

---

# **python-bitget Documentation**

***Release 1.0.0.4***

**Cuongitl**

**Aug 06, 2023**



# CONTENTS

<b>1</b>	<b>Getting Started</b>	<b>3</b>
1.1	Installation . . . . .	3
1.2	Register on bitget . . . . .	3
1.3	Generate an API Key . . . . .	3
1.4	Initialise the client . . . . .	3
1.5	Making API Calls . . . . .	3
1.6	API Rate Limit . . . . .	4
1.7	Examples . . . . .	4
<b>2</b>	<b>Spot</b>	<b>7</b>
2.1	Public . . . . .	7
2.1.1	Get Server Time . . . . .	7
2.1.2	Get Coin List . . . . .	7
2.1.3	Get Symbols . . . . .	7
2.1.4	Get Single Symbol . . . . .	7
2.2	Market . . . . .	7
2.2.1	Get Single Ticker . . . . .	7
2.2.2	Get All Tickers . . . . .	8
2.2.3	Get Market Trades . . . . .	8
2.2.4	Get Candle Data . . . . .	8
2.2.5	Get Depth . . . . .	8
2.3	Wallet . . . . .	8
2.3.1	Transfer . . . . .	8
2.3.2	Sub Transfer . . . . .	8
2.3.3	Get Coin Address . . . . .	8
2.3.4	Withdraw . . . . .	8
2.3.5	Inner Withdraw . . . . .	9
2.3.6	Get Withdraw list . . . . .	9
2.3.7	Get Deposit List . . . . .	9
2.4	Account . . . . .	9
2.4.1	Get ApiKey Info . . . . .	9
2.4.2	Get Account Assets . . . . .	9
2.4.3	Get sub Account Spot Assets . . . . .	9
2.4.4	Get Bills . . . . .	9
2.4.5	Get Transfer List . . . . .	9
2.5	Trade . . . . .	10
2.5.1	Place order . . . . .	10
2.5.2	Batch order . . . . .	10
2.5.3	Cancel order . . . . .	10
2.5.4	Cancel order in batch (single instruments) . . . . .	10

2.5.5	Get order details	10
2.5.6	Get order List	10
2.5.7	Get order history	10
2.5.8	Get transaction details	10
2.5.9	Place plan order	11
2.5.10	Modify plan order	11
2.5.11	Cancel plan order	11
2.5.12	Get current plan orders	11
2.5.13	Get history plan orders	11
<b>3</b>	<b>Futures(USDT/Coin)</b>	<b>13</b>
3.1	Market	13
3.1.1	Get All Symbols	13
3.1.2	Get Depth	13
3.1.3	Get Single Symbol Ticker	13
3.1.4	Get All Symbol Ticker	13
3.1.5	Get Fills	13
3.1.6	Get Candle Data	14
3.1.7	Get Symbol Index Price	14
3.1.8	Get Symbol Next Funding Time	14
3.1.9	Get History Funding Rate	14
3.1.10	Get Current Funding Rate	14
3.1.11	Get Open Interest	14
3.1.12	Get Symbol Mark Price	14
3.1.13	Get Symbol Leverage	14
3.2	Account	15
3.2.1	Get Single Account	15
3.2.2	Get Account List	15
3.2.3	Get sub Account Contract Assets	15
3.2.4	Get Open Count	15
3.2.5	Change Leverage	15
3.2.6	Change Margin	15
3.2.7	Change Margin Mode	15
3.2.8	Change Hold Mode	15
3.2.9	Get Symbol Position	16
3.2.10	Get All Position	16
3.2.11	Get Account Bill	16
3.2.12	Get Business Account Bill	16
3.3	Trade	16
3.3.1	Place Order	16
3.3.2	Reversal	16
3.3.3	Batch Order	16
3.3.4	Cancel Order	17
3.3.5	Batch Cancel Order	17
3.3.6	Cancel All Order	17
3.3.7	Get Open Order	17
3.3.8	Get All Open Order	17
3.3.9	Get History Orders	17
3.3.10	Get ProductType History Orders	17
3.3.11	Get Order Details	17
3.3.12	Get Order fill detail	18
3.3.13	Get ProductType Order fill detail	18
3.3.14	Place Plan order	18
3.3.15	Modify Plan Order	18

3.3.16	Modify Plan Order TPSL	18
3.3.17	Place Stop Order	18
3.3.18	Place Trailing Stop Order	18
3.3.19	Place Position TPSL	19
3.3.20	Modify Stop Order	19
3.3.21	Cancel Plan Order (TPSL)	19
3.3.22	Cancel All trigger Order (TPSL)	19
3.3.23	Get Plan Order (TPSL) List	19
3.3.24	Get History Plan Orders (TPSL)	19
3.4	CopyTrade	19
3.4.1	Get Trader Open order	19
3.4.2	Get Follower Open Orders	19
3.4.3	Trader Close Position	20
3.4.4	Trader Modify TPSL	20
3.4.5	Get Traders History Orders	20
3.4.6	Get Trader Profit Summary	20
3.4.7	Get Trader History Profit Summary (according to settlement currency)	20
3.4.8	Get Trader History Profit Summary (according to settlement currency and date)	20
3.4.9	Get Trader History Profit Detail	20
3.4.10	Get Trader Profits Details	20
3.4.11	Get CopyTrade Symbols	21
3.4.12	Trader Change CopyTrade symbol	21
<b>4</b>	<b>Broker</b>	<b>23</b>
4.1	Sub Account Interface	23
4.1.1	Get Broker Info	23
4.1.2	Create Sub Account	23
4.1.3	Get Sub List	23
4.1.4	Modify Sub Account	23
4.1.5	Modify Sub Email	23
4.1.6	GET Sub Email	23
4.1.7	Get Sub Spot Assets	24
4.1.8	Get Sub Future Assets	24
4.1.9	Get Sub Deposit Address (Only Broker)	24
4.1.10	Sub Withdrawal (Only Broker)	24
4.1.11	Sub Deposit Auto Transfer (Only Broker)	24
4.2	Sub API Interface	24
4.2.1	Create Sub ApiKey (Only Broker)	24
4.2.2	Get Sub ApiKey List	24
4.2.3	Modify Sub ApiKey (Only Broker)	24
<b>5</b>	<b>WebSocketAPI</b>	<b>25</b>
5.1	Overview	25
5.2	Example connect	25
<b>6</b>	<b>Index</b>	<b>27</b>



bitget is a cryptocurrency derivatives exchange.

If you're new to Bitget, use the following link to [save 10% on all of your trade fees, and can get rewards up to \\$5005](#).

This is a wrapper around the Bitget API as described on Bitget, including all features the API provides using clear and readable objects, both for the REST as the websocket API.

Contents:



## GETTING STARTED

### 1.1 Installation

`python-bitget` is available on [PYPI](#). Install with `pip`:

```
pip install python-bitget
```

### 1.2 Register on bitget

Firstly register an account with bitget [save 10%](#) on all of your trade fee, and can get rewards up to \$5005.

### 1.3 Generate an API Key

To use signed account methods you are required to [create an API Key](#).

### 1.4 Initialise the client

Pass your API Key and Secret

```
api_key = "your-api-key"  
api_secret = "your-secret-key"  
api_passphrase = "your-api-passphrase"  
client = Client(api_key, api_secret, api_passphrase, use_server_time=False)
```

### 1.5 Making API Calls

Every method supports the passing of arbitrary parameters via keyword matching those in the [bitget API documentation](#). These keyword arguments will be sent directly to the relevant endpoint.

Each API method returns a dictionary of the JSON response as per the [bitget API documentation](#). The docstring of each method in the code references the endpoint it implements.

The bitget API documentation references a *timestamp* parameter, this is generated for you where required.

API Endpoints are rate limited by bitget, it's diff on per endpoint, ask them if you require more.

## 1.6 API Rate Limit

The Sub Account and Main Account have their own UIDs. Because the Limit Rules are based on each UID, every sub account has the same limit rate as main account. It's not necessary to worry about the limit rate for main account when you are trading with sub accounts Or their API Keys. And the limit rate is same for every user including VIP account. You may contact your Bitget business manager for more information.

Some calls have a higher weight than others especially if a call returns information about all symbols. Read the [official bitget documentation](#) for specific information.

## 1.7 Examples

Here are examples to access these

restAPI example

```
from bitget import Client

api_key = "your-api-key"
api_secret = "your-secret-key"
api_passphrase = "your-api-passphrase"

client = Client(api_key, api_secret, passphrase=api_passphrase)
result = client.mix_get_accounts(productType='UMCBL')
print(result)
```

websocketAPI example

```
from pybitget.stream import BitgetWsClient, SubscribeReq, handel_error

from pybitget.enums import *
from pybitget import logger

def on_message(message):
    logger.info(message)

api_key = "your-api-key"
api_secret = "your-secret-key"
api_passphrase = "your-api-passphrase"

if __name__ == '__main__':
    # Un-auth subscribe
    # client = BitgetWsClient() \
    #     .error_listener(handel_error) \
    #     .build()

    # Auth subscribe
    client = BitgetWsClient(api_key=api_key,
                           api_secret=api_secret,
                           passphrase=api_passphrase,
```

(continues on next page)

(continued from previous page)

```
        verbose=True) \  
    .error_listener(handel_error) \  
    .build()  
  
    # multi subscribe - Public Channels  
    channels = [SubscribeReq("mc", "ticker", "BTCUSD"), SubscribeReq("SP", "candle1W",  
↪ "BTCUSDT")]  
    client.subscribe(channels, on_message)  
  
    # single subscribe - # multi subscribe Public Channels  
    # channels = [SubscribeReq("mc", "ticker", "BTCUSD")]  
    # client.subscribe(channels, on_message)  
  
    # single subscribe - Order Channel - Private Channels  
    channels = [SubscribeReq(WS_CHANNEL_INSTTYPE, WS_PRIVATE_ORDERS_CHANNEL, WS_CHANNEL_  
↪ INSTID)]  
    client.subscribe(channels, on_message)
```



## 2.1 Public

### 2.1.1 Get Server Time

```
data = client.spot_get_server_time()
```

### 2.1.2 Get Coin List

```
data = client.spot_get_coin_list()
```

### 2.1.3 Get Symbols

```
data = client.spot_get_symbols()
```

### 2.1.4 Get Single Symbol

```
data = client.spot_get_symbol()
```

## 2.2 Market

### 2.2.1 Get Single Ticker

```
data = client.spot_get_ticker(self, symbol)
```

## 2.2.2 Get All Tickers

```
data = client.spot_get_tickers()
```

## 2.2.3 Get Market Trades

```
data = client.pot_get_market_trades(self, symbol, limit=100)
```

## 2.2.4 Get Candle Data

```
data = client.spot_get_candle_data(self, symbol, period, after='', before='', limit=100)
```

## 2.2.5 Get Depth

```
data = client.spot_get_depth(self, symbol, limit='150', type='step0')
```

## 2.3 Wallet

### 2.3.1 Transfer

```
data = client.spot_transfer(self, fromType, toType, amount, coin, clientOrderId=None)
```

### 2.3.2 Sub Transfer

```
data = client.spot_sub_transfer(self, fromType, toType, amount, coin, clientOrderId, ↵  
↵fromUserId, toUserId)
```

### 2.3.3 Get Coin Address

```
data = client.spot_get_depositAddress(self, coin, chain)
```

### 2.3.4 Withdraw

```
data = client.spot_withdrawal(self, coin, address, chain, amount, remark='', ↵  
↵clientOrderId=None, tag=None)
```

### 2.3.5 Inner Withdraw

```
data = client.spot_withdrawal_inner(self, coin, toUid, amount, clientOrderId=None)
```

### 2.3.6 Get Withdraw list

```
data = client.spot_get_withdrawallList(self, coin, startTime, endTime, pageSize=20,  
↳ pageNo=1)
```

### 2.3.7 Get Deposit List

```
data = client.spot_get_depositList(self, coin, startTime, endTime, pageSize=20, pageNo=1)
```

## 2.4 Account

### 2.4.1 Get ApiKey Info

```
data = client.spot_get_ApiKeyInfo()
```

### 2.4.2 Get Account Assets

```
data = client.spot_get_account_assets(self, coin=None)
```

### 2.4.3 Get sub Account Spot Assets

```
data = client.spot_get_sub_account_assets()
```

### 2.4.4 Get Bills

```
data = client.spot_get_bills(self, coinId='', groupId='', bizType='', after='', before=  
↳ '', limit=100)
```

### 2.4.5 Get Transfer List

```
data = client.spot_get_transfer_list(self, coinId='', fromType='', after='', before='',  
↳ limit=100)
```

## 2.5 Trade

### 2.5.1 Place order

```
data = client.spot_place_order(self, symbol, quantity, side, orderType, force, price='', ↵
↪ clientOrderId=None)
```

### 2.5.2 Batch order

```
data = client.spot_place_batch_orders(self, symbol, orderList)
```

### 2.5.3 Cancel order

```
data = client.spot_cancel_order(self, symbol, orderId)
```

### 2.5.4 Cancel order in batch (single instruments)

```
data = client.spot_cancel_batch_orders(self, symbol, orderIds)
```

### 2.5.5 Get order details

```
data = client.spot_get_order_details(self, symbol, orderId, clientOrderId=None)
```

### 2.5.6 Get order List

```
data = client.spot_get_open_orders(self, symbol='')
```

### 2.5.7 Get order history

```
data = client.spot_get_order_history(self, symbol, after='', before='', limit=100)
```

### 2.5.8 Get transaction details

```
data = client.spot_get_transaction_details(self, symbol='', orderId='', after='', before=
↪ '', limit=100)
```

### 2.5.9 Place plan order

```
data = client.spot_place_plan_order(self, symbol, side, triggerPrice, size, triggerType, ↵  
↵ orderType,  
                                  executePrice=None, timeInForceValue=None, clientId=None)
```

### 2.5.10 Modify plan order

```
data = client.spot_modify_plan_order(self, orderId, orderType, triggerPrice,  
                                     size=None, executePrice=None)
```

### 2.5.11 Cancel plan order

```
data = client.spot_cancel_plan_order(self, orderId)
```

### 2.5.12 Get current plan orders

```
data = client.spot_get_plan_orders(self, symbol, pageSize=20, lastEndId='')
```

### 2.5.13 Get history plan orders

```
data = client.spot_get_history_plan_orders(self, symbol, startTime, endTime, pageSize=20,  
↵ lastEndId='')
```



## FUTURES(USDT/COIN)

### 3.1 Market

#### 3.1.1 Get All Symbols

```
data = client.mix_get_symbols_info(productType)
```

#### 3.1.2 Get Depth

```
data = client.mix_get_depth(symbol, limit=100)
```

#### 3.1.3 Get Single Symbol Ticker

```
data = client.mix_get_single_symbol_ticker(symbol)
```

#### 3.1.4 Get All Symbol Ticker

```
data = client.mix_get_all_symbol_ticker(productType)
```

#### 3.1.5 Get Fills

```
# Get recent trades.  
data = client.mix_get_fills(symbol, limit=100)
```

### 3.1.6 Get Candle Data

```
data = client.mix_get_candles(symbol, granularity, startTime, endTime)
```

### 3.1.7 Get Symbol Index Price

```
data = client.mix_get_symbol_index_price(symbol)
```

### 3.1.8 Get Symbol Next Funding Time

```
data = client.mix_get_symbol_next_funding(symbol)
```

### 3.1.9 Get History Funding Rate

```
data = client.mix_get_history_fund_rate(symbol, pageSize=20, pageNo=1, nextPage=False)
```

### 3.1.10 Get Current Funding Rate

```
data = client.mix_get_current_fund_rate(symbol)
```

### 3.1.11 Get Open Interest

```
data = client.mix_get_open_interest(symbol)
```

### 3.1.12 Get Symbol Mark Price

```
data = client.mix_get_market_price(symbol)
```

### 3.1.13 Get Symbol Leverage

```
data = client.mix_get_leverage(symbol)
```

## 3.2 Account

### 3.2.1 Get Single Account

```
data = client.mix_get_account(symbol, marginCoin)
```

### 3.2.2 Get Account List

```
data = client.mix_get_accounts(productType)
```

### 3.2.3 Get sub Account Contract Assets

```
data = client.mix_get_sub_account_contract_assets(productType)
```

### 3.2.4 Get Open Count

```
data = client.mix_get_open_count(symbol, marginCoin, openPrice, openAmount, leverage=20)
```

### 3.2.5 Change Leverage

```
data = client.mix_adjust_leverage(symbol, marginCoin, leverage, holdSide=None)
```

### 3.2.6 Change Margin

```
data = client.mix_adjust_margin(symbol, marginCoin, amount, holdSide=None)
```

### 3.2.7 Change Margin Mode

```
data = client.mix_adjust_margintype(symbol, marginCoin, marginMode)
```

### 3.2.8 Change Hold Mode

```
data = client.mix_adjust_hold_mode(productType, holdMode)
```

### 3.2.9 Get Symbol Position

```
data = client.mix_get_single_position(symbol, marginCoin=None)
```

### 3.2.10 Get All Position

```
data = client.mix_get_all_positions(productType, marginCoin=None)
```

### 3.2.11 Get Account Bill

```
data = client.mix_get_accountBill(symbol, marginCoin, startTime, endTime, lastEndId='',  
↳ pageSize=20, next=False)
```

### 3.2.12 Get Business Account Bill

```
data = client.mix_get_accountBusinessBill(productType, startTime, endTime, lastEndId='',  
↳ pageSize=20, next=False)
```

## 3.3 Trade

### 3.3.1 Place Order

```
data = client.mix_place_order(symbol, marginCoin, size, side, orderType,  
                             price='', clientOrderId=None, reduceOnly=False,  
                             timeInForceValue='normal', presetTakeProfitPrice='',  
↳ presetStopLossPrice='')
```

### 3.3.2 Reversal

```
data = client.mix_reversal(symbol, marginCoin, side, orderType,  
                           size=None, clientOrderId=None, timeInForceValue='normal', reverse=False)
```

### 3.3.3 Batch Order

```
data = client.mix_batch_orders(symbol, marginCoin, orderDataList)
```

### 3.3.4 Cancel Order

```
data = client.mix_cancel_order(symbol, marginCoin, orderId)
```

### 3.3.5 Batch Cancel Order

```
data = client.mix_batch_cancel_orders(symbol, marginCoin, orderIds)
```

### 3.3.6 Cancel All Order

```
data = client.mix_cancel_all_orders(productType, marginCoin)
```

### 3.3.7 Get Open Order

```
data = client.mix_get_open_order(symbol)
```

### 3.3.8 Get All Open Order

```
data = client.mix_get_all_open_orders(productType, marginCoin=None)
```

### 3.3.9 Get History Orders

```
data = client.mix_get_history_orders(symbol, startTime, endTime, pageSize, lastEndId='',  
↪ isPre=False)
```

### 3.3.10 Get ProductType History Orders

```
data = client.mix_get_productType_history_orders(productType, startTime, endTime,  
↪ pageSize, lastEndId='', isPre=False)
```

### 3.3.11 Get Order Details

```
data = client.mix_get_order_details(symbol, orderId=None, clientOrderId=None)
```

### 3.3.12 Get Order fill detail

```
data = client.mix_get_order_fill_detail(symbol, orderId=None, startTime=None, ↵
↵ endTime=None, lastEndId=None)
```

### 3.3.13 Get ProductType Order fill detail

```
data = client.mix_get_productType_order_fill_detail(productType, startTime=None, ↵
↵ endTime=None, lastEndId=None)
```

### 3.3.14 Place Plan order

```
data = client.mix_place_plan_order(symbol, marginCoin, size, side, orderType, ↵
↵ triggerPrice, triggerType
                                , executePrice=None, clientOrderId=None, ↵
↵ presetTakeProfitPrice=None, presetStopLossPrice=None, reduceOnly=False)
```

### 3.3.15 Modify Plan Order

```
data = client.mix_modify_plan_order(symbol, marginCoin, orderId, orderType, triggerPrice,
↵ triggerType
                                , executePrice=None)
```

### 3.3.16 Modify Plan Order TPSL

```
data = client.mix_modify_plan_order_tpsl(symbol, marginCoin, orderId
                                , presetTakeProfitPrice=None, presetStopLossPrice=None)
```

### 3.3.17 Place Stop Order

```
data = client.mix_place_stop_order(symbol, marginCoin, triggerPrice, planType, holdSide,
                                triggerType='fill_price', size=None, rangeRate=None)
```

### 3.3.18 Place Trailing Stop Order

```
data = client.mix_place_trailing_stop_order(symbol, marginCoin, triggerPrice, side,
                                triggerType=None, size=None, rangeRate=None)
```

### 3.3.19 Place Position TPSL

```
data = client.mix_place_PositionsTPSL(symbol, marginCoin, planType, triggerPrice, ↵  
↵triggerType, holdSide=None)
```

### 3.3.20 Modify Stop Order

```
data = client.mix_modify_stop_order(symbol, marginCoin, orderId, triggerPrice, planType)
```

### 3.3.21 Cancel Plan Order (TPSL)

```
data = client.mix_cancel_plan_order(symbol, marginCoin, orderId, planType)
```

### 3.3.22 Cancel All trigger Order (TPSL)

```
data = client.mix_cancel_all_trigger_orders(productType, planType)
```

### 3.3.23 Get Plan Order (TPSL) List

```
data = client.mix_get_plan_order_tpsl(symbol=None, productType=None, isPlan=None)
```

### 3.3.24 Get History Plan Orders (TPSL)

```
data = client.mix_get_history_plan_orders(symbol, startTime, endTime, pageSize=100, ↵  
↵lastEndId=None, isPre=False, isPlan=None)
```

## 3.4 CopyTrade

### 3.4.1 Get Trader Open order

```
data = client.mix_get_cp_open_order(symbol, productType, pageSize=20, pageNo=1)
```

### 3.4.2 Get Follower Open Orders

```
data = client.mix_get_cp_follower_open_orders(symbol, productType, pageSize=20, pageNo=1)
```

### 3.4.3 Trader Close Position

```
data = client.mix_cp_close_position(symbol, trackingNo)
```

### 3.4.4 Trader Modify TP/SL

```
data = client.mix_cp_modify_tp_sl(symbol, trackingNo, stopProfitPrice=None, ↵  
↵stopLossPrice=None)
```

### 3.4.5 Get Traders History Orders

```
data = client.mix_get_cp_history_orders(startTime, endTime, pageSize=20, pageNo=1)
```

### 3.4.6 Get Trader Profit Summary

```
data = client.mix_get_cp_profit_summary()
```

### 3.4.7 Get Trader History Profit Summary (according to settlement currency)

```
data = client.mix_get_cp_profit_settle_margin_coin()
```

### 3.4.8 Get Trader History Profit Summary (according to settlement currency and date)

```
data = client.mix_get_cp_profit_date_group(pageSize=20, pageNo=1)
```

### 3.4.9 Get Trader History Profit Detail

```
data = client.mix_get_cp_profit_date_detail(marginCoin, date, pageSize=20, pageNo=1)
```

### 3.4.10 Get Trader Profits Details

```
data = client.mix_get_cp_wait_profit_detail(pageSize=20, pageNo=1)
```

### 3.4.11 Get CopyTrade Symbols

```
data = client.mix_get_cp_symbols()
```

### 3.4.12 Trader Change CopyTrade symbol

```
data = client.mix_cp_change_symbol(symbol, operation)
```



## 4.1 Sub Account Interface

### 4.1.1 Get Broker Info

```
data = client.broker_get_info()
```

### 4.1.2 Create Sub Account

```
data = client.broker_sub_create(subName, remark=None)
```

### 4.1.3 Get Sub List

```
data = client.broker_get_sub_list(pageSize=10, lastEndId=None, status=None)
```

### 4.1.4 Modify Sub Account

```
data = client.broker_sub_modify_account(subUid, perm, status)
```

### 4.1.5 Modify Sub Email

```
data = client.broker_sub_modify_email(subUid, subEmail):
```

### 4.1.6 GET Sub Email

```
data = client.broker_get_sub_email(subUid)
```

### 4.1.7 Get Sub Spot Assets

```
data = client.broker_get_sub_spot_assets(subUid)
```

### 4.1.8 Get Sub Future Assets

```
data = client.broker_get_sub_future_assets(subUid, productType)
```

### 4.1.9 Get Sub Deposit Address (Only Broker)

```
data = client.broker_get_sub_deposit_address(subUid, coin, chain=None)
```

### 4.1.10 Sub Withdrawal (Only Broker)

```
data = client.broker_sub_withdrawal(subUid, coin, address, chain, amount,  
                                     tag=None, clientId=None, remark=None)
```

### 4.1.11 Sub Deposit Auto Transfer (Only Broker)

```
data = client.broker_sub_auto_transfer(subUid, coin, toAccountType)
```

## 4.2 Sub API Interface

### 4.2.1 Create Sub ApiKey (Only Broker)

```
data = client.broker_sub_create_api(subUid, passphrase, remark, ip, perm)
```

### 4.2.2 Get Sub ApiKey List

```
data = client.broker_get_sub_api_list(subUid)
```

### 4.2.3 Modify Sub ApiKey (Only Broker)

```
data = client.broker_sub_modify_api(subUid, apikey, remark=None, ip=None, perm=None)
```

## WEBSOCKETAPI

### 5.1 Overview

This is a wrapper around the Bitget API as described on Bitget, so please read the [official documents](#) for more details.

### 5.2 Example connect

Pass your API Key and Secret

```
from pybitget.stream import BitgetWsClient, SubscribeReq, handel_error

from pybitget.enums import *
from pybitget import logger

def on_message(message):
    logger.info(message)

api_key = "your-api-key"
api_secret = "your-secret-key"
api_passphrase = "your-api-passphrase"

if __name__ == '__main__':
    # Un-auth subscribe
    # client = BitgetWsClient() \
    #     .error_listener(handel_error) \
    #     .build()

    # Auth subscribe
    client = BitgetWsClient(api_key=api_key,
                           api_secret=api_secret,
                           passphrase=api_passphrase,
                           verbose=True) \
        .error_listener(handel_error) \
        .build()

    # multi subscribe - Public Channels
    channels = [SubscribeReq("mc", "ticker", "BTCUSD"), SubscribeReq("SP", "candle1W",
```

(continues on next page)

(continued from previous page)

```
↪ "BTCUSD"]
    client.subscribe(channels, on_message)

    # single subscribe -      # multi subscribe Public Channels
    # channels = [SubscribeReq("mc", "ticker", "BTCUSD")]
    # client.subscribe(channels, on_message)

    # single subscribe - Order Channel - Private Channels
    channels = [SubscribeReq(WS_CHANNEL_INSTTYPE, WS_PRIVATE_ORDERS_CHANNEL, WS_CHANNEL_
↪ INSTID)]
    client.subscribe(channels, on_message)
```

**INDEX**

- genindex