
pyweixin Documentation

0.0.1

ZWCZOU

2018 05 20

Contents

1	pyweixin	1
1.1	pyweixin package	1
	Python	7

1.1 pyweixin package

1.1.1 Submodules

1.1.2 pyweixin.client module

`pyweixin.client.debug_request` (*name, resp, start*)
curl

- **name** –
- **resp** – Response
- **start** –,

class `pyweixin.client.Client` (*path=""*)
`pyweixin.client._Client`

HTTP http

```
client = Client("http://example.com")
resp = client.v1.users.get(params={"user_id": 1})
print(resp.status_code)
print(resp.content)
```

.

```
resp = client.v1("to path").delete()
```

path – host

```
delete (*args, **kwargs)
    DELETE

get (*args, **kwargs)
    GET

post (*args, **kwargs)
    POST

put (*args, **kwargs)
    PUT
```

1.1.3 pyweixin.exceptions module

exception pyweixin.exceptions.**WeixinError** (*code, msg*)
exceptions.Exception

```
raise WeixinError("fail", "invalid args")
```

1.1.4 pyweixin.pay module

class pyweixin.pay.**WeixinPay** (*app_id, mch_id, mch_key, notify_url, refund_notify_url=None, key=None, cert=None*)
pyweixin.client.Client

```
from pyweixin.pay import WeixinPay

wp = WeixinPay('app id', 'mch id', 'mch key', 'notify url')

#
# https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=7_7&index=6
wp.jsapi("out_no", 1, "", "127.0.0.1", "wx.openid", attach="")

#
wp.order_query("out_no")

# url ``/pay/downloadfundflow``,
wp.pay.downloadfundflow.do(dict(bill_date="2018-04-10", account_type="Basic"))
```

- **app_id** – id
- **mch_id** – id
- **mch_key** – key, 32
- **notify_url** – , POST
- **refund_notify_url** –
- **key** – key,
- **cert** – cert,

```
FAIL = u'FAIL'
```

```
PAY_HOST = u'https://api.mch.weixin.qq.com'
```

```
SUCCESS = u'SUCCESS'
```

```
check (data)
```

```
    data -
```

```
close_order (out_trade_no, **kwargs)
```

```
    out_trade_no -
```

```
do (data, use_cert=False, app_id_Key=u'appid')
```

```
    appid, mch_id, nonce_str
```

- , sign
-
- xml dict
- `pyweixin.exceptions.WeixinError`

```
#
try:
    data = wp.pay.orderquery.do(dict(out_trade_no="trade_no"))
    print(data)
except pyweixin.exception.WeixinError:
    import traceback
    traceback.print_exc()
```

- data -
- use_cert -
- app_id_key - app_id key, app_id

```
download_bill (bill_date, bill_type=u'ALL', **kwargs)
```

```
https://pay.weixin.qq.com/wiki/doc/api/micropay.php?chapter=9_6
```

```
:
```

- 1.
2. 910
3. ""
- 4.

- bill_data -
- bill_type -

```
https://pay.weixin.qq.com/wiki/doc/api/micropay.php?chapter=9_6
```

```
get_read_pack_info (mch_billno, bill_type=u'MCHT', **kwargs)
```

```
https://pay.weixin.qq.com/wiki/doc/api/tools/cash_coupon.php?chapter=13_6&index=5
```

- bill_type - MCHT:
- mch_billno -

jsapi (*out_trade_no, total_fee, body, spbill_create_ip, openid, **kwargs*)

jsapi() *unified_order()*

https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=7_7&index=6

- **out_trade_no** -
- **trade_type** -
- **total_fee** -
- **body** -
- **spbill_create_ip** - ip
- **openid** - id

classmethod **nonce_str** ()

32

order_query (*out_trade_no=None, transaction_id=None, **kwargs*)

https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=9_2

:

-
-
- APIUSERPAYING
- API

- **out_trade_no** -
- **transaction_id** - ,

refund (*out_refund_no, total_fee, refund_fee, out_trade_no=None, transaction_id=None, **kwargs*)

https://pay.weixin.qq.com/wiki/doc/api/micropay.php?chapter=9_4

:

- 1.
- 2.
3. 150qps150, 6qps6
4. 50

- **out_trade_no** -
- **transaction_id** - ,
- **out_refund_no** -
- **total_fee** -
- **refund_fee** -
- **notify_url** - , , refund_notify_url

refund_query (*refund_id=None, out_refund_no=None, out_trade_no=None, transaction_id=None, **kwargs*)

203

https://pay.weixin.qq.com/wiki/doc/api/micropay.php?chapter=9_5

- **refund_id** – id
- **out_refund_no** –
- **out_trade_no** –
- **transaction_id** – ,

classmethod reply (*msg, ok=True*)

- **msg** –
- **ok** –

send_group_red_pack (*send_name, re_openid, total_amount, total_num, wishing, act_name, remark, amt_type=u'ALL_RANDOM', **kwargs*)

https://pay.weixin.qq.com/wiki/doc/api/tools/cash_coupon.php?chapter=13_5&index=4

- **send_name** –
- **re_openid** –
- **total_amount** –
- **wishing** –
- **amt_type** – , *ALL_RANDOM*
- **act_name** –
- **remark** –
- **total_num** – , 1
- **mch_billno** –
- **scene_id** –

send_red_pack (*send_name, re_openid, total_amount, wishing, client_ip, act_name, remark, **kwargs*)

https://pay.weixin.qq.com/wiki/doc/api/tools/cash_coupon.php?chapter=13_4&index=3

- **send_name** –
- **re_openid** –
- **total_amount** –
- **wishing** –
- **client_ip** – Ip
- **act_name** –
- **remark** –
- **total_num** – , 1

- `mch_billno` -
- `scene_id` -

`sign` (*data*)

data -

`classmethod` `to_dict` (*content*)

XMLdict,

```
WeixinPay.to_dict('<xml><a>123</a><b>345</b></xml>')
# {"a": "123", "b": "345"}
```

content - XML

`classmethod` `to_xml` (*data*)

XML

```
WeixinPay.to_xml(dict(a=123,b=234))
# <?xml version="1.0" encoding="utf-8"?><xml><a>123</a><b>234</b></xml>
```

data -

`unified_order` (*out_trade_no*, *trade_type*, *total_fee*, *body*, *spbill_create_ip*, ***kwargs*)

JSAPIAPP

https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=9_1

- `out_trade_no` -
- `trade_type` -
- `total_fee` -
- `body` -
- `spbill_create_ip` - ip
- `openid` - id, *trade_type* == *JSAPI*
- `product_id` - ID, *trade_type* == *NATIVE*

1.1.5 pyweixin.utils module

`class` `pyweixin.utils.dotdict`

`dict`

`dot`(.)

1.1.6 Module contents

p

pyweixin, 6
pyweixin.client, 1
pyweixin.exceptions, 2
pyweixin.pay, 2
pyweixin.utils, 6

C

check() (pyweixin.pay.WeixinPay), 3
close_order() (pyweixin.pay.WeixinPay), 3

D

delete() (pyweixin.client.Client), 1
do() (pyweixin.pay.WeixinPay), 3
download_bill() (pyweixin.pay.WeixinPay), 3

F

FAIL (pyweixin.pay.WeixinPay), 2

G

get() (pyweixin.client.Client), 2
get_read_pack_info() (pyweixin.pay.WeixinPay), 3

J

jsapi() (pyweixin.pay.WeixinPay), 3

O

order_query() (pyweixin.pay.WeixinPay), 4

P

PAY_HOST (pyweixin.pay.WeixinPay), 2
post() (pyweixin.client.Client), 2
put() (pyweixin.client.Client), 2
pyweixin (), 6
pyweixin.client (), 1
pyweixin.exceptions (), 2
pyweixin.pay (), 2
pyweixin.utils (), 6

R

refund() (pyweixin.pay.WeixinPay), 4
refund_query() (pyweixin.pay.WeixinPay), 4

S

send_group_red_pack() (pyweixin.pay.WeixinPay), 5
send_red_pack() (pyweixin.pay.WeixinPay), 5
sign() (pyweixin.pay.WeixinPay), 6

SUCCESS (pyweixin.pay.WeixinPay), 3

U

unified_order() (pyweixin.pay.WeixinPay), 6

W

WeixinError, 2