, {'value': 2762}, {'value': 1846}, {'value': 8320}, {'value': 8179}, {'value': 6651}, {'value': 2149}, {'value': 7217}, {'value': 1572}, {'value': 976}]}
```

```
eXVhbnJlbnh1ZTU%3D  
{'status': '1', 'state': 'success', 'data': [{'value': 1233}, {'value': 8043}, {'value': 5598}, {'value': 9933}, {'value': 4680}, {'value': 324}, {'value': 9866}, {'value': 4169}, {'value': 3414}, {'value': 998}]}
```

```
最终求和结果: 247082
```