

Data Pullers: Requirements

The api must pull data from a range of different crypto exchanges. The scope can be discussed but looking for the top 10 exchanges. The pullers should be written in Golang. We are looking for a fast and scalable solution. There should be **five** types of API's we are looking for, each pulling from a multiple of different exchanges:

The data we are looking to pull is:

1. Aggregated Spot Wallet Balances (current)
2. Aggregated Derivative Wallet Balances
3. Derivative Collateral Balances
4. Derivative Positions Data
5. Daily Wallet Balances (historic)

1. Spot Wallet Balances Puller

The following outlines how to read the wallet balances from example exchanges and accounts.

1. Read User ID

`uuid = 12345`

User ID should be used to identify the respective keys for the given account.

2. Define start and end time for historical balance query

```
end_time = int(datetime.datetime.now().timestamp() * 1000)
start_time = end_time - (86400000 * 30) # 30 days in milliseconds
```

3. Set API KEYS - Ideally this should be pulled from some db - using account id

This section should read all the available keys for the respective exchange(s). There are two types of keys, secret and api. An example below:

```
accountXYZ_apiKey = "YOUR_API_KEY"
accountXYZ_secretKey = "YOUR_API_SECRET"
```

4. Set up instance with exchange(s)

An example below is given for how to set up an instance with Binance and Bybit (two different exchanges) using the ccxt library in Python:

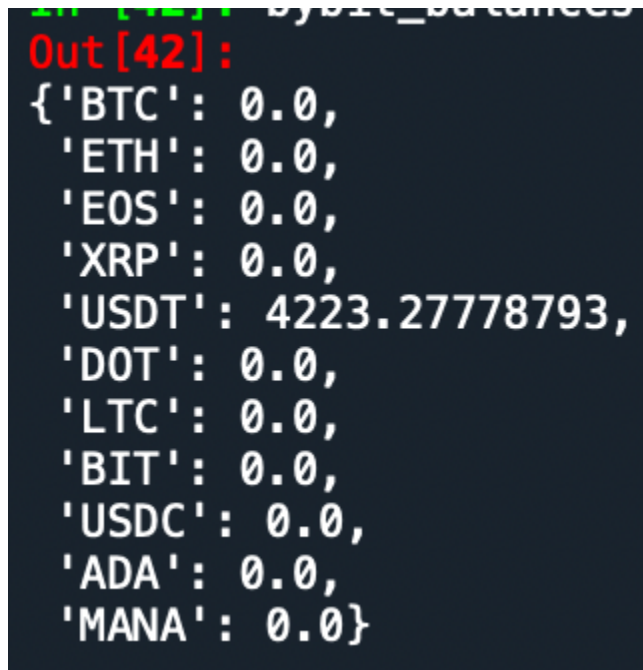
```
# Set up Binance exchange instance
binance = ccxt.binance({
    'apiKey': accountXYZ_apiKey,
    'secret': 'accountXYZ_apiKey',
})
```

```
# Set up Bybit exchange instance
bybit = ccxt.binance({
    'apiKey': 'apiKey',
    'secret': 'secretKey',
})
```

5. Get list of currencies in account balance

```
binance_balances = binance.fetch_balance()['total']
```

This should provide a list like this:



The screenshot shows the output of a Jupyter Notebook cell, labeled 'Out[42]:'. It displays a dictionary representing account balances for various cryptocurrencies. The keys are currency codes in uppercase, and the values are floating-point numbers representing the balance. The 'USDT' balance is notably higher than the others.

```
Out[42]:
{'BTC': 0.0,
 'ETH': 0.0,
 'EOS': 0.0,
 'XRP': 0.0,
 'USDT': 4223.27778793,
 'DOT': 0.0,
 'LTC': 0.0,
 'BIT': 0.0,
 'USDC': 0.0,
 'ADA': 0.0,
 'MANA': 0.0}
```

6. The outputted data should contain all non zero balances and a time stamp.

2. Derivative Wallet Balances Puller

The below illustrates the desired mapping for Binance and Bybit Derivatives accounts wallet balances. The same approach with regards to signing the api with an api and secret key. The output name field may be empty for some of the fields, but the api should be standardised such

that all fields are present in the output. We also include the data type. The same as in 1 applies in terms of key management.

Output Name	Binance (fapi)	Bybit (Inverse Perpetual / USDT Perpetual)
account_type (varchar 255)	"USDT margin"	"perpetual-"+coin
total_collateral (decimal)	totalMarginBalance	equity
total_initial_margin (decimal)	totalInitialMargin	used_margin
total_maintenance_margin (decimal)	totalMaintMargin	EMPTY
unrealized_pl (decimal)	totalUnrealizedProfit	unrealised_pnl
realized_pl (decimal)	EMPTY	cum_realised_pnl
total_pl (decimal)	EMPTY	unrealised_pn+cum_realised_pnl
total_available (decimal)	availableBalance	available_balance
currency (varchar 255)	"USDT"	coin
risk_ratio (decimal)	EMPTY	EMPTY
status (varchar 255)	EMPTY	EMPTY
options_delta (decimal)	EMPTY	EMPTY
options_gamma (decimal)	EMPTY	EMPTY
options_theta (decimal)	EMPTY	EMPTY
options_vega (decimal)	EMPTY	EMPTY
request	GET /fapi/v2/account	GET /v2/private/wallet/balance
api link	https://binance-docs.github.io/apidocs/futures/en/#account-information-v2-user_data	https://bybit-exchange.github.io/docs/inverse/?console#t-balance

3. Derivative Collateral Balances Puller

	Binance (fapi)	Bybit (Inverse Perpetual / USDT Perpetual)
account_type (varchar 255)	"USDT margin"	"perpetual-"+coin
asset (varchar 255)	asset	coin
balance (decimal)	walletBalance	wallet_balance
available (decimal)	MaxWithdrawalAmount	available_balance
contract_type (varchar 255)	"future"	"perpetual"
request	GET /fapi/v2/balance	GET /v2/private/wallet/balance
api link	https://binance-docs.github.io/apidocs/futures/en/#futures-account-balance-v2-user_data	https://bybit-exchange.github.io/docs/inverse/?console#t-balance

4. Positions Puller

Positions Should be mapped as per the below. Here we include different mappings depending on whether the rest api or websocket is used.

Output Name	Binance (fapi)	
contract (varchar 255)	symbol	P.s
contract_type (varchar 255)	"future"	"future"
position_side (varchar 255)	positionSide	P.ps
margin (varchar 255)	marginType	P.mt
leverage (decimal)	leverage	Not updated
balance (decimal)	positionAmt	P.pa
contract_currency (varchar 255)	instruments.base_currency	instruments.base_currency

average_price (decimal)	entryPrice	P.ep
liquidation_price (decimal)	liquidationPrice	Not updated
initial_margin (decimal)	account.positions[symbol].initialMar gin	Not updated
maintenance_margin (decimal)	account.position[s ymbol].maintMarg in	Not updated
unrealized_pl (decimal)	unRealizedProfit	P.up
realized_pl (decimal)	EMPTY	EMPTY
total_pl (decimal)	EMPTY	EMPTY
delta (decimal)	EMPTY	EMPTY
gamma (decimal)	EMPTY	EMPTY
theta (decimal)	EMPTY	EMPTY
vega (decimal)	EMPTY	EMPTY
is_active(boolean)		
Comment		
request	GET /fapi/v2/positionRisk	ACCOUNT_UPDATE
api link	https://binance-docs.github.io/apidocs/futures/en/#position-information-v2-user_data	
websocket		https://binance-docs.github.io/apidocs/futures/en/#event-balance-and-position-update
request2 (to join via positions: symbol + positionSide) + ref = account	GET /fapi/v2/account	
api link2	https://binance-docs.github.io/apidocs/futures/en/#account-information-v2-user_data	

	la	
--	----	--

Bybit Mappings Positions

Same as the above, just for the various bybit accounts.

Output Name	Bybit (Inverse Perpetual)	Bybit (USDT Perpetual)	Bybit (Invers Future)
contract (varchar 255)	symbol	symbol	data.symbol
contract_type (varchar 255)	"perpetual"	"perpetual"	"future-coin-m"
position_side (varchar 255)	side	side	if data.side: data.side else 'pos'
margin (varchar 255)	ONLY REST: is_isolated === 'true' ? "Isolated" : "Cross"	ONLY REST: is_isolated === 'true' ? "Isolated" : "Cross"	ONLY REST: data.is_isolated === 'true' ? "Isolated" : "Cross"
leverage (decimal)	leverage	leverage	data.leverage
balance (decimal)	size * (if side = "Sell", then multiply by (-1), otherwise by (1))	size * (if side = "Sell", then multiply by (-1), otherwise by (1))	data.size * (if side = "Sell", then multiply by (-1), otherwise by (1))
contract_currency (varchar 255)	instrument.marginAss et	instrument.marginAs set	instrument.marginAsset
average_price (decimal)	entry_price	entry_price	data.entry_price
liquidation_price (decimal)	liq_price	liq_price	data.liq_price
initial_margin (decimal)	EMPTY	EMPTY	
maintenance_margin (decimal)	EMPTY	EMPTY	
unrealized_pl (decimal)	ONLY REST: unrealised_pnl ELSE EMPTY	ONLY REST: unrealised_pnl ELSE EMPTY	data.unrealised_pnl
realized_pl (decimal)	cum_realised_pnl	cum_realised_pnl	data.cum_realised_pnl
total_pl (decimal)	unrealised_pnl+cum_r ealised_pnl	unrealised_pnl+cum _realised_pnl	data.unrealised_pnl + data.cum_realised_pnl
delta (decimal)	EMPTY	EMPTY	EMPTY
gamma (decimal)	EMPTY	EMPTY	EMPTY
theta (decimal)	EMPTY	EMPTY	EMPTY

vega (decimal)	EMPTY	EMPTY	EMPTY
is_active(booleam)			
Comment			
request	GET /v2/private/position/list	GET /private/linear/position/list	GET /futures/private/position/list
api link	https://bybit-exchange.github.io/docs/inverse/?console#t-myposition	https://bybit-exchange.github.io/docs/linear/#t-myposition	https://bybit-exchange.github.io/docs/inverse_futures/#t-myposition
websocket	websocket:position	websocket:position	{"op": "subscribe", "args": ["position"]}
request2 (to join via positions: symbol + positionSide) + ref = account	https://bybit-exchange.github.io/docs/inverse/#t-websocketposition	https://bybit-exchange.github.io/docs/linear/#t-websocketposition	https://bybit-exchange.github.io/docs/inverse_futures/#t-websocketposition

5. Daily Wallet Balances (historic)

We can pull daily historical data from exchanges. This may vary for each exchange, but for Binance it is the last 30 days. We want an api that pulls **all** available data. Just as in 1. We are interested in the total balance by coin for coin with balance > 0.

Instantiate the Binance client with your API credentials

```
client = Client('YOUR_API_KEY', 'YOUR_API_SECRET')
```

Define the type of wallet ("SPOT", "MARGIN", "FUTURES")

```
wallet_type = 'SPOT'
```

Define the timestamp for which you want the snapshot (for example, 1 day ago)

```
timestamp = int((time.time() - 60*60*24) * 1000) # in milliseconds
```

Get daily account snapshot for the specified timestamp

```
snapshots = client.get_account_snapshot(type=wallet_type,
timestamp=timestamp)
```

Parameters:

Parameters:

Name	Type	Mandatory	Description
type	STRING	YES	"SPOT", "MARGIN", "FUTURES"
startTime	LONG	NO	
endTime	LONG	NO	
limit	INT	NO	min 7, max 30, default 7
recvWindow	LONG	NO	
timestamp	LONG	YES	

- The query time period must be less than 30 days
- Support query within the last one month only
- If startTime and endTime not sent, return records of the last 7 days by default

Response:**Response:**

```
{
  "code":200, // 200 for success; others are error codes
  "msg":""," // error message
  "snapshotVos":[
    {
      "data":{
        "balances":[
          {
            "asset":"BTC",
            "free":"0.09905021",
            "locked":"0.00000000"
          },
          {
            "asset":"USDT",
            "free":"1.89109409",
            "locked":"0.00000000"
          }
        ],
        "totalAssetOfBtc":"0.09942700"
      },
      "type":"spot",
      "updateTime":1576281599000
    }
  ]
}
```