

文档四：执行结果演示

21301114 俞贤皓

0. 目录

- [1. 如何使用本系统](#)
- [2. 效果图](#)
- [3. 关键代码](#)

1. 如何使用本系统

- 公众号：咸鱼鱼的Blog
 - 如果没反应的话，可能是后端没开。阿里云便宜服务器配置太差，不够稳定。
 - 可以直接打我电话开后端TAT，18105028615
- 进入聊天页面后，可以输入 `help` 查看手册
- Examples

```
help
# 详细手册请见 文档三

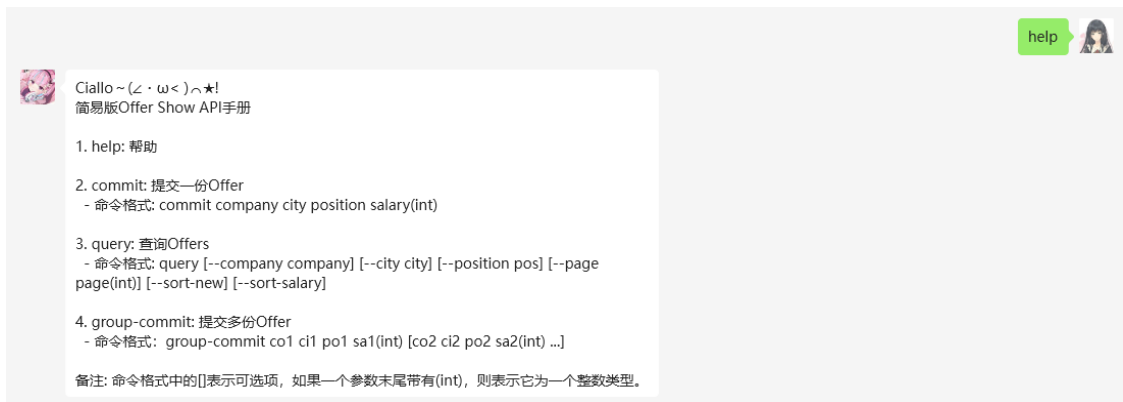
query --city 南京 --page 2
query --sort-new --page 1
query --sort-salary --sort-new --page 1
query --city 南京 --company 谷雅丰帝国
query --page 3 --sort-salary --company 谷雅丰帝国
query --page 2 --company 未来科技有限公司
query --page -1
query --page 100
query --page 3 --sort-new
query --position Up主

commit guuu帝国 南京 南京皇帝 19260817

group-commit guuu帝国 南京 南京皇帝 192608171 guuu帝国 南京 南京皇帝 192608172 guuu帝国 南京 南京皇帝 192608173 guuu帝国 南京 南京皇帝 192608174 guuu帝国 南京 南京皇帝 192608175 guuu帝国 南京 南京皇帝 192608176 guuu帝国 南京 南京皇帝 192608177 guuu帝国 南京 南京皇帝 192608178 guuu帝国 南京 南京皇帝 192608179 guuu帝国 南京 南京皇帝 1926081710
# 连续添加了10个记录
# 如果想从文件导入记录的话，需要在后台进行导入，因为个人微信号无法传文件，企业微信号才可以
```

2. 效果图

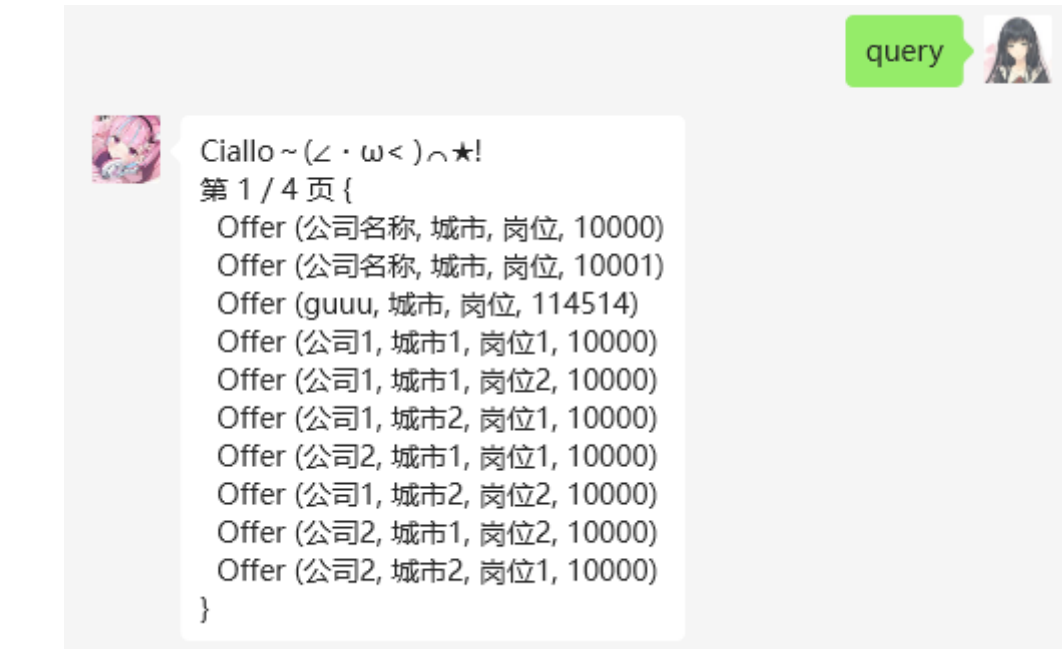
- `help`
 -



- **commit & group-commit**



- **query**



query --company 谷雅丰帝国 --page 2



Ciallo~(∠・ω<)∩★!

第 2 / 2 页 {

Offer (谷雅丰帝国, 南京, guuu, 1013)
Offer (谷雅丰帝国, 南京, guuu, 1014)
Offer (谷雅丰帝国, 南京, guuu, 1015)
Offer (谷雅丰帝国, 南京, guuu, 1016)
Offer (谷雅丰帝国, 南京, guuu, 1017)
Offer (谷雅丰帝国, 南京, guuu, 1018)
Offer (谷雅丰帝国, 南京, guuu, 1019)

}

query --company --sort-salary



Ciallo~(∠・ω<)∩★!

第 1 / 0 页 {

}

query --company 谷雅丰帝国 --sort-salary



Ciallo~(∠・ω<)∩★!

第 1 / 2 页 {

Offer (谷雅丰帝国, 南京, Up主, 1919810)
Offer (谷雅丰帝国, 南京, guuu, 1019)
Offer (谷雅丰帝国, 南京, guuu, 1018)
Offer (谷雅丰帝国, 南京, guuu, 1017)
Offer (谷雅丰帝国, 南京, guuu, 1016)
Offer (谷雅丰帝国, 南京, guuu, 1015)
Offer (谷雅丰帝国, 南京, guuu, 1014)
Offer (谷雅丰帝国, 南京, guuu, 1013)
Offer (谷雅丰帝国, 南京, guuu, 1012)
Offer (谷雅丰帝国, 南京, guuu, 1011)

}

query --city 南京 --position Up主 --sort-new



Ciallo~(∠ · ω<)∩★!

第 1 / 1 页 {

Offer (谷雅丰帝国, 南京, Up主, 1919810)
}

query --city 南京 --position guuu --sort-salary --sort-new



Ciallo~(∠ · ω<)∩★!

第 1 / 2 页 {

Offer (谷雅丰帝国, 南京, guuu, 1019)
Offer (谷雅丰帝国, 南京, guuu, 1018)
Offer (谷雅丰帝国, 南京, guuu, 1017)
Offer (谷雅丰帝国, 南京, guuu, 1016)
Offer (谷雅丰帝国, 南京, guuu, 1015)
Offer (谷雅丰帝国, 南京, guuu, 1014)
Offer (谷雅丰帝国, 南京, guuu, 1013)
Offer (谷雅丰帝国, 南京, guuu, 1012)
Offer (谷雅丰帝国, 南京, guuu, 1011)
Offer (谷雅丰帝国, 南京, guuu, 1010)
}

。

query --city 南京 --position Up主 --sort-new --page -123



Ciallo~(∠ · ω<)∩★!

第 1 / 1 页 {

Offer (谷雅丰帝国, 南京, Up主, 1919810)
}

query --city 南京 --position Up主 --sort-new --page 999



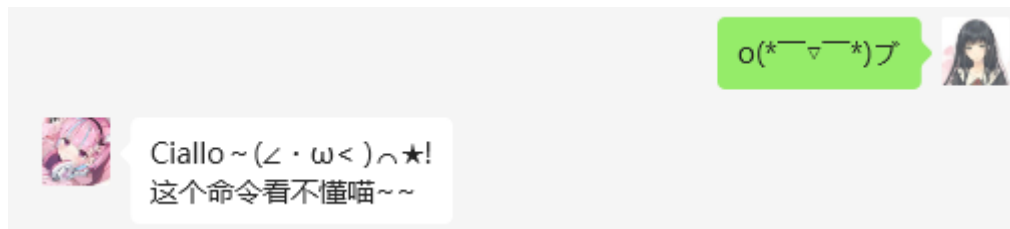
Ciallo~(∠ · ω<)∩★!

第 1 / 1 页 {

Offer (谷雅丰帝国, 南京, Up主, 1919810)
}

• 未知命令

。



- 爬虫检测
 - 如果一个用户在 **30s内提交了超过30条信息**，那么这个用户就会被永久添加到作弊用户/爬虫名单中。
 - 在数据库中储存一个队列，基于队列对用户提交记录进行判断。
 - 首先，查看用户状态，当前用户状态为正常（默认值 = 0）

■

| | |
|--------------------------|--|
| <input type="checkbox"/> | USER |
| <input type="checkbox"/> | User (oOBX96SvQKCjV5Zv2NOHNdnaG9P4, 0) |

- 然后，模拟爬虫，对系统进行刷屏

■



Ciallo~(∠ · ω<)∩★!
第 1 / 0 页 {
}

query --city 不存在 --position 为了避免系统响应消息太多



Ciallo~(∠ · ω<)∩★!
第 1 / 0 页 {
}

query --city 不存在 --position 为了避免系统响应消息太多



Ciallo~(∠ · ω<)∩★!
第 1 / 0 页 {
}

query --city 不存在 --position 为了避免系统响应消息太多



Ciallo~(∠ · ω<)∩★!
第 1 / 0 页 {
}

query --city 不存在 --position 为了避免系统响应消息太多



Ciallo~(∠ · ω<)∩★!
第 1 / 0 页 {
}

- 然后发现某个时刻，系统不再回应

■



Ciallo~(∠ · ω<)∩★!
第 1 / 0 页 {
}

query --city 不存在 --position 为了避免系统响应消息太多



query --city 不存在 --position 为了避免系统响应消息太多



query --city 不存在 --position 为了避免系统响应消息太多



query --city 不存在 --position 为了避免系统响应消息太多



query --city 不存在 --position 为了避免系统响应消息太多



query --city 不存在 --position 为了避免系统响应消息太多



- 进入后台，发现用户状态变成了爬虫

■



USER



User (oOBX96SvQKCjV5Zv2NOHNdnaG9P4, 2)

- 和预计一致，用户状态变成爬虫之后，系统会无视其所有请求

- 在后台，手动将用户状态修改回正常值 0

■

Change user

User (oOBX96SvQKCjV5Zv2NOHNdnaG9P4, 2)

Username:

State:

Request queue:

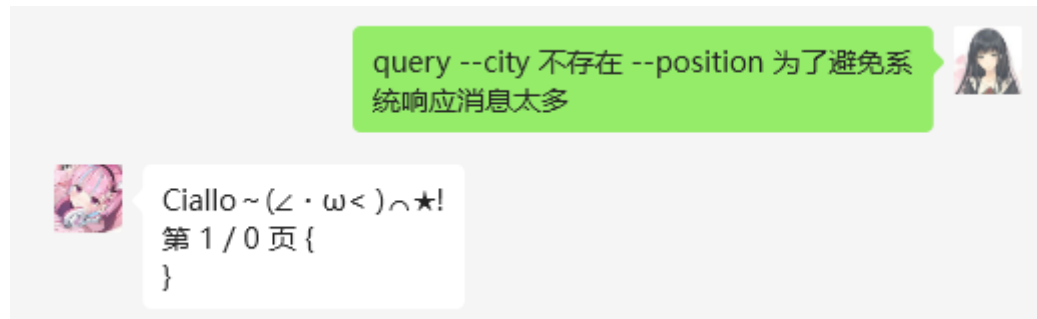
SAVE

Save and add another

Save and continue editing

✓ The user "User (oOBX96SvQKCjV5Zv2NOHNdnaG9P4, 0)" was changed successfully.

- 又可以重新接受信息



3. 关键代码

- 此处只列出部分代码，具体代码详见 `wechat_robot/main/`

3.1 路由

-

```
1 urlpatterns = [  
2     path('', wechat_main),  
3 ]
```

3.2 入口

-

```
1 @csrf_exempt  
2 def wechat_main(request):  
3     get_access_token()  
4     if request.method == 'GET':  
5         return wechat_heartbeat(request)  
6     else:  
7         return wechat_distributor(request)
```


3.3 词法分析器

- 入口

- ```
2 def resolve(content):
1 tokens = tokenize(content)
5 print(f"Command {content} was tokenized to {tokens if tokens else 'None'}")
1 return tokens
```

- 正则表达式

- ```
1 patterns = {
2     'help': {
3         'base': r'help',
4     },
5     'commit': {
6         'base': r'commit\s+(\S+)\s+(\S+)\s+(\S+)\s+(\d+)',
7     },
8     'query': {
9         'base'      : r'query',
10        'company'   : r'(?::--company|-co)\s+(\S+)',
11        'city'      : r'(?::--city|-ci)\s+(\S+)',
12        'position'  : r'(?::--position|-po)\s+(\S+)',
13        'page'      : r'(?::--page|-pa)\s+(\d+)',
14        'sort-new'   : r'(--sort-new)',
15        'sort-salary': r'(--sort-salary)',
16    },
17    'group-commit': {
18        'base': r'group-commit(?:\s+\S+)+',
19        'sub': r'(\S+)\s+(\S+)\s+(\S+)\s+(\d+)',
20    },
21 }
```

3.4 部分服务

- 服务列表

-

```

wechat_robot > main > services > _init__.py
20
19 from .offer_services import get_offer
18 from .offer_services import list_offers
17 from .offer_services import list_offers_with_page
16 from .offer_services import create_offer
15 from .offer_services import update_offer
14 from .offer_services import delete_offer
13 from .offer_services import replace_offer
12
11 from .offer_services import create_offers
10 from .offer_services import update_offers
9 from .offer_services import delete_offers
8 from .offer_services import replace_offers
7
6 from .user_services import get_user
5 from .user_services import create_user
4 from .user_services import get_user_state
3 from .user_services import check_user_state
2 from .user_services import reset_user_state
1 from .user_services import update_user_state
21

```

- get_offer()

- ```

107 def get_offer(args) -> Optional[Offer]:
106 '''查询匹配Filter的第一个Offer. 若不存在, 则返回 None'''
105
104 offers = list_offers(args)
103 if offers:
102 return offers[0]
101 else:
100 return None

```

- list\_offers()

- ```

98 def list_offers(args) -> List[Offer]:
97     '''查询匹配Filter的Offer集合'''
96
95     # Filter
94     filter = {key: args[key] for key in filter_keys if args[key] is not None}
93     query = Q()
92     for key, value in filter.items():
91         query &= Q(**{f"{key}__exact": value})
90     results = Offer.objects.filter(query)
89
88     # Sort
87     if args['sort-new'] is not None and args['sort-new']:
86         results = results.order_by('-datetime')
85     if args['sort-salary'] is not None and args['sort-salary']:
84         results = results.order_by('-salary')
83
82     # Return
81     return results

```

- create_offers()

-

```

1  def create_offers(args, username):
118  '''根据对应信息创建多个新的Offer'''
1
2      user = User.objects.filter(username=username).first()
3
4      for arg in args:
5          offer = Offer(
6              company=arg['company'],
7              city=arg['city'],
8              position=arg['position'],
9              salary=arg['salary'],
10             from_user=user,
11         )
12         offer.save()
13
14     return RETURN_STATE_SUCCESS

```

- 部分用户相关API

- ```

49 def get_user(username):
48 | return User.objects.filter(username=username).first()
47
46 def create_user(username):
45 | user = User(username=username)
44 | user.save()
43
42 | return RETURN_STATE_SUCCESS
41
40 def get_user_state(username):
39 | user = User.objects.filter(username=username).first()
38
37 | return user.state

```

- 反爬虫相关代码

- ```

5  def wechat_command_distributor(from_user, tokens):
4  |     if tokens is None:
3  |         return MESSAGE_UNKNOWN_COMMAND
2
1  |     if check_user_state(from_user, tokens['command'], update=True) != AUTHORITY_CHECK_PASS:
57  |         return 不返回任何消息
1

```
- ```

28 | if update:
29 | queue = eval(user.request_queue)
30 | cur_time = int(time.time())
31 | while len(queue) > 0 and cur_time - queue[0] >= REQUEST_QUEUE_LIMIT_TIME:
32 | queue.pop(0)
33 | queue.append(cur_time)
34 | if len(queue) >= REQUEST_QUEUE_LIMIT_SIZE:
35 | user.state |= USER_STATE_SPIDER
36 | user.request_queue = str(queue)
37 | user.save()
38
39 | if user.state & USER_STATE_BANNED != 0:
40 | return AUTHORITY_CHECK_FAILED
41
42 | if user.state & USER_STATE_SPIDER != 0:
43 | return AUTHORITY_CHECK_FAILED
44
45 | return AUTHORITY_CHECK_PASS

```