GAINS HTTP API for Freqtrade/CCXT compatibility

GET Order (`fetch_order`)

Request Params

Required

- `id`

Optional

- `pair` (if order IDs are not globally unique but only unique per trading pair)

Response

An order object e.g.

```
{
    'id':                '12345-67890:09876/54321', // string
    'timestamp':          1502962946216, // order placing/opening Unix
timestamp in milliseconds
    'status':      'open',        // 'open', 'closed', 'canceled', 'expired',
'rejected'
    'symbol':      'ETH/BTC',     // symbol
    'type':        'limit',       // 'market', 'limit'
    'timeInForce': 'GTC',         // 'GTC', 'IOC', 'FOK', 'PO'
    'side':        'buy',         // 'buy', 'sell'
    'price':        0.06917684,   // float price in quote currency (may be
empty for market orders)
    'average':      0.06917684,   // float average filling price
    'amount':       1.5,          // ordered amount of base currency
    'filled':       1.1,          // filled amount of base currency
    'remaining':    0.4,          // remaining amount to fill
    'cost':         0.076094524,  // 'filled' * 'price' (filling price used
where available)
    'trades':     [ ... ],        // a list of order trades/executions
    'fee': {                      // fee info, if available
        'currency': 'BTC',        // which currency the fee is (usually quote)
        'cost': 0.0009,           // the fee amount in that currency
        'rate': 0.002,            // the fee rate (if available)
    },
}
```

Notes:

1. If orders are always either completely filled or rejected, then fields like `amount`, `filled`, and `remaining` can be omitted.

GET Orders (`fetch_orders`)

Params

Optional

- `pair` (filter by pair; return all orders if no pair provided)
- `since` (filter orders after a specific Unix timestamp; can define a default period like past 24 hours if no timestamp is provided)
- `limit` (number of most recent results to fetch)

Response

List of orders.

POST Order (`create_order`)

Params

Required

- `pair`
- `side`: can be `buy` or `sell`
- `type`: can be `market` or `limit`
- `amount`: amount to trade in base currency of the pair

Optional

- `price`: limit price in quote currency (ignored for market orders)

Response

An order object.

DELETE Order (`cancel_order`)

Params

Required

- `id`

Optional

- `pair` (if order IDs are not globally unique but only unique per trading pair)

Response

The order object with status "canceled" or error.

GET Balance (`fetch_balance`)

> **Confirmation needed**: Since GAINS assumes self-custody, this method is probably not implementable. If so, we need to emulate the response by quering relevant wallets on our end.

Params

No params.

Response

A balance structure e.g.

```
{
    'timestamp': 1499280391811, // Unix Timestamp in milliseconds (seconds * 
1000)

    //--------------------------------------------------------------------------
    // indexed by availability of funds first, then by currency

    'free':  {           // money, available for trading, by currency
        'BTC': 321.00,   // floats...
        'USD': 123.00,
        ...
    },

    'used':  { ... },    // money on hold, locked, frozen, or pending, by 
currency

    'total': { ... },    // total (free + used), by currency

    'debt': { ... },     // debt, by currency

    //--------------------------------------------------------------------------
    // indexed by currency first, then by availability of funds

    'BTC':   {           // string, three-letter currency code, uppercase
        'free': 321.00   // float, money available for trading
        'used': 234.00,  // float, money on hold, locked, frozen or pending
```

```
        'total': 555.00, // float, total balance (free + used)
    },

    'USD':   {           // ...
        'free': 123.00   // ...
        'used': 456.00,
        'total': 579.00,
    },

    ...
}
```

GET OHLCV (`fetch_ohlcv`)

> **Confirmation needed**: Can GAINS return volume?

Params

Required

- `pair`
- `timeframe` e.g. `1m`, `5m`, `30m`, `1h`, `1d`

Optional

- `since` (filter candles after a specific Unix timestamp; use a default period for each timeframe if no timestamp is provided)
- `limit` (number of most recent candles to fetch; use a default limit if none is provided)

Response

```
[
    [
        1504541580000, // UTC timestamp in milliseconds, integer
        4235.4,        // (O)pen price, float
        4240.6,        // (H)ighest price, float
        4230.0,        // (L)owest price, float
        4230.7,        // (C)losing price, float
        37.72941911    // (V)olume float in base currency
    ],
    ...
]
```

GET Ticker (`fetch_ticker`)

Params

Required

- `pair`

Response

Last 24 hour information about the pair

```
{
    'symbol':        string symbol of the market ('BTC/USD', 'ETH/BTC', ...)
    'timestamp':     int (64-bit Unix Timestamp in milliseconds since Epoch 1
Jan 1970)
    'high':          float, // highest price
    'low':           float, // lowest price
    'bid':           float, // current best bid (buy) price
    'bidVolume':     float, // current best bid (buy) amount (may be missing or
undefined)
    'ask':           float, // current best ask (sell) price
    'askVolume':     float, // current best ask (sell) amount (may be missing
or undefined)
    'vwap':          float, // volume weighed average price
    'open':          float, // opening price
    'close':         float, // price of last trade (closing price for current
period)
    'previousClose': float, // closing price for the previous period
    'baseVolume':    float, // volume of base currency traded for last 24 hours
    'quoteVolume':   float, // volume of quote currency traded for last 24
hours
}
```

Notes:

- Since Gains doesn't have an order book, bid and ask may be undefined. In that case, it is OK to omit them.
- If Gains doesn't have volume information, it is also OK to omit volume related fields.

GET Leverage Tiers (`fetch_leverage_tiers`)

Params

None

Response

```
{
  "BTC/USDT:USDT":
    [
      {
```

```
            "tier": 1,                        // tier index
            "notionalCurrency": "USDT",       // the currency that minNotional and
maxNotional are in
            "minNotional": 0,                 // the lowest amount of this tier //
stake = 0.0
            "maxNotional": 10000,             // the highest amount of this tier
// max stake amount at 75x leverage = 133.33333333333334
            "maintenanceMarginRate": 0.0065, // maintenance margin rate
            "maxLeverage": 75,                // max available leverage for this
market when the value of the trade is > minNotional and < maxNotional
        },
        {
            "tier": 2,
            "notionalCurrency": "USDT",
            "minNotional": 10000,             // min stake amount at 50x leverage
= 200.0
            "maxNotional": 50000,             // max stake amount at 50x leverage
= 1000.0
            "maintenanceMarginRate": 0.01,
            "maxLeverage": 50,
        },
        ...
        {
            "tier": 9,
            "notionalCurrency": "USDT",
            "minNotional": 20000000,
            "maxNotional": 50000000,
            "maintenanceMarginRate": 0.5,
            "maxLeverage": 1,
        },
    ]
    ...
}
```

Notes:

1. If maintenance margin rate is not a thing in Gains, can omit this field.

POST Leverage (`set_leverage`)

Params

Required

- `pair`
- `leverage`

Optional

- `long_leverage`

- `short_leverage`

Response

```
{
    "symbol": "BTC/USDT:USDT",  // unified market symbol
    "longLeverage": 100,        // the set leverage for a long position
    "shortLeverage": 75,        // the set leverage for a short position
}
```

GET Trades (`fetch_my_trades`)

Params

Required

None

Optional

- `symbol`
- `order_id` (to find matching trades for an order)
- `since`
- `limit`

Response

List of trades of the form

```
{
    'id':           '12345-67890:09876/54321',  // string trade id
    'timestamp':    1502962946216,              // Unix timestamp in
milliseconds
    'symbol':       'ETH/BTC',                  // symbol
    'order':        '12345-67890:09876/54321',  // string order id or
undefined/None/null
    'type':         'limit',                    // order type, 'market',
'limit' or undefined/None/null
    'side':         'buy',                      // direction of the trade,
'buy' or 'sell'
    'takerOrMaker': 'taker',                    // string, 'taker' or 'maker'
    'price':        0.06917684,                 // float price in quote
currency
    'amount':       1.5,                        // amount of base currency
    'cost':         0.10376526,                 // total cost, `price *
amount`,
    'fee':          {
```

```
        'cost':  0.0015,                        // float
        'currency': 'ETH',                      // usually base currency for
buys, quote currency for sells
        'rate': 0.002,                          // the fee rate (if available)
    },
    'fees': [                                   // an array of fees if paid in
multiple currencies
        {
            'cost':  0.0015,                    // float
            'currency': 'ETH',                  // usually base currency for
buys, quote currency for sells
            'rate': 0.002,                      // the fee rate (if available)
        },
    ],
}
```

Note:

1. The distinction between taker and maker may not be applicable