OANDA REST-V20 API Documentation

Release 0.6.3

Feite Brekeveld

oandapyV20 REST-V20 API wrapper

1	Introduction 1.1 Install	3 3 3
2	Interface OANDA's REST-V20 2.1 The client	5 5 8 8
3	oandapyV20.endpoints3.1oandapyV20.endpoints.accounts	11 11 21 35 43 51 57 63 71
4	oandapyV20.definitions 4.1 oandapyV20.definitions.accounts	79 79 81 83 87 88
5	oandapyV20.types 1 5.1 AccountID 5.2 AccountUnits 5.2 AccountUnits 5.3 ClientComment 5.4 ClientID 5.5 ClientTag 5.5 ClientTime 5.6 DateTime 5.7 OrderID 5.8 OrderIdentifier 5.9 OrderSpecifier 5.9 OrderSpecifier	104 104 104 104 105 105

	5.10	PriceValue	106					
	5.11	TradeID	106					
	5.12	Units	107					
6	oand	npyV20.contrib	109					
	6.1	ApyV20.contrib Factories	109					
	6.2	Generic						
	6.3	Order Classes						
	6.4	support classes	121					
7	Exan	ples	127					
	7.1	Example for trades-endpoints	127					
8	Indic	Indices and tables						
Ру	Python Module Index							

Contents:

CHAPTER 1

Introduction

The oandapyV20 package offers an API to the OANDA V20 REST service. To use the REST-API-service you will need a *token* and an *account*. This applies for both *live* and *practice* accounts. For details check oanda.com.

1.1 Install

Install the pypi package with pip:

```
$ pip install oandapyV20
```

Or alternatively install the latest development version from github:

```
$ pip install git+https://github.com/hootnot/oanda-api-v20.git
```

You may consider using *virtualenv* to create isolated Python environments. Python 3.4 has *pyvenv* providing the same kind of functionality.

1.2 Download from Github

If you want to run the tests, download the source from github:

```
$ git clone https://github.com/hootnot/oanda-api-v20.git
$ cd oanda-api-v20
$ python setup.py test
$ python setup.py install
```

Interface OANDA's REST-V20

2.1 The client

The oandapyV20 package contains a client class, oandapyV20.API, to communicate with the REST-V20 interface. It processes requests that can be created from the endpoint classes. For it's communication it relies on: requests (requests).

The client keeps no state of a requests. The response of a request is assigned to the request instance. The response is also returned as a return value by the client.

```
class oandapy V20.API (access_token, environment='practice', headers=None, request_params=None)

Bases: object
```

API - class to handle APIRequests objects to access API endpoints.

Examples

```
# get a list of trades
from oandapyV20 import API
import oandapyV20.endpoints.trades as trades

api = API(access_token="xxx")
accountID = "101-305-3091856-001"

r = trades.TradesList(accountID)
# show the endpoint as it is constructed for this call
print("REQUEST:{}".format(r))
rv = api.request(r)
print("RESPONSE:\n{}".format(json.dumps(rv, indent=2)))
```

Output:

```
REQUEST: v3/accounts/101-305-3091856-001/trades
RESPONSE:
"trades": [
   {
      "financing": "0.0000",
      "openTime": "2016-07-21T15:47:05.170212014Z",
      "price": "10133.9",
      "unrealizedPL": "8.0000",
      "realizedPL": "0.0000",
      "instrument": "DE30_EUR",
      "state": "OPEN",
      "initialUnits": "-10",
      "currentUnits": "-10",
      "id": "1032"
    },
      "financing": "0.0000",
      "openTime": "2016-07-21T15:47:04.963590941Z",
      "price": "10134.4",
      "unrealizedPL": "13.0000",
      "realizedPL": "0.0000",
      "instrument": "DE30_EUR",
      "state": "OPEN",
      "initialUnits": "-10",
      "currentUnits": "-10",
      "id": "1030"
   }
  ],
  "lastTransactionID": "1040"
```

```
# reduce a trade by it's id
from oandapyV20 import API
import oandapyV20.endpoints.trades as trades

api = API(access_token="...")

accountID = "101-305-3091856-001"
tradeID = "1030"
cfg = {"units": 5}
r = trades.TradeClose(accountID, tradeID=tradeID, data=cfg)
# show the endpoint as it is constructed for this call
print("REQUEST:{}".format(r))
rv = api.request(r)
print("RESPONSE\n{}".format(json.dumps(rv, indent=2)))
```

Output:

```
REQUEST:v3/accounts/101-305-3091856-001/trades/1030/close
RESPONSE: {
    "orderFillTransaction": {
        "orderID": "1041",
        "financing": "-0.1519",
        "instrument": "DE30_EUR",
        "userID": 1435156,
        "price": "10131.6",
        "tradeReduced": {
```

```
"units": "5",
    "financing": "-0.1519",
    "realizedPL": "14.0000",
    "tradeID": "1030"
  "batchID": "1041",
  "accountBalance": "44876.2548",
  "reason": "MARKET_ORDER_TRADE_CLOSE",
  "time": "2016-07-21T17:32:51.361464739Z",
  "units": "5",
  "type": "ORDER_FILL",
  "id": "1042",
  "pl": "14.0000",
  "accountID": "101-305-3091856-001"
},
"orderCreateTransaction": {
  "timeInForce": "FOK",
  "positionFill": "REDUCE_ONLY",
  "userID": 1435156,
  "batchID": "1041",
  "instrument": "DE30_EUR",
  "reason": "TRADE_CLOSE",
  "tradeClose": {
    "units": "5",
    "tradeID": "1030"
  },
  "time": "2016-07-21T17:32:51.361464739Z",
  "units": "5",
  "type": "MARKET_ORDER",
  "id": "1041",
  "accountID": "101-305-3091856-001"
"relatedTransactionIDs": [
 "1041",
  "1042"
],
"lastTransactionID": "1042"
```

__init__ (access_token, environment='practice', headers=None, request_params=None)
Instantiate an instance of OandaPy's API wrapper.

Parameters

- access_token (string) Provide a valid access token.
- **environment** (*string*) Provide the environment for OANDA's REST api. Valid values: 'practice' or 'live'. Default: 'practice'.
- headers (dict (optional)) Provide request headers to be set for a request.

Note: There is no need to set the 'Content-Type: application/json' for the endpoints that require this header. The API-request classes covering those endpoints will take care of the header.

request_params [(optional)] parameters to be passed to the request. This can be used to apply for instance a timeout value:

2.1. The client 7

```
request_params={"timeout": 0.1}
```

See specs of the requests module for full details of possible parameters.

Warning: parameters belonging to a request need to be set on the requestinstance and are NOT passed via the client.

```
request (endpoint)
```

Perform a request for the APIRequest instance 'endpoint'.

Parameters endpoint (APIRequest) – The endpoint parameter contains an instance of an APIRequest containing the endpoint, method and optionally other parameters or body data.

Raises V20Error in case of HTTP response code >= 400

request_params

request_params property.

2.2 Exceptions

```
class oandapyV20.V20Error(code, msg)
    Bases: exceptions.Exception
```

Generic error class.

In case of HTTP response codes >= 400 this class can be used to raise an exception representing that error.

```
__init__ (code, msg)
Instantiate a V20Error.
```

Parameters

- code (int) the HTTP-code of the response
- msg(str) the message returned with the response

2.3 Logging

The oandapyV20 package has *logging* integrated. Logging can be simply applied by enabling a *logger*. The example below will log INFO-level logging to the file v20.log. For details check the logger module in the standard Python documentation.

```
# code snippet
from oandapyV20 import API
import oandapyV20.endpoints.orders as orders
from oandapyV20.exceptions import V20Error
from exampleauth import exampleAuth
import logging

logging.basicConfig(
    filename="v20.log",
    level=logging.INFO,
    format='%(asctime)s [%(levelname)s] %(name)s : %(message)s',
)
```

```
accountID, token = exampleAuth()
...
```

Resulting loglines:

```
2016-10-22 17:50:37,988 [INFO] oandapyV20.oandapyV20 : setting up API-client for environment practice
2016-10-22 17:50:37,990 [INFO] oandapyV20.oandapyV20 : performing request https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/orders
2016-10-22 17:50:37,998 [INFO] requests.packages.urllib3.connectionpool : Starting enew HTTPS connection (1): api-fxpractice.oanda.com
2016-10-22 17:50:38,866 [INFO] oandapyV20.oandapyV20 : performing request https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/orders
2016-10-22 17:50:39,066 [ERROR] oandapyV20.oandapyV20 : request https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/orders failed [400,{
environment practice.oanda.com/v3/accounts/101-004-1435156-001/orders failed [400,{
environment practice.oa
```

2.3. Logging 9

CHAPTER 3

oandapyV20.endpoints

3.1 oandapyV20.endpoints.accounts

3.1.1 AccountChanges

```
 \textbf{class} \  \, \texttt{oandapyV20.endpoints.accounts.AccountChanges} \, (\textit{accountID}, \textit{params=None}) \\ \text{Bases:} \, \texttt{oandapyV20.endpoints.accounts.Accounts} \, . \\
```

AccountChanges.

Endpoint used to poll an Account for its current state and changes since a specified TransactionID.

```
ENDPOINT = 'v3/accounts/{accountID}/changes'
EXPECTED_STATUS = 200

METHOD = 'GET'
__init__(accountID, params=None)
    Instantiate an AccountChanges request.
```

Parameters

- account ID (string (required)) id of the account to perform the request on.
- params (dict (optional)) query params to send, check developer.oanda.com for details.

Query Params example:

```
{
  "sinceTransactionID": 2308
}
```

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.accounts as accounts
```

```
>>> client = oandapyV20.API(access_token=...)
>>> params = ...
>>> r = accounts.AccountChanges(accountID=..., params=params)
>>> client.request(r)
>>> print r.response
```

Output:

```
"state": {
 "trades": [],
 "marginCloseoutNAV": "33848.2663",
 "marginUsed": "0.0000",
 "marginAvailable": "33848.2663",
  "marginCallPercent": "0.00000",
 "NAV": "33848.2663",
  "marginCloseoutMarginUsed": "0.0000",
  "orders": [],
  "withdrawalLimit": "33848.2663",
  "marginCloseoutPercent": "0.00000",
  "positions": [],
  "unrealizedPL": "0.0000",
  "marginCallMarginUsed": "0.0000",
  "marginCloseoutUnrealizedPL": "0.0000",
  "positionValue": "0.0000"
},
"changes": {
 "tradesReduced": [],
  "tradesOpened": [],
  "ordersFilled": [],
  "tradesClosed": [],
  "transactions": [
      "price": "1.20000",
      "stopLossOnFill": {
       "timeInForce": "GTC",
        "price": "1.22000"
      "timeInForce": "GTC",
      "reason": "CLIENT_ORDER",
      "id": "2309",
      "batchID": "2309",
      "triggerCondition": "TRIGGER_DEFAULT",
      "positionFill": "DEFAULT",
      "userID": 1435156,
      "instrument": "EUR_USD",
      "time": "2016-10-25T21:07:21.065554321Z",
      "units": "-100",
      "type": "LIMIT_ORDER",
      "accountID": "101-004-1435156-001"
    }
  ],
  "ordersCreated": [
    {
      "partialFill": "DEFAULT_FILL",
      "price": "1.20000",
      "stopLossOnFill": {
```

```
"timeInForce": "GTC",
        "price": "1.22000"
     },
      "timeInForce": "GTC",
      "createTime": "2016-10-25T21:07:21.065554321Z",
      "triggerCondition": "TRIGGER_DEFAULT",
      "positionFill": "POSITION_DEFAULT",
      "id": "2309",
      "instrument": "EUR_USD",
      "state": "PENDING",
      "units": "-100",
      "type": "LIMIT"
   }
 ],
  "positions": [],
 "ordersTriggered": [],
  "ordersCancelled": []
"lastTransactionID": "2309"
```

3.1.2 AccountConfiguration

```
class oandapyV20.endpoints.accounts.AccountConfiguration(accountID, data)
    Bases: oandapyV20.endpoints.accounts.Accounts
```

Set the client-configurable portions of an Account.

Parameters

- accountID (string (required)) id of the account to perform the request on.
- data (dict (required)) json body to send

body example:

```
{
   "marginRate": "0.05"
}
```

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.accounts as accounts
>>> client = oandapyV20.API(access_token=...)
>>> r = accounts.AccountConfiguration(accountID, data=data)
>>> client.request(r)
>>> print r.response
```

```
{
  "lastTransactionID": "830",
  "clientConfigureTransaction": {
    "userID": 1435156,
    "marginRate": "0.05",
    "batchID": "830",
    "time": "2016-07-12T19:48:11.657494168Z",
    "type": "CLIENT_CONFIGURE",
    "id": "830",
    "accountID": "101-004-1435156-001"
  }
}
```

3.1.3 AccountDetails

class oandapyV20.endpoints.accounts.AccountDetails(accountID)
 Bases: oandapyV20.endpoints.accounts.Accounts

AccountDetails.

Get the full details for a single Account that a client has access to. Full pending Order, open Trade and open Position representations are provided.

```
ENDPOINT = 'v3/accounts/{accountID}'
EXPECTED_STATUS = 200
METHOD = 'GET'
__init__(accountID)
    Instantiate an AccountDetails request.
```

Parameters accountID(string (required))—id of the account to perform the request on.

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.accounts as accounts
>>> client = oandapyV20.API(access_token=...)
>>> r = accounts.AccountDetails(accountID)
>>> client.request(r)
>>> print r.response
```

```
"pl": "-3.8046"
    },
    "instrument": "EUR_USD",
    "resettablePL": "-3.8046",
    "pl": "-3.8046"
  },
    "short": {
      "unrealizedPL": "682.0000",
      "units": "-20",
      "resettablePL": "-1744.8000",
      "tradeIDs": [
       "821",
        "823"
      ],
      "averagePrice": "9984.7",
      "pl": "-1744.8000"
    "unrealizedPL": "682.0000",
    "long": {
      "units": "0",
      "resettablePL": "447.6000",
      "unrealizedPL": "0.0000",
      "pl": "447.6000"
    "instrument": "DE30_EUR",
    "resettablePL": "-1297.2000",
    "pl": "-1297.2000"
  }
],
"unrealizedPL": "682.0000",
"marginCloseoutNAV": "49393.6580",
"marginUsed": "9948.9000",
"currency": "EUR",
"resettablePL": "-1301.0046",
"NAV": "49377.6580",
"marginCloseoutMarginUsed": "9949.8000",
"id": "101-004-1435156-001",
"marginCloseoutPositionValue": "198996.0000",
"openTradeCount": 2,
"orders": [
    "partialFill": "DEFAULT_FILL",
    "price": "0.87000",
    "stopLossOnFill": {
      "timeInForce": "GTC",
      "price": "0.88000"
    "timeInForce": "GTC",
    "clientExtensions": {
      "comment": "myComment",
      "id": "myID"
    },
    "id": "204",
    "triggerCondition": "TRIGGER_DEFAULT",
    "replacesOrderID": "200",
    "positionFill": "POSITION_DEFAULT",
```

```
"createTime": "2016-07-08T07:18:47.623211321Z",
      "instrument": "EUR GBP",
      "state": "PENDING",
      "units": "-50000",
      "type": "LIMIT"
  ],
  "openPositionCount": 1,
  "marginCloseoutPercent": "0.10072",
  "marginCallMarginUsed": "9949.8000",
  "hedgingEnabled": false,
  "positionValue": "198978.0000",
  "pl": "-1301.0046",
  "lastTransactionID": "833",
  "marginAvailable": "39428.7580",
  "marginRate": "0.05",
  "marginCallPercent": "0.20144",
  "pendingOrderCount": 1,
  "withdrawalLimit": "39428.7580",
  "trades": [
      "instrument": "DE30_EUR",
      "financing": "0.0000",
      "openTime": "2016-07-12T09:32:18.062823776Z",
      "initialUnits": "-10",
      "currentUnits": "-10",
      "price": "9984.7",
      "unrealizedPL": "341.0000",
      "realizedPL": "0.0000",
      "state": "OPEN",
      "id": "821"
    },
      "instrument": "DE30_EUR",
      "financing": "0.0000",
      "openTime": "2016-07-12T09:32:18.206929733Z",
      "initialUnits": "-10",
      "currentUnits": "-10",
      "price": "9984.7",
      "unrealizedPL": "341.0000",
      "realizedPL": "0.0000",
      "state": "OPEN",
      "id": "823"
    }
  ],
  "alias": "hootnotv20",
  "createdByUserID": 1435156,
  "marginCloseoutUnrealizedPL": "698.0000",
  "createdTime": "2016-06-24T21:03:50.914647476Z",
  "balance": "48695.6580"
},
"lastTransactionID": "833"
```

3.1.4 AccountInstruments

class oandapyV20.endpoints.accounts.AccountInstruments(accountID, params=None)
 Bases: oandapyV20.endpoints.accounts.Accounts

AccountInstruments.

Get the list of tradable instruments for the given Account. The list of tradeable instruments is dependent on the regulatory division that the Account is located in, thus should be the same for all Accounts owned by a single user.

```
ENDPOINT = 'v3/accounts/{accountID}/instruments'
EXPECTED_STATUS = 200

METHOD = 'GET'
__init__(accountID, params=None)
    Instantiate an AccountInstruments request.
```

Parameters

- accountID (string (required)) id of the account to perform the request on.
- params (dict (optional)) query params to send, check developer.oanda.com for details.

Query Params example:

```
{
   "instruments": "EU50_EUR,EUR_USD,US30_USD,FR40_EUR,EUR_CHF,DE30_EUR"
}
```

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.accounts as accounts
>>> client = oandapyV20.API(access_token=...)
>>> params = ...
>>> r = accounts.AccountInstruments(accountID=..., params=params)
>>> client.request(r)
>>> print r.response
```

Output:

```
"instruments": [
  {
    "marginRate": "0.05",
    "minimumTrailingStopDistance": "5.0",
    "maximumPositionSize": "0",
    "minimumTradeSize": "1",
    "displayName": "Europe 50",
    "name": "EU50_EUR",
    "displayPrecision": 1,
    "maximumTrailingStopDistance": "10000.0",
    "maximumOrderUnits": "3000",
    "tradeUnitsPrecision": 0,
    "pipLocation": 0,
    "type": "CFD"
  },
  {
```

```
"marginRate": "0.05",
  "minimumTrailingStopDistance": "0.00050",
  "maximumPositionSize": "0",
  "minimumTradeSize": "1",
  "displayName": "EUR/USD",
  "name": "EUR_USD",
  "displayPrecision": 5,
  "maximumTrailingStopDistance": "1.00000",
  "maximumOrderUnits": "100000000",
  "tradeUnitsPrecision": 0,
  "pipLocation": -4,
  "type": "CURRENCY"
},
 "marginRate": "0.05",
 "minimumTrailingStopDistance": "5.0",
  "maximumPositionSize": "0",
  "minimumTradeSize": "1",
  "displayName": "US Wall St 30",
  "name": "US30_USD",
  "displayPrecision": 1,
  "maximumTrailingStopDistance": "10000.0",
  "maximumOrderUnits": "1000",
  "tradeUnitsPrecision": 0,
 "pipLocation": 0,
 "type": "CFD"
}.
  "marginRate": "0.05",
  "minimumTrailingStopDistance": "5.0",
  "maximumPositionSize": "0",
  "minimumTradeSize": "1",
  "displayName": "France 40",
  "name": "FR40_EUR",
  "displayPrecision": 1,
  "maximumTrailingStopDistance": "10000.0",
  "maximumOrderUnits": "2000",
 "tradeUnitsPrecision": 0,
 "pipLocation": 0,
  "type": "CFD"
},
 "marginRate": "0.05",
  "minimumTrailingStopDistance": "0.00050",
  "maximumPositionSize": "0",
  "minimumTradeSize": "1",
  "displayName": "EUR/CHF",
  "name": "EUR CHF",
  "displayPrecision": 5,
  "maximumTrailingStopDistance": "1.00000",
  "maximumOrderUnits": "100000000",
  "tradeUnitsPrecision": 0,
  "pipLocation": -4,
  "type": "CURRENCY"
},
  "marginRate": "0.05",
```

```
"minimumTrailingStopDistance": "5.0",
    "maximumPositionSize": "0",
    "minimumTradeSize": "1",
    "displayName": "Germany 30",
    "name": "DE30_EUR",
    "displayPrecision": 1,
    "maximumTrailingStopDistance": "10000.0",
    "maximumOrderUnits": "2500",
    "tradeUnitsPrecision": 0,
    "pipLocation": 0,
    "type": "CFD"
    }
},
"lastTransactionID": "2124"
```

3.1.5 AccountList

 $\begin{tabular}{ll} \textbf{class} & \texttt{oandapyV20.endpoints.accounts.AccountList} \\ \textbf{Bases:} & \texttt{oandapyV20.endpoints.accounts.Accounts} \end{tabular}$

Get a list of all Accounts authorized for the provided token.

```
ENDPOINT = 'v3/accounts'
EXPECTED_STATUS = 200
METHOD = 'GET'
__init__()
    Instantiate an AccountList request.
```

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.accounts as accounts
>>> client = oandapyV20.API(access_token=...)
>>> r = accounts.AccountList()
>>> client.request(r)
>>> print r.response
```

3.1.6 AccountSummary

```
class oandapyV20.endpoints.accounts.AccountSummary (accountID)
    Bases: oandapyV20.endpoints.accounts.Accounts
    Get a summary for a single Account that a client has access to.
    ENDPOINT = 'v3/accounts/{accountID}/summary'
    EXPECTED STATUS = 200
    METHOD = 'GET'
    __init__(accountID)
         Instantiate an AccountSummary request.
            Parameters account ID (string (required)) – id of the account to perform the request
         >>> import oandapyV20
         >>> import oandapyV20.endpoints.accounts as accounts
         >>> client = oandapyV20.API(access_token=...)
         >>> r = accounts.AccountSummary(accountID)
         >>> client.request(r)
         >>> print r.response
           "account": {
             "marginCloseoutNAV": "35454.4740",
             "marginUsed": "10581.5000",
            "currency": "EUR",
            "resettablePL": "-13840.3525",
            "NAV": "35454.4740",
             "marginCloseoutMarginUsed": "10581.5000",
             "marginCloseoutPositionValue": "211630.0000",
             "openTradeCount": 2,
             "id": "101-004-1435156-001",
             "hedgingEnabled": false,
             "marginCloseoutPercent": "0.14923",
             "marginCallMarginUsed": "10581.5000",
             "openPositionCount": 1,
             "positionValue": "211630.0000",
             "pl": "-13840.3525",
             "lastTransactionID": "2123",
             "marginAvailable": "24872.9740",
             "marginRate": "0.05",
             "marginCallPercent": "0.29845",
             "pendingOrderCount": 0,
             "withdrawalLimit": "24872.9740",
             "unrealizedPL": "0.0000",
             "alias": "hootnotv20",
             "createdByUserID": 1435156,
             "marginCloseoutUnrealizedPL": "0.0000",
             "createdTime": "2016-06-24T21:03:50.914647476Z",
             "balance": "35454.4740"
           },
           "lastTransactionID": "2123"
```

3.2 oandapyV20.endpoints.forexlabs

3.2.1 Autochartist

```
class oandapyV20.endpoints.forexlabs.Autochartist (params=None)
    Bases: oandapyV20.endpoints.forexlabs.ForexLabs
Autochartist.
Get the 'autochartist data'.
ENDPOINT = 'labs/v1/signal/autochartist'
EXPECTED_STATUS = 200
METHOD = 'GET'
__init__ (params=None)
    Instantiate an Autochartist request.
```

Parameters params (dict (optional)) - query params to send, check developer.oanda.com for details.

```
>>> r = labs.Autochartist(params=params)
>>> client.request(r)
>>> print(r.response)
```

Output:

```
"signals": [
   "data": {
      "points": {
        "support": {
         "y1": 0.72456,
          "y0": 0.725455,
          "x0": 1520420400,
          "x1": 1520503200
        },
        "resistance": {
          "y1": 0.729755,
          "y0": 0.731095,
          "x0": 1520323200,
          "x1": 1520463600
      "patternendtime": 1520589600,
      "prediction": {
```

```
"pricelow": 0.7316,
        "timefrom": 1520589600,
        "pricehigh": 0.7349,
        "timeto": 1520773200
      }
    },
    "meta": {
      "direction": 1,
      "completed": 1,
      "probability": 72.36,
      "scores": {
        "clarity": 7,
        "breakout": 10,
        "quality": 8,
        "initialtrend": 10,
        "uniformity": 6
      "pattern": "Channel Down",
      "historicalstats": {
        "hourofday": {
          "total": 1909,
          "percent": 71.08,
          "correct": 1357
        },
        "pattern": {
          "total": 3361,
          "percent": 73.61,
          "correct": 2474
        },
        "symbol": {
          "total": 429,
          "percent": 65.5,
          "correct": 281
        }
      },
      "interval": 60,
      "trendtype": "Continuation",
      "length": 73
    },
    "type": "chartpattern",
    "id": 458552738,
    "instrument": "NZD_USD"
  }
],
"provider": "autochartist"
```

3.2.2 Calendar

```
class oandapyV20.endpoints.forexlabs.Calendar(params)
    Bases: oandapyV20.endpoints.forexlabs.ForexLabs
    Calendar.
    Get calendar information.
    ENDPOINT = 'labs/v1/calendar'
```

```
EXPECTED_STATUS = 200

METHOD = 'GET'

__init__ (params)

Instantiate a Calendar request.
```

Parameters params (dict (required)) - query params to send, check developer.oanda.com for details.

```
>>> r = labs.Calendar(params=params)
>>> client.request(r)
>>> print(r.response)
```

Output:

```
[
    "impact": 3,
    "currency": "USD",
    "actual": "60.8",
    "market": "58.7",
    "title": "ISM Manufacturing",
    "timestamp": 1519916400,
    "region": "americas",
    "previous": "59.1",
    "unit": "index",
    "forecast": "59.5"
}
```

3.2.3 CommitmentsOfTraders

Parameters params (dict (required)) – query params to send, check developer.oanda.com for details.

```
>>> r = labs.CommitmentOfTraders(params=params)
>>> client.request(r)
>>> print(r.response)
```

Output:

```
"EUR_USD": [
 {
   "oi": "603460",
    "price": "1.2315925",
    "ncs": "109280",
    "ncl": "258022",
    "date": 1517288400,
    "unit": "Contracts Of EUR 125,000"
  },
    "oi": "596937".
    "price": "1.2364",
    "ncs": "110546",
   "ncl": "251369",
    "date": 1517893200,
    "unit": "Contracts Of EUR 125,000"
  },
   "oi": "564233",
   "price": "1.2330275",
   "ncs": "103496",
    "ncl": "230785",
    "date": 1518498000,
    "unit": "Contracts Of EUR 125,000"
  },
    "oi": "567534",
    "price": "1.2346025",
    "ncs": "103147",
    "ncl": "229273",
    "date": 1519102800,
    "unit": "Contracts Of EUR 125,000"
  },
    "oi": "567463",
    "price": "1.23557",
    "ncs": "100310",
    "ncl": "238287",
    "date": 1519707600,
```

```
"unit": "Contracts Of EUR 125,000"
}
]
```

3.2.4 HistoricalPositionRatios

```
class oandapyV20.endpoints.forexlabs.HistoricalPositionRatios(params)
    Bases: oandapyV20.endpoints.forexlabs.ForexLabs
```

HistoricalPositionRatios.

Get the historical position ratios for an instrument.

```
ENDPOINT = 'labs/v1/historical_position_ratios'
EXPECTED_STATUS = 200
METHOD = 'GET'
__init__(params)
    Instantiate a HistoricalPositionRatios request.
```

Parameters params (dict (required)) - query params to send, check developer.oanda.com for details.

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.forexlabs as labs
>>> accountID = ...
>>> client = oandapyV20.API(access_token=...)
>>> params =
{
    "instrument": "EUR_USD",
    "period": 86400
}
```

```
>>> r = labs.HistoricalPositionRatios(params=params)
>>> client.request(r)
>>> print(r.response)
```

Output:

```
1519927200,
  44.16,
  1.2212
],
  1519928400,
  44.2,
  1.2209
],
[
  1519929601,
 43.88,
 1.2201
],
[
  1519930800,
  44.15,
  1.2197
  1519932000,
 44.51,
  1.2204
],
  1519933200,
  44.55,
  1.2233
],
  1519934401,
  44.55,
  1.2254
],
 1519935601,
 44.08,
  1.226
],
  1519936801,
  43.67,
  1.2264
],
  1519938001,
 43.55,
  1.2263
],
  1519939201,
 43.25,
  1.2261
],
  1519940401,
  43.28,
```

```
1.2263
],
  1519941601,
  43.39,
  1.2267
],
  1519942801,
 43.69,
  1.227
],
  1519944001,
  43.57,
  1.2269
],
  1519945201,
 43.68,
  1.2272
],
  1519946400,
 43.51,
 1.2268
],
  1519947601,
  43.53,
  1.2267
  1519948801,
 43.71,
  1.2271
],
  1519950001,
  43.66,
  1.2265
],
  1519951201,
 43.78,
  1.2269
  1519952401,
  43.86,
  1.2273
  1519953600,
  43.85,
  1.2273
```

```
[
  1519954800,
  43.81,
  1.2271
  1519956001,
  44,
 1.2275
],
  1519957200,
 43.89,
  1.2274
],
  1519958401,
 43.95,
  1.2273
  1519959601,
  43.93,
  1.2273
],
  1519960800,
 43.86,
  1.2276
],
  1519962000,
 44.02,
 1.2278
],
[
 1519963200,
  44.18,
  1.228
],
[
  1519964401,
 44.52,
  1.2283
],
  1519965600,
  44.19,
  1.2281
  1519966801,
  44.14,
  1.2278
],
  1519968000,
```

```
43.93,
  1.2276
],
  1519969201,
 43.82,
 1.2277
],
 1519970401,
 43.77,
  1.2279
],
 1519971601,
 43.02,
 1.2269
],
  1519972801,
 42.99,
 1.2265
],
 1519974001,
 42.73,
 1.2263
],
 1519975201,
 42.22,
 1.2262
],
 1519976400,
 42.13,
  1.2255
],
 1519977601,
 42.02,
 1.2263
],
 1519978801,
 42.15,
 1.2261
 1519980000,
 42.5,
 1.2273
],
  1519981201,
 42.2,
  1.2274
```

```
],
[
  1519982400,
  42.06,
 1.2271
],
  1519983600,
 42.38,
  1.2279
],
 1519984800,
 42.29,
 1.2276
],
  1519986000,
 42.16,
 1.2281
[
 1519987201,
 43.46,
 1.2291
],
  1519988401,
 43.51,
 1.2291
],
  1519989601,
 43.4,
  1.2317
],
 1519990800,
 43.46,
 1.2317
],
 1519992001,
 43.07,
 1.2304
 1519993201,
 43.56,
  1.2316
],
  1519994401,
 43.75,
  1.2319
],
```

```
1519995601,
  43.15,
  1.2308
],
  1519996801,
  42.94,
  1.2309
],
[
  1519998001,
 42.99,
 1.2315
],
  1519999201,
  42.33,
  1.2309
  1520000400,
  41.93,
  1.2299
],
  1520001601,
  42.31,
  1.2303
],
  1520002801,
  42.5,
  1.2313
],
  1520004000,
 42.8,
  1.2326
],
  1520005201,
  42.67,
  1.2317
],
  1520006401,
 42.29,
  1.2309
],
  1520007600,
  42.33,
  1.2309
],
  1520008800,
  42.63,
```

3.2.5 OrderbookData

```
class oandapyV20.endpoints.forexlabs.OrderbookData(params)
    Bases: oandapyV20.endpoints.forexlabs.ForexLabs
    OrderbookData.
    Get the 'orderbook data' for an instrument.
    ENDPOINT = 'labs/v1/orderbook_data'
    EXPECTED_STATUS = 200
    METHOD = 'GET'
    __init__(params)
```

Instantiate an OrderbookData request.

Parameters params (dict (required)) – query params to send, check developer.oanda.com for details.

```
>>> r = labs.CommitmentOfTraders(params=params)
>>> client.request(r)
>>> print(r.response)
```

Output:

```
"os": 0.2615,
  "pl": 0.5633
},
"1.223": {
  "ps": 1.1266,
  "ol": 0.5021,
  "os": 0.2197,
  "pl": 0.3854
"1.288": {
  "ps": 0,
  "ol": 0.0105,
  "os": 0.0105,
  "pl": 0
},
"1.22": {
  "ps": 0.9191,
  "ol": 0.6486,
  "os": 0.136,
  "pl": 0.2965
},
"1.2245": {
  "ps": 0.5336,
  "ol": 0.5021,
  "os": 0.3975,
  "pl": 0.4447
},
"1.1825": {
  "ps": 0.1779,
  "ol": 0.1465,
  "os": 0.0628,
  "pl": 0
},
"1.2085": {
  "ps": 0.1482,
  "ol": 0.2092,
  "os": 0.2197,
  "pl": 0.1482
},
"1.26": {
  "ps": 0,
  "ol": 0.2197,
  "os": 0.68,
  "pl": 0
},
"1.25": {
  "ps": 0.0593,
  "ol": 0.272,
  "os": 1.0566,
  "pl": 0.1186
},
"1.24": {
  "ps": 0.1186,
  "ol": 0.4289,
  "os": 0.8264,
  "pl": 0.4447
```

```
}
}
}
```

3.2.6 Spreads

```
class oandapyV20.endpoints.forexlabs.Spreads(params)
Bases: oandapyV20.endpoints.forexlabs.ForexLabs

Spreads.

Get the spread information for an instrument.
```

Parameters params (dict (required)) – query params to send, check developer.oanda.com for details.

```
>>> r = labs.Spreads(params=params)
>>> client.request(r)
>>> print(r.response)
```

Output:

```
1.7
]
]
}
```

3.3 oandapyV20.endpoints.instruments

3.3.1 InstrumentsCandles

Parameters

- instrument (string (required)) the instrument to fetch candle data for
- params (dict) optional request query parameters, check developer.oanda.com for details

Params example:

```
{
  "count": 5,
  "granularity": "M5"
}
```

Candle data example:

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.instruments as instruments
>>> client = oandapyV20.API(access_token=...)
>>> params = ...
>>> r = instruments.InstrumentsCandles(instrument="DE30_EUR",
>>> params=params)
>>> client.request(r)
>>> print r.response
```

Output:

```
{
  "candles": [
     {
        "volume": 132,
        "mid": {
```

```
"h": "10508.0",
      "c": "10506.0",
      "1": "10503.8",
      "o": "10503.8"
    },
    "complete": true,
    "time": "2016-10-17T19:35:00.00000000Z"
  },
    "volume": 162,
    "mid": {
     "h": "10507.0",
     "c": "10504.9",
     "1": "10502.0",
      "o": "10506.0"
    },
    "complete": true,
    "time": "2016-10-17T19:40:00.000000000Z"
  },
  {
    "volume": 196,
    "mid": {
      "h": "10509.8",
      "c": "10505.0",
     "1": "10502.6",
     "o": "10504.9"
   },
    "complete": true,
    "time": "2016-10-17T19:45:00.00000000Z"
  },
    "volume": 153,
    "mid": {
     "h": "10510.1",
      "c": "10509.0",
      "1": "10504.2",
      "o": "10505.0"
    "complete": true,
    "time": "2016-10-17T19:50:00.00000000Z"
  },
    "volume": 172,
    "mid": {
     "h": "10509.8",
      "c": "10507.8",
     "1": "10503.2",
      "o": "10509.0"
    },
    "complete": true,
    "time": "2016-10-17T19:55:00.00000000Z"
],
"granularity": "M5",
"instrument": "DE30/EUR"
```

3.3.2 InstrumentsOrderBook

Parameters

- instrument (string (required)) the instrument to fetch candle data for
- params (dict) optional request query parameters, check developer.oanda.com for details

Params example:

```
{}
```

OrderBook data example:

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.instruments as instruments
>>> client = oandapyV20.API(access_token=...)
>>> params = ...
>>> r = instruments.InstrumentsOrderBook(instrument="EUR_USD",
>>> params=params)
>>> client.request(r)
>>> print r.response
```

Output:

```
"orderBook": {
  "instrument": "EUR_USD",
  "buckets": [
      "price": "1.12850",
      "shortCountPercent": "0.2352",
      "longCountPercent": "0.2666"
    },
    {
      "price": "1.12900",
      "shortCountPercent": "0.2195",
      "longCountPercent": "0.3293"
    },
      "price": "1.12950",
      "shortCountPercent": "0.3136",
      "longCountPercent": "0.2901"
    },
    {
```

```
"price": "1.13000",
  "shortCountPercent": "0.3842",
  "longCountPercent": "0.4156"
},
  "price": "1.13050",
  "shortCountPercent": "0.1960",
  "longCountPercent": "0.3685"
},
  "price": "1.13100",
  "shortCountPercent": "0.2431",
  "longCountPercent": "0.2901"
},
{
  "price": "1.13150",
  "shortCountPercent": "0.2509",
  "longCountPercent": "0.3136"
},
  "price": "1.13200",
  "shortCountPercent": "0.2587",
  "longCountPercent": "0.3450"
},
  "price": "1.13250",
  "shortCountPercent": "0.3842",
  "longCountPercent": "0.2666"
},
{
  "price": "1.13300",
  "shortCountPercent": "0.3371",
  "longCountPercent": "0.3371"
},
  "price": "1.13350",
  "shortCountPercent": "0.3528",
  "longCountPercent": "0.2744"
},
{
  "price": "1.13400",
  "shortCountPercent": "0.3842",
  "longCountPercent": "0.3136"
},
{
  "price": "1.13450",
  "shortCountPercent": "0.2039",
  "longCountPercent": "0.2744"
},
  "price": "1.13500",
  "shortCountPercent": "0.1882",
  "longCountPercent": "0.3371"
},
{
  "price": "1.13550",
  "shortCountPercent": "0.0235",
```

```
"longCountPercent": "0.0392"
},
{
  "price": "1.13600",
  "shortCountPercent": "0.0549",
  "longCountPercent": "0.0314"
},
  "price": "1.13650",
  "shortCountPercent": "0.1333",
  "longCountPercent": "0.0314"
},
  "price": "1.13700",
  "shortCountPercent": "0.1176",
  "longCountPercent": "0.1019"
},
{
  "price": "1.13750",
  "shortCountPercent": "0.1568",
  "longCountPercent": "0.0784"
},
  "price": "1.13800",
  "shortCountPercent": "0.1176",
  "longCountPercent": "0.0862"
},
{
  "price": "1.13850",
  "shortCountPercent": "0.2117",
  "longCountPercent": "0.1960"
},
  "price": "1.13900",
  "shortCountPercent": "0.4548",
  "longCountPercent": "0.2587"
},
  "price": "1.13950",
  "shortCountPercent": "0.2979",
  "longCountPercent": "0.3215"
},
{
  "price": "1.14000",
  "shortCountPercent": "0.7449",
  "longCountPercent": "0.2901"
},
  "price": "1.14050",
  "shortCountPercent": "0.2117",
  "longCountPercent": "0.1176"
},
  "price": "1.14100",
  "shortCountPercent": "0.1960",
  "longCountPercent": "0.1333"
```

```
{
    "price": "1.14150",
        "shortCountPercent": "0.1882",
        "longCountPercent": "0.1176"
    }
    l,
    "time": "2017-06-28T10:00:00Z",
    "price": "1.13609",
    "bucketWidth": "0.00050"
}
```

3.3.3 InstrumentsPositionBook

Bases: oandapyV20.endpoints.instruments.Instruments

Get positionbook data for a specified Instrument.

```
ENDPOINT = 'v3/instruments/{instrument}/positionBook'
```

```
EXPECTED_STATUS = 200
```

METHOD = 'GET'

__init___(instrument, params=None)

Instantiate an InstrumentsPositionBook request.

Parameters

- instrument (string (required)) the instrument to fetch candle data for
- params (dict) optional request query parameters, check developer.oanda.com for details

Params example:

```
{}
```

PositionBook data example:

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.instruments as instruments
>>> client = oandapyV20.API(access_token=...)
>>> params = ...
>>> r = instruments.InstrumentsPositionBook(instrument="EUR_USD",
>>> params=params)
>>> client.request(r)
>>> print r.response
```

Output:

40

```
"price": "1.12800",
  "shortCountPercent": "0.2670",
  "longCountPercent": "0.2627"
},
  "price": "1.12850",
  "shortCountPercent": "0.2034",
  "longCountPercent": "0.2712"
},
  "price": "1.12900",
  "shortCountPercent": "0.2034",
  "longCountPercent": "0.2161"
},
{
  "price": "1.12950",
  "shortCountPercent": "0.2670",
  "longCountPercent": "0.2839"
},
  "price": "1.13000",
  "shortCountPercent": "0.2755",
  "longCountPercent": "0.3221"
},
  "price": "1.13050",
  "shortCountPercent": "0.1949",
  "longCountPercent": "0.2839"
},
{
  "price": "1.13100",
  "shortCountPercent": "0.2288",
  "longCountPercent": "0.2712"
},
  "price": "1.13150",
  "shortCountPercent": "0.2416",
  "longCountPercent": "0.2712"
},
{
  "price": "1.13200",
  "shortCountPercent": "0.2204",
  "longCountPercent": "0.3178"
},
{
  "price": "1.13250",
  "shortCountPercent": "0.2543",
  "longCountPercent": "0.2458"
},
  "price": "1.13300",
  "shortCountPercent": "0.2839",
  "longCountPercent": "0.2585"
},
{
  "price": "1.13350",
  "shortCountPercent": "0.3602",
```

```
"longCountPercent": "0.3094"
},
{
  "price": "1.13400",
  "shortCountPercent": "0.2882",
  "longCountPercent": "0.3560"
},
  "price": "1.13450",
  "shortCountPercent": "0.2500",
  "longCountPercent": "0.3009"
},
  "price": "1.13500",
  "shortCountPercent": "0.1738",
  "longCountPercent": "0.3475"
},
{
  "price": "1.13550",
  "shortCountPercent": "0.2119",
  "longCountPercent": "0.2797"
},
  "price": "1.13600",
  "shortCountPercent": "0.1483",
  "longCountPercent": "0.3094"
},
{
  "price": "1.13650",
  "shortCountPercent": "0.1483",
  "longCountPercent": "0.1314"
},
  "price": "1.13700",
  "shortCountPercent": "0.1568",
  "longCountPercent": "0.2034"
},
  "price": "1.13750",
  "shortCountPercent": "0.1398",
  "longCountPercent": "0.1271"
},
{
  "price": "1.13800",
  "shortCountPercent": "0.1314",
  "longCountPercent": "0.2034"
},
  "price": "1.13850",
  "shortCountPercent": "0.1483",
  "longCountPercent": "0.1695"
},
  "price": "1.13900",
  "shortCountPercent": "0.2924",
  "longCountPercent": "0.1653"
```

```
{
    "price": "1.13950",
    "shortCountPercent": "0.1526",
    "longCountPercent": "0.1865"
    "price": "1.14000",
    "shortCountPercent": "0.4365",
    "longCountPercent": "0.2034"
 },
    "price": "1.14050",
    "shortCountPercent": "0.1398",
    "longCountPercent": "0.1144"
 }
],
"time": "2017-06-28T10:00:00Z",
"price": "1.13609",
"bucketWidth": "0.00050"
```

3.4 oandapyV20.endpoints.orders

3.4.1 OrderCancel

```
class oandapyV20.endpoints.orders.OrderCancel (accountID, orderID)
    Bases: oandapyV20.endpoints.orders.Orders

Cancel a pending Order in an Account.

ENDPOINT = 'v3/accounts/{accountID}/orders/{orderID}/cancel'

EXPECTED_STATUS = 200

METHOD = 'PUT'

__init__ (accountID, orderID)
    Instantiate an OrdersCancel request.
```

Parameters

- accountID (string (required)) id of the account to perform the request on.
- orderID (string (required)) id of the account to perform the request on.

Example:

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.orders as orders
>>> client = oandapyV20.API(access_token=...)
>>> r = orders.OrderCancel(accountID= ..., orderID=...)
>>> client.request(r)
>>> print r.response
```

Output:

```
{
   "orderCancelTransaction": {
      "orderID": "2307",
      "clientOrderID": "myID",
      "reason": "CLIENT_REQUEST",
      "batchID": "2308",
      "time": "2016-10-25T20:53:03.789670387Z",
      "type": "ORDER_CANCEL",
      "userID": 1435156,
      "id": "2308",
      "accountID": "101-004-1435156-001"
},
   "lastTransactionID": "2308",
   "relatedTransactionIDs": [
      "2308"
]
```

3.4.2 OrderClientExtensions

 $\textbf{class} \ \, \texttt{oandapyV20.endpoints.orders.OrderClientExtensions} \, (\textit{accountID}, \textit{orderID}, \textit{data}) \\ \text{Bases:} \ \, \texttt{oandapyV20.endpoints.orders.Orders} \, . \\$

Update the Client Extensions for an Order in an Account.

Warning: Do not set, modify or delete clientExtensions if your account is associated with MT4.

Parameters

- accountID (string (required)) id of the account to perform the request on.
- orderID (string (required)) id of the order to perform the request on.
- data (JSON (required)) json orderbody to send

Orderbody example:

```
{
  "clientExtensions": {
    "comment": "myComment",
    "id": "myID"
    }
}
```

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.orders as orders
>>> client = oandapyV20.API(access_token=...)
>>> r = orders.OrderClientExtensions(accountID, orderID, data=data)
>>> client.request(r)
>>> print r.response
```

```
"lastTransactionID": "2305",
   "orderClientExtensionsModifyTransaction": {
      "orderID": "2304",
      "batchID": "2305",
      "clientExtensionsModify": {
            "comment": "myComment",
            "id": "myID"
      },
      "time": "2016-10-25T15:56:43.075594239Z",
      "type": "ORDER_CLIENT_EXTENSIONS_MODIFY",
      "userID": 1435156,
      "id": "2305",
      "accountID": "101-004-1435156-001"
      },
      "relatedTransactionIDs": [
            "2305"
      ]
}
```

3.4.3 OrderCreate

```
class oandapyV20.endpoints.orders.OrderCreate(accountID, data)
    Bases: oandapyV20.endpoints.orders.Orders

Create an Order for an Account.

ENDPOINT = 'v3/accounts/{accountID}/orders'

EXPECTED_STATUS = 201

HEADERS = {'Content-Type': 'application/json'}

METHOD = 'POST'

__init__(accountID, data)
    Instantiate an OrderCreate request.
```

Parameters

- accountID (string (required)) id of the account to perform the request on.
- data (JSON (required)) json orderbody to send

Orderbody example:

```
{
  "order": {
    "price": "1.2",
    "stopLossOnFill": {
       "timeInForce": "GTC",
       "price": "1.22"
```

```
},
  "timeInForce": "GTC",
  "instrument": "EUR_USD",
  "units": "-100",
  "type": "LIMIT",
  "positionFill": "DEFAULT"
}
```

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.orders as orders
>>> client = oandapyV20.API(access_token=...)
>>> r = orders.OrderCreate(accountID, data=data)
>>> client.request(r)
>>> print r.response
```

```
"orderCreateTransaction": {
 "price": "1.20000",
  "stopLossOnFill": {
   "timeInForce": "GTC",
    "price": "1.22000"
  },
 "timeInForce": "GTC",
  "reason": "CLIENT_ORDER",
  "id": "2304",
  "batchID": "2304",
 "triggerCondition": "TRIGGER_DEFAULT",
 "positionFill": "DEFAULT",
 "userID": 1435156,
 "instrument": "EUR_USD",
 "time": "2016-10-24T21:48:18.593753865Z",
 "units": "-100",
  "type": "LIMIT_ORDER",
  "accountID": "101-004-1435156-001"
"lastTransactionID": "2304",
"relatedTransactionIDs": [
  "2304"
]
```

3.4.4 OrderDetails

```
class oandapyV20.endpoints.orders.OrderDetails(accountID, orderID)
   Bases: oandapyV20.endpoints.orders.Orders
   Get details for a single Order in an Account.

ENDPOINT = 'v3/accounts/{accountID}/orders/{orderID}'

EXPECTED_STATUS = 200

METHOD = 'GET'

__init__(accountID, orderID)
   Instantiate an OrderDetails request.
```

Parameters

- accountID (string (required)) id of the account to perform the request on.
- orderID (string (required)) id of the order to perform the request on.

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.orders as orders
>>> client = oandapyV20.API(access_token=...)
>>> r = orders.OrderDetails(accountID=..., orderID=...)
>>> client.request(r)
>>> print r.response
```

Output:

```
"order": {
 "partialFill": "DEFAULT_FILL",
 "price": "1.20000",
  "stopLossOnFill": {
   "timeInForce": "GTC",
    "price": "1.22000"
  },
  "timeInForce": "GTC",
  "createTime": "2016-10-25T21:07:21.065554321Z",
 "triggerCondition": "TRIGGER_DEFAULT",
 "positionFill": "POSITION_DEFAULT",
 "id": "2309",
 "instrument": "EUR_USD",
 "state": "PENDING",
 "units": "-100",
 "type": "LIMIT"
},
"lastTransactionID": "2309"
```

3.4.5 OrderList

```
class oandapyV20.endpoints.orders.OrderList (accountID, params=None)
    Bases: oandapyV20.endpoints.orders.Orders

Create an Order for an Account.

ENDPOINT = 'v3/accounts/{accountID}/orders'

EXPECTED_STATUS = 200

METHOD = 'GET'

__init___(accountID, params=None)
    Instantiate an OrderList request.
```

Parameters

- accountID (string (required)) id of the account to perform the request on.
- params (dict) optional request query parameters, check developer.oanda.com for details

Example:

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.orders as orders
>>> client = oandapyV20.API(access_token=...)
>>> r = orders.OrderList(accountID)
>>> client.request(r)
>>> print r.response
```

Output:

```
"orders": [
 {
    "partialFill": "DEFAULT_FILL",
    "price": "1.20000",
    "stopLossOnFill": {
     "timeInForce": "GTC",
     "price": "1.22000"
   "timeInForce": "GTC",
   "createTime": "2016-10-05T10:25:47.627003645Z",
    "triggerCondition": "TRIGGER_DEFAULT",
    "positionFill": "POSITION_DEFAULT",
    "id": "2125",
    "instrument": "EUR_USD",
    "state": "PENDING",
    "units": "-100",
    "type": "LIMIT"
  }
],
"lastTransactionID": "2129"
```

3.4.6 OrderReplace

Parameters

- accountID (string (required)) id of the account to perform the request on.
- orderID (string (required)) id of the order to perform the request on.
- data (JSON (required)) json orderbody to send

Orderbody example:

```
{
  "order": {
    "units": "-500000",
    "instrument": "EUR_USD",
    "price": "1.25000",
    "type": "LIMIT"
  }
}
```

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.orders as orders
>>> client = oandapyV20.API(access_token=...)
>>> data =

{
    "order": {
        "units": "-500000",
        "instrument": "EUR_USD",
        "price": "1.25000",
        "type": "LIMIT"
        }
}
```

```
>>> r = orders.OrderReplace(accountID=..., orderID=..., data=data)
>>> client.request(r)
>>> print r.response
```

Output:

```
"orderCreateTransaction": {
 "price": "1.25000",
 "timeInForce": "GTC",
 "reason": "REPLACEMENT",
  "clientExtensions": {
   "comment": "myComment",
   "id": "myID"
 },
 "id": "2307",
 "batchID": "2306",
 "triggerCondition": "TRIGGER_DEFAULT",
 "replacesOrderID": "2304",
 "positionFill": "DEFAULT",
 "userID": 1435156,
 "instrument": "EUR_USD",
 "time": "2016-10-25T19:45:38.558056359Z",
  "units": "-500000",
  "type": "LIMIT_ORDER",
  "accountID": "101-004-1435156-001"
},
"orderCancelTransaction": {
 "orderID": "2304",
 "clientOrderID": "myID",
 "reason": "CLIENT_REQUEST_REPLACED",
 "batchID": "2306",
 "time": "2016-10-25T19:45:38.558056359Z",
  "type": "ORDER_CANCEL",
  "replacedByOrderID": "2307",
```

3.4.7 OrdersPending

```
class oandapyV20.endpoints.orders.OrdersPending(accountID)
    Bases: oandapyV20.endpoints.orders.Orders

List all pending Orders in an Account.

ENDPOINT = 'v3/accounts/{accountID}/pendingOrders'

EXPECTED_STATUS = 200

METHOD = 'GET'

__init__(accountID)

Instantiate an OrdersPending request.
```

Parameters accountID (string (required)) - id of the account to perform the request on.

Example:

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.orders as orders
>>> client = oandapyV20.API(access_token=...)
>>> r = orders.OrdersPending(accountID)
>>> client.request(r)
>>> print r.response
```

Output:

```
"positionFill": "POSITION_DEFAULT",
    "createTime": "2016-10-24T21:48:18.593753865Z",
    "instrument": "EUR_USD",
    "state": "PENDING",
    "units": "-100",
    "type": "LIMIT"
    }
],
    "lastTransactionID": "2305"
}
```

3.5 oandapyV20.endpoints.positions

3.5.1 OpenPositions

```
\begin{tabular}{ll} \textbf{class} & \texttt{oandapyV20.endpoints.positions.OpenPositions} \ (\textit{accountID}) \\ \textbf{Bases:} & \texttt{oandapyV20.endpoints.positions.Positions} \end{tabular}
```

OpenPositions.

List all open Positions for an Account. An open Position is a Position in an Account that currently has a Trade opened for it.

```
ENDPOINT = 'v3/accounts/{accountID}/openPositions'
EXPECTED_STATUS = 200

METHOD = 'GET'
__init__(accountID)
    Instantiate an OpenPositions request.
```

Parameters accountID (string (required))—id of the account to perform the request on.

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.positions as positions
>>> accountID = ...
>>> client = oandapyV20.API(access_token=...)
>>> r = positions.OpenPositions(accountID=accountID)
>>> client.request(r)
>>> print r.response
```

Output:

```
"unrealizedPL": "-284.0000",
      "tradeIDs": [
        "2315"
      "resettablePL": "404.5000",
      "units": "10",
      "averagePrice": "10678.3",
      "pl": "404.5000"
    },
    "instrument": "DE30_EUR",
    "resettablePL": "-13759.8000",
    "pl": "-13759.8000"
  },
    "short": {
      "unrealizedPL": "-0.0738",
      "tradeIDs": [
        "2323"
      "resettablePL": "0.0000",
      "units": "-100",
      "averagePrice": "1.09843",
      "pl": "0.0000"
    },
    "unrealizedPL": "-0.0738",
    "long": {
      "units": "0",
      "resettablePL": "-44.6272",
      "unrealizedPL": "0.0000",
      "pl": "-44.6272"
    "instrument": "EUR_USD",
    "resettablePL": "-44.6272",
    "pl": "-44.6272"
  }
],
"lastTransactionID": "2327"
```

3.5.2 PositionClose

Parameters

```
class oandapyV20.endpoints.positions.PositionClose (accountID, instrument, data)
    Bases: oandapyV20.endpoints.positions.Positions

Closeout the open Position regarding instrument in an Account.

ENDPOINT = 'v3/accounts/{accountID}/positions/{instrument}/close'

EXPECTED_STATUS = 200

HEADERS = {'Content-Type': 'application/json'}

METHOD = 'PUT'

__init__ (accountID, instrument, data)
    Instantiate a PositionClose request.
```

- accountID (string (required)) id of the account to perform the request on.
- instrument (string (required)) instrument to close partially or fully.
- data (dict (required)) closeout specification data to send, check developer.oanda.com for details.

Data body example:

```
{
    "longUnits": "ALL"
}
```

Output:

```
"longOrderCreateTransaction": {
  "longPositionCloseout": {
    "units": "ALL",
    "instrument": "EUR_USD"
  } ,
 "batchID": "6390",
 "reason": "POSITION_CLOSEOUT",
 "id": "6390",
 "timeInForce": "FOK",
 "positionFill": "REDUCE_ONLY",
 "userID": "<USERID>",
 "instrument": "EUR_USD",
 "time": "2016-06-22T18:41:35.034041665Z", "units": "-251",
  "type": "MARKET_ORDER",
  "accountID": "<ACCOUNT>"
"relatedTransactionIDs": [
 "6390",
 "6391"
"lastTransactionID": "6391",
"longOrderFillTransaction": {
  "price": "1.13018",
  "batchID": "6390",
  "accountBalance": "43650.69807",
```

```
"reason": "MARKET_ORDER_POSITION_CLOSEOUT",
"tradesClosed": [
  {
    "units": "-1",
    "financing": "0.00000",
    "realizedPL": "-0.00013",
    "tradeID": "6383"
  },
  {
    "units": "-250",
    "financing": "0.00000",
    "realizedPL": "-0.03357",
    "tradeID": "6385"
  }
],
"id": "6391",
"orderID": "6390",
"financing": "0.00000",
"userID": "<USERID>",
"instrument": "EUR_USD",
"time": "2016-06-22T18:41:35.034041665Z",
"units": "-251",
"type": "ORDER_FILL",
"pl": "-0.03370",
"accountID": "<ACCOUNT>"
```

3.5.3 PositionDetails

```
 \textbf{class} \  \, \texttt{oandapyV20.endpoints.positions.PositionDetails} \, (\textit{accountID}, \textit{instrument}) \\ \text{Bases:} \, \texttt{oandapyV20.endpoints.positions.Positions}
```

PositionDetails.

Get the details of a single instrument's position in an Account. The position may be open or not.

Parameters

- account ID (string (required)) id of the account to perform the request on.
- instrument (string (required))—id of the instrument to get the position details for.

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.positions as positions
>>> accountID = ...
>>> instrument = ...
>>> client = oandapyV20.API(access_token=...)
```

```
>>> r = positions.PositionDetails(accountID=accountID, instrument)
>>> client.request(r)
>>> print r.response
```

Output:

```
"position": {
 "short": {
   "unrealizedPL": "-0.0738",
   "tradeIDs": [
     "2323"
   ],
   "resettablePL": "0.0000",
   "units": "-100",
   "averagePrice": "1.09843",
    "pl": "0.0000"
  },
  "unrealizedPL": "-0.0738",
  "long": {
   "units": "0",
   "resettablePL": "-44.6272",
   "unrealizedPL": "0.0000",
   "pl": "-44.6272"
 },
 "instrument": "EUR_USD",
 "resettablePL": "-44.6272",
 "pl": "-44.6272"
},
"lastTransactionID": "2327"
```

3.5.4 PositionList

```
class oandapyV20.endpoints.positions.PositionList(accountID)
    Bases: oandapyV20.endpoints.positions.Positions
```

PositionList.

List all Positions for an Account. The Positions returned are for every instrument that has had a position during the lifetime of the Account.

```
ENDPOINT = 'v3/accounts/{accountID}/positions'
EXPECTED_STATUS = 200

METHOD = 'GET'
__init__(accountID)
    Instantiate a PositionList request.
```

Parameters accountID (string (required)) - id of the account to perform the request on.

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.positions as positions
>>> accountID = ...
```

```
>>> client = oandapyV20.API(access_token=...)
>>> r = positions.PositionList(accountID=accountID)
>>> client.request(r)
>>> print r.response
```

Output:

```
"positions": [
 {
   "short": {
     "units": "0",
     "resettablePL": "-272.6805",
     "unrealizedPL": "0.0000",
     "pl": "-272.6805"
   },
    "unrealizedPL": "0.0000",
    "long": {
     "units": "0",
      "resettablePL": "0.0000",
     "unrealizedPL": "0.0000",
     "pl": "0.0000"
   "instrument": "EUR_GBP",
   "resettablePL": "-272.6805",
   "pl": "-272.6805"
  },
    "short": {
      "unrealizedPL": "870.0000",
      "tradeIDs": [
       "2121",
       "2123"
     ],
      "resettablePL": "-13959.3000",
      "units": "-20",
     "averagePrice": "10581.5",
     "pl": "-13959.3000"
   "unrealizedPL": "870.0000",
   "long": {
     "units": "0",
     "resettablePL": "404.5000",
     "unrealizedPL": "0.0000",
      "pl": "404.5000"
   "instrument": "DE30_EUR",
    "resettablePL": "-13554.8000",
    "pl": "-13554.8000"
  },
    "short": {
     "units": "0",
     "resettablePL": "0.0000",
     "unrealizedPL": "0.0000",
      "pl": "0.0000"
   },
```

```
"unrealizedPL": "0.0000",
    "long": {
        "units": "0",
        "resettablePL": "-12.8720",
        "unrealizedPL": "0.0000",
        "pl": "-12.8720"
        },
        "instrument": "EUR_USD",
        "resettablePL": "-12.8720",
        "pl": "-12.8720"
        }
     ],
     "lastTransactionID": "2124"
}
```

3.6 oandapyV20.endpoints.pricing

3.6.1 PricingInfo

```
class oandapyV20.endpoints.pricing.PricingInfo(accountID, params=None)
    Bases: oandapyV20.endpoints.pricing.Pricing
Pricing.

Get pricing information for a specified list of Instruments within an account.

ENDPOINT = 'v3/accounts/{accountID}/pricing'

EXPECTED_STATUS = 200

METHOD = 'GET'

__init__(accountID, params=None)
    Instantiate a PricingStream APIRequest instance.
```

Parameters

- accountID (string (required)) the accountID of the account.
- params (dict (required)) parameters for the request, check developer.oanda.com for details.

Example

```
>>> r = pricing.PricingInfo(accountID=accountID, params=params)
>>> rv = api.request(r)
>>> print r.response
```

Output:

```
"prices": [
   "status": "tradeable",
   "instrument": "EUR_USD",
    "quoteHomeConversionFactors": {
     "negativeUnits": "0.89160730",
      "positiveUnits": "0.89150397"
   },
    "asks": [
     {
        "price": "1.12170",
        "liquidity": 10000000
     },
        "price": "1.12172",
        "liquidity": 10000000
      }
   ],
    "time": "2016-10-05T05:28:16.729643492Z",
    "closeoutAsk": "1.12174",
    "bids": [
        "price": "1.12157",
        "liquidity": 10000000
     },
        "price": "1.12155",
        "liquidity": 10000000
      }
    "closeoutBid": "1.12153",
    "unitsAvailable": {
     "default": {
       "short": "506246",
        "long": "506128"
      "reduceOnly": {
        "short": "0",
        "long": "0"
     },
      "openOnly": {
        "short": "506246",
        "long": "506128"
      },
      "reduceFirst": {
        "short": "506246",
        "long": "506128"
   }
  },
```

```
"status": "tradeable",
"instrument": "EUR_JPY",
"quoteHomeConversionFactors": {
 "negativeUnits": "0.00867085",
  "positiveUnits": "0.00866957"
"asks": [
    "price": "115.346",
    "liquidity": 1000000
  },
    "price": "115.347",
    "liquidity": 2000000
  },
    "price": "115.348",
    "liquidity": 5000000
  },
    "price": "115.350",
    "liquidity": 10000000
  }
],
"time": "2016-10-05T05:28:15.621238671Z",
"closeoutAsk": "115.350",
"bids": [
    "price": "115.329",
    "liquidity": 1000000
  },
    "price": "115.328",
    "liquidity": 2000000
  },
    "price": "115.327",
    "liquidity": 5000000
  },
    "price": "115.325",
    "liquidity": 10000000
"closeoutBid": "115.325",
"unitsAvailable": {
  "default": {
    "short": "506262",
    "long": "506112"
  },
  "reduceOnly": {
    "short": "0",
    "long": "0"
  "openOnly": {
    "short": "506262",
```

```
"long": "506112"
},
"reduceFirst": {
    "short": "506262",
    "long": "506112"
}
}
}
```

3.6.2 PricingStream

```
class oandapyV20.endpoints.pricing.PricingStream(accountID, params=None)
    Bases: oandapyV20.endpoints.pricing.Pricing
```

PricingStream.

Get realtime pricing information for a specified list of Instruments.

```
ENDPOINT = 'v3/accounts/{accountID}/pricing/stream'
EXPECTED_STATUS = 200

METHOD = 'GET'

STREAM = True
__init__(accountID, params=None)
```

Instantiate a PricingStream APIRequest instance.

Parameters

- accountID (string (required)) the accountID of the account.
- params (dict (required)) parameters for the request, check developer.oanda.com for details.

Example

```
>>> import oandapyV20
>>> from oandapyV20 import API
>>> import oandapyV20.endpoints.pricing as pricing
>>> accountID = "..."
>>> api = API(access_token="...")
>>> params =
{
        "instruments": "EUR_USD,EUR_JPY"
}
```

```
>>> r = pricing.PricingStream(accountID=accountID, params=params)
>>> rv = api.request(r)
>>> maxrecs = 100
>>> for ticks in r:
>>> print json.dumps(R, indent=4),","
>>> if maxrecs == 0:
>>> r.terminate("maxrecs records received")
```

Output:

```
"status": "tradeable",
  "instrument": "EUR_JPY",
  "asks": [
   {
      "price": "114.312",
      "liquidity": 1000000
    },
    {
      "price": "114.313",
      "liquidity": 2000000
    },
      "price": "114.314",
      "liquidity": 5000000
    },
      "price": "114.316",
      "liquidity": 10000000
   }
 ],
  "time": "2016-10-27T08:38:43.094548890Z",
  "closeoutAsk": "114.316",
  "type": "PRICE",
  "closeoutBid": "114.291",
  "bids": [
      "price": "114.295",
      "liquidity": 1000000
    },
      "price": "114.294",
      "liquidity": 2000000
     "price": "114.293",
     "liquidity": 5000000
    },
      "price": "114.291",
      "liquidity": 10000000
 ]
},
{
 "type": "HEARTBEAT",
  "time": "2016-10-27T08:38:44.327443673Z"
},
{
 "status": "tradeable",
 "instrument": "EUR_USD",
  "asks": [
      "price": "1.09188",
      "liquidity": 10000000
    },
```

```
"price": "1.09190",
      "liquidity": 10000000
  ],
  "time": "2016-10-27T08:38:45.664613867Z",
  "closeoutAsk": "1.09192",
  "type": "PRICE",
  "closeoutBid": "1.09173",
  "bids": [
      "price": "1.09177",
      "liquidity": 10000000
    },
      "price": "1.09175",
      "liquidity": 10000000
},
{
  "status": "tradeable",
  "instrument": "EUR_JPY",
  "asks": [
      "price": "114.315",
      "liquidity": 1000000
    },
      "price": "114.316",
      "liquidity": 2000000
    },
      "price": "114.317",
      "liquidity": 5000000
    },
      "price": "114.319",
      "liquidity": 10000000
 ],
  "time": "2016-10-27T08:38:45.681572782Z",
  "closeoutAsk": "114.319",
  "type": "PRICE",
  "closeoutBid": "114.294",
  "bids": [
    {
      "price": "114.298",
      "liquidity": 1000000
    },
      "price": "114.297",
     "liquidity": 2000000
    },
      "price": "114.296",
      "liquidity": 5000000
```

```
},
    {
        "price": "114.294",
        "liquidity": 10000000
     }
     ]
}
```

terminate (message=")

terminate the stream.

Calling this method will stop the generator yielding tickrecords. A message can be passed optionally.

3.7 oandapyV20.endpoints.trades

3.7.1 OpenTrades

```
class oandapyV20.endpoints.trades.OpenTrades (accountID)
    Bases: oandapyV20.endpoints.trades.Trades

Get the list of open Trades for an Account.

ENDPOINT = 'v3/accounts/{accountID}/openTrades'

EXPECTED_STATUS = 200

METHOD = 'GET'

__init__ (accountID)

Instantiate an OpenTrades request.
```

Parameters accountID (string (required)) – id of the account to perform the request on

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.trades as trades
>>> client = oandapyV20.API(access_token=...)
>>> r = trades.OpenTrades(accountID=...)
>>> client.request(r)
>>> print r.response
```

Output:

```
}
],
"lastTransactionID": "2317"
}
```

3.7.2 TradeCRCDO

```
class oandapyV20.endpoints.trades.TradeCRCDO (accountID, tradeID, data)
    Bases: oandapyV20.endpoints.trades.Trades

Trade Create Replace Cancel Dependent Orders.

ENDPOINT = 'v3/accounts/{accountID}/trades/{tradeID}/orders'

EXPECTED_STATUS = 200

HEADERS = {'Content-Type': 'application/json'}

METHOD = 'PUT'

__init__ (accountID, tradeID, data)
    Instantiate a TradeClientExtensions request.
```

Parameters

- accountID (string (required)) id of the account to perform the request on.
- tradeID (string (required)) id of the trade to update client extensions for.
- data (dict (required)) clientextension data to send, check developer.oanda.com for details.

Data body example:

```
{
  "takeProfit": {
    "timeInForce": "GTC",
    "price": "1.05"
},
  "stopLoss": {
    "timeInForce": "GTC",
    "price": "1.10"
}
}
```

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.trades as trades
>>> accountID = ...
>>> tradeID = ...
>>> client = oandapyV20.API(access_token=...)
>>> data =

{
    "takeProfit": {
        "timeInForce": "GTC",
        "price": "1.05"
      },
      "stopLoss": {
        "timeInForce": "GTC",
```

```
"price": "1.10"
}
}
```

```
>>> r = trades.TradeCRCDO(accountID=accountID,
>>> tradeID=tradeID,
>>> data=data)
>>> client.request(r)
>>> print r.response
```

Output:

```
"lastTransactionID": "2327",
"stopLossOrderCancelTransaction": {
  "orderID": "2324",
  "batchID": "2325",
  "reason": "CLIENT_REQUEST_REPLACED",
  "time": "2016-10-28T21:00:19.978476830Z",
  "type": "ORDER_CANCEL",
  "replacedByOrderID": "2327",
  "userID": 1435156,
  "id": "2326",
  "accountID": "101-004-1435156-001"
"stopLossOrderTransaction": {
  "tradeID": "2323",
  "price": "1.10000",
  "timeInForce": "GTC",
  "reason": "REPLACEMENT",
  "id": "2327",
  "batchID": "2325",
  "triggerCondition": "TRIGGER_DEFAULT",
  "replacesOrderID": "2324",
  "userID": 1435156,
  "time": "2016-10-28T21:00:19.978476830Z",
  "cancellingTransactionID": "2326",
  "type": "STOP_LOSS_ORDER",
  "accountID": "101-004-1435156-001"
},
"relatedTransactionIDs": [
  "2325",
  "2326",
  "2327"
],
"takeProfitOrderTransaction": {
  "tradeID": "2323",
  "price": "1.05000",
  "timeInForce": "GTC",
  "reason": "CLIENT_ORDER",
  "id": "2325",
  "batchID": "2325",
  "triggerCondition": "TRIGGER_DEFAULT",
  "userID": 1435156,
  "time": "2016-10-28T21:00:19.978476830Z",
  "type": "TAKE_PROFIT_ORDER",
```

```
"accountID": "101-004-1435156-001"
}
```

3.7.3 TradeClientExtensions

Dases. Dandapy vzv. endpoints. trades. Irade

TradeClientExtensions.

Update the Client Extensions for a Trade. Do not add, update or delete the Client Extensions if your account is associated with MT4.

```
ENDPOINT = 'v3/accounts/{accountID}/trades/{tradeID}/clientExtensions'
EXPECTED_STATUS = 200

HEADERS = {'Content-Type': 'application/json'}

METHOD = 'PUT'
__init__(accountID, tradeID, data=None)
    Instantiate a TradeClientExtensions request.
```

Parameters

- account ID (string (required)) id of the account to perform the request on.
- **tradeID** (string (required)) id of the trade to update client extensions for.
- data (dict (required)) clientextension data to send, check developer.oanda.com for details.

Data body example:

```
{
  "clientExtensions": {
    "comment": "myComment",
    "id": "myID2315"
  }
}
```

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.trades as trades
>>> accountID = ...
>>> tradeID = ...
>>> client = oandapyV20.API(access_token=...)
>>> data =

{
    "clientExtensions": {
        "comment": "myComment",
        "id": "myID2315"
      }
}
```

Output:

```
"tradeClientExtensionsModifyTransaction": {
 "batchID": "2319",
 "tradeID": "2315",
  "time": "2016-10-28T20:32:39.356516787Z",
  "tradeClientExtensionsModify": {
    "comment": "myComment",
    "id": "myID2315"
  },
 "type": "TRADE_CLIENT_EXTENSIONS_MODIFY",
 "userID": 1435156,
  "id": "2319",
  "accountID": "101-004-1435156-001"
"lastTransactionID": "2319",
"relatedTransactionIDs": [
  "2319"
]
```

3.7.4 TradeClose

```
class oandapyV20.endpoints.trades.TradeClose(accountID, tradeID, data=None)
    Bases: oandapyV20.endpoints.trades.Trades
```

TradeClose.

Close (partially or fully) a specific open Trade in an Account.

```
ENDPOINT = 'v3/accounts/{accountID}/trades/{tradeID}/close'

EXPECTED_STATUS = 200

HEADERS = {'Content-Type': 'application/json'}

METHOD = 'PUT'

__init__ (accountID, tradeID, data=None)
    Instantiate a TradeClose request.
```

Parameters

- account ID (string (required)) id of the account to perform the request on.
- tradeID (string (required)) id of the trade to close.
- data (dict (optional)) data to send, use this to close a trade partially. Check developer.oanda.com for details.

Data body example:

```
{
    "units": 100
}
```

```
>>> r = trades.TradeClose(accountID=..., data=data)
>>> client.request(r)
>>> print r.response
```

Output:

```
"orderFillTransaction": {
  "price": "1.09289",
  "batchID": "2316",
  "accountBalance": "33848.1208",
  "reason": "MARKET_ORDER_TRADE_CLOSE",
  "tradesClosed": [
      "units": "-100",
     "financing": "0.0000",
     "realizedPL": "-0.1455",
      "tradeID": "2313"
   }
  ],
  "id": "2317",
  "orderID": "2316",
  "financing": "0.0000",
 "userID": 1435156,
 "instrument": "EUR_USD",
 "time": "2016-10-28T15:11:58.023004583Z",
 "units": "-100",
 "type": "ORDER_FILL",
  "pl": "-0.1455",
  "accountID": "101-004-1435156-001"
},
"orderCreateTransaction": {
  "timeInForce": "FOK",
  "reason": "TRADE_CLOSE",
  "tradeClose": {
   "units": "100",
    "tradeID": "2313"
 },
  "id": "2316",
 "batchID": "2316",
 "positionFill": "REDUCE_ONLY",
  "userID": 1435156,
  "instrument": "EUR USD",
  "time": "2016-10-28T15:11:58.023004583Z",
  "units": "-100",
```

```
"type": "MARKET_ORDER",
    "accountID": "101-004-1435156-001"
},
    "lastTransactionID": "2317",
    "relatedTransactionIDs": [
        "2316",
        "2317"
    ]
}
```

3.7.5 TradeDetails

```
class oandapyV20.endpoints.trades.TradeDetails(accountID, tradeID)
   Bases: oandapyV20.endpoints.trades.Trades
   Get the details of a specific Trade in an Account.
   ENDPOINT = 'v3/accounts/{accountID}/trades/{tradeID}'
   EXPECTED_STATUS = 200
   METHOD = 'GET'
   __init__(accountID, tradeID)
        Instantiate a TradeDetails request.
```

Parameters

- accountID (string (required)) id of the account to perform the request on.
- tradeID (string (required)) id of the trade.

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.trades as trades
>>> client = oandapyV20.API(access_token=...)
>>> r = accounts.TradeDetails(accountID=..., tradeID=...)
>>> client.request(r)
>>> print r.response
```

Output:

```
{
  "lastTransactionID": "2317",
  "trade": {
      "instrument": "DE30_EUR",
      "financing": "0.0000",
      "openTime": "2016-10-28T14:28:05.231759081Z",
      "initialUnits": "10",
      "currentUnits": "10",
      "price": "10678.3",
      "unrealizedPL": "226.0000",
      "realizedPL": "0.0000",
      "state": "OPEN",
      "id": "2315"
    }
}
```

3.7.6 TradesList

```
class oandapyV20.endpoints.trades.TradesIist (accountID, params=None)
    Bases: oandapyV20.endpoints.trades.Trades

Get a list of trades for an Account.

ENDPOINT = 'v3/accounts/{accountID}/trades'

EXPECTED_STATUS = 200

METHOD = 'GET'

__init__ (accountID, params=None)
    Instantiate a TradesList request.
```

Parameters

- accountID (string (required)) id of the account to perform the request on.
- params (dict (optional)) query params to send, check developer.oanda.com for details.

Query Params example:

```
{
  "instrument": "DE30_EUR,EUR_USD"
}
```

```
>>> r = trades.TradesList(accountID=..., params=params)
>>> client.request(r)
>>> print r.response
```

Output:

```
"openTime": "2016-10-28T14:27:19.011002322Z",
    "initialUnits": "100",
    "currentUnits": "100",
    "price": "1.09448",
    "unrealizedPL": "-0.0933",
    "realizedPL": "0.0000",
    "state": "OPEN",
    "id": "2313"
    }
],
    "lastTransactionID": "2315"
}
```

3.8 oandapyV20.endpoints.transactions

3.8.1 TransactionDetails

Parameters

- account ID (string (required)) id of the account to perform the request on.
- transactionID (string (required)) id of the transaction

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.transactions as trans
>>> client = oandapyV20.API(access_token=...)
>>> r = trans.TransactionDetails(accountID=..., transactionID=...)
>>> client.request(r)
>>> print r.response
```

Output:

```
"transaction": {
    "price": "1.20000",
    "stopLossOnFill": {
        "timeInForce": "GTC",
        "price": "1.22000"
    },
    "timeInForce": "GTC",
    "reason": "CLIENT_ORDER",
```

```
"id": "2304",
   "batchID": "2304",
   "triggerCondition": "TRIGGER_DEFAULT",
   "positionFill": "DEFAULT",
   "userID": 1435156,
   "instrument": "EUR_USD",
   "time": "2016-10-24T21:48:18.593753865Z",
   "units": "-100",
   "type": "LIMIT_ORDER",
   "accountID": "101-004-1435156-001"
},
   "lastTransactionID": "2311"
}
```

3.8.2 TransactionIDRange

```
class oandapy V20.endpoints.transactions.TransactionIDRange (accountID, params=None)
```

Bases: oandapyV20.endpoints.transactions.Transactions

TransactionIDRange.

Get a range of Transactions for an Account based on Transaction IDs.

```
ENDPOINT = 'v3/accounts/{accountID}/transactions/idrange'
EXPECTED_STATUS = 200
METHOD = 'GET'
__init__(accountID, params=None)
```

Instantiate an TransactionIDRange request.

Parameters

- accountID(string(required)) id of the account to perform the request on.
- params (dict (required)) query params to send, check developer.oanda.com for details.

Query Params example:

```
{
  "to": 2306,
  "from": 2304
}
```

```
>>> r = trans.TransactionIDRange(accountID=..., params=params)
>>> client.request(r)
>>> print r.response
```

Output:

```
"lastTransactionID": "2311",
"transactions": [
    "price": "1.20000",
    "stopLossOnFill": {
      "timeInForce": "GTC",
      "price": "1.22000"
    },
    "timeInForce": "GTC",
    "reason": "CLIENT_ORDER",
    "id": "2304",
    "batchID": "2304",
    "triggerCondition": "TRIGGER_DEFAULT",
    "positionFill": "DEFAULT",
    "userID": 1435156,
    "instrument": "EUR_USD",
    "time": "2016-10-24T21:48:18.593753865Z",
    "units": "-100",
    "type": "LIMIT ORDER",
    "accountID": "101-004-1435156-001"
  },
    "orderID": "2304",
    "batchID": "2305",
    "clientExtensionsModify": {
     "comment": "myComment",
      "id": "myID"
    "time": "2016-10-25T15:56:43.075594239Z",
    "type": "ORDER_CLIENT_EXTENSIONS_MODIFY",
    "userID": 1435156,
    "id": "2305",
    "accountID": "101-004-1435156-001"
  },
    "orderID": "2304",
    "clientOrderID": "myID",
    "reason": "CLIENT_REQUEST_REPLACED",
    "batchID": "2306",
    "time": "2016-10-25T19:45:38.558056359Z",
    "type": "ORDER_CANCEL",
    "replacedByOrderID": "2307",
    "userID": 1435156,
    "id": "2306",
    "accountID": "101-004-1435156-001"
]
```

3.8.3 TransactionList

Parameters

- accountID (string (required)) id of the account to perform the request on.
- params (dict (optional)) query params to send, check developer.oanda.com for details.

Query Params example:

```
{
    "pageSize": 200
}
```

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.transactions as trans
>>> client = oandapyV20.API(access_token=...)
>>> r = trans.TransactionList(accountID) # params optional
>>> client.request(r)
>>> print r.response
```

Output:

```
"count": 2124,
 "from": "2016-06-24T21:03:50.914647476Z",
 "lastTransactionID": "2124",
 "pageSize": 100,
 "to": "2016-10-05T06:54:14.025946546Z",
 "pages": [
   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/
→transactions/idrange?from=1&to=100",
   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/
→transactions/idrange?from=101&to=200",
   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/
→transactions/idrange?from=201&to=300",
   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/

→transactions/idrange?from=301&to=400",
   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/
→transactions/idrange?from=401&to=500",
   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/
→transactions/idrange?from=501&to=600",
   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/
→transactions/idrange?from-601&to-700",
                                                                 (continues on next page)
```

```
"https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/
→transactions/idrange?from=701&to=800",
   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/
→transactions/idrange?from=801&to=900",
   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/
→transactions/idrange?from=901&to=1000",
   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/

→transactions/idrange?from=1001&to=1100",

   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/
→transactions/idrange?from=1101&to=1200",
   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/
→transactions/idrange?from=1201&to=1300",
   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/

→transactions/idrange?from=1301&to=1400",

   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/
→transactions/idrange?from=1401&to=1500",
   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/
→transactions/idrange?from=1501&to=1600",
   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/
→transactions/idrange?from=1601&to=1700",
   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/
→transactions/idrange?from=1701&to=1800",
   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/
→transactions/idrange?from=1801&to=1900",
   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/
→transactions/idrange?from=1901&to=2000",
   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/
→transactions/idrange?from=2001&to=2100",
   "https://api-fxpractice.oanda.com/v3/accounts/101-004-1435156-001/
→transactions/idrange?from=2101&to=2124"
 1
```

3.8.4 TransactionsSinceID

Parameters

- account ID (string (required)) id of the account to perform the request on.
- params (dict (required)) query params to send, check developer.oanda.com for details.

Query Params example:

```
{
    "id": 2306
}
```

```
>>> r = trans.TransactionsSinceID(accountID=..., params=params)
>>> client.request(r)
>>> print r.response
```

Output:

```
"lastTransactionID": "2311",
"transactions": [
   "price": "1.25000",
   "timeInForce": "GTC",
    "reason": "REPLACEMENT",
    "clientExtensions": {
      "comment": "myComment",
      "id": "myID"
    "id": "2307",
    "batchID": "2306",
    "triggerCondition": "TRIGGER_DEFAULT",
    "replacesOrderID": "2304",
    "positionFill": "DEFAULT",
    "userID": 1435156,
    "instrument": "EUR_USD",
    "time": "2016-10-25T19:45:38.558056359Z",
    "units": "-500000",
    "type": "LIMIT_ORDER",
    "accountID": "101-004-1435156-001"
  },
    "orderID": "2307",
    "clientOrderID": "myID",
    "reason": "CLIENT_REQUEST",
    "batchID": "2308",
    "time": "2016-10-25T20:53:03.789670387Z",
    "type": "ORDER_CANCEL",
    "userID": 1435156,
    "id": "2308",
    "accountID": "101-004-1435156-001"
  },
    "price": "1.20000",
    "stopLossOnFill": {
```

```
"timeInForce": "GTC",
      "price": "1.22000"
    },
    "timeInForce": "GTC",
    "reason": "CLIENT_ORDER",
    "id": "2309",
    "batchID": "2309",
    "triggerCondition": "TRIGGER_DEFAULT",
    "positionFill": "DEFAULT",
    "userID": 1435156,
    "instrument": "EUR_USD",
    "time": "2016-10-25T21:07:21.065554321Z",
    "units": "-100",
    "type": "LIMIT_ORDER",
    "accountID": "101-004-1435156-001"
  },
    "userID": 1435156,
    "marginRate": "0.01",
    "batchID": "2310",
    "time": "2016-10-26T13:28:00.507651360Z",
    "type": "CLIENT_CONFIGURE",
    "id": "2310",
    "accountID": "101-004-1435156-001"
    "userID": 1435156,
    "marginRate": "0.01",
    "batchID": "2311",
    "time": "2016-10-26T13:28:13.597103123Z",
    "type": "CLIENT_CONFIGURE",
    "id": "2311",
    "accountID": "101-004-1435156-001"
1
```

3.8.5 TransactionsStream

Performing this request will result in a generator yielding transactions.

Parameters accountID (string (required)) - id of the account to perform the request on.

```
>>> import oandapyV20
>>> import oandapyV20.endpoints.transactions as trans
>>> client = oandapyV20.API(access_token=...)
>>> r = trans.TransactionsStream(accountID=...)
>>> rv = client.request(r)
>>> maxrecs = 5
>>> try:
        for T in r.response: # or rv ...
>>>
           print json.dumps(R, indent=4), ","
>>>
           maxrecs -= 1
>>>
           if maxrecs == 0:
>>>
>>>
               r.terminate("Got them all")
>>> except StreamTerminated as e:
      print("Finished: {msg}".format(msg=e))
>>>
```

Output:

```
{
        "type": "HEARTBEAT",
        "lastTransactionID": "2311",
        "time": "2016-10-28T11:56:12.002855862Z"
      },
        "type": "HEARTBEAT",
        "lastTransactionID": "2311",
        "time": "2016-10-28T11:56:17.059535527Z"
      },
      {
        "type": "HEARTBEAT",
        "lastTransactionID": "2311",
        "time": "2016-10-28T11:56:22.142256403Z"
      },
        "type": "HEARTBEAT",
        "lastTransactionID": "2311",
        "time": "2016-10-28T11:56:27.238853774Z"
      },
      {
        "type": "HEARTBEAT",
        "lastTransactionID": "2311",
        "time": "2016-10-28T11:56:32.289316796Z"
      }
Finished: Got them all
```

terminate (message=")

terminate the stream.

Calling this method will stop the generator yielding transaction records. A message can be passed optionally.

CHAPTER 4

oandapyV20.definitions

The oandapyV20.definitions module holds all the definitions as in the definitions section of the REST-V20 specs of OANDA, see developer.oanda.com.

4.1 oandapyV20.definitions.accounts

Account Definitions.

```
\begin{tabular}{ll} \textbf{class} & \texttt{oandapyV20.definitions.accounts.AccountFinancingMode} \\ & \textbf{Bases: object} \end{tabular}
```

Definition representation of AccountFinancingMode

```
DAILY = 'DAILY'

NO_FINANCING = 'NO_FINANCING'

SECOND_BY_SECOND = 'SECOND_BY_SECOND'

__getitem__ (definitionID)

return description for definitionID.
```

definitions

readonly property holding definition dict.

```
class oandapyV20.definitions.accounts.GuaranteedStopLossOrderMode
    Bases: object
```

Definition representation of GuaranteedStopLossOrderMode

Definitions used in requests and responses. This class provides the ID and the description of the definitions.

```
>>> import oandapyV20.definitions.accounts as defaccounts
>>> print defaccounts.GuaranteedStopLossOrderMode.DISABLED
DISABLED
>>> c = defaccounts.GuaranteedStopLossOrderMode()
>>> print c[c.DISABLED]
The account is not permitted to create guaranteed Stop Loss Orders.
>>> # or
>>> print defaccounts.GuaranteedStopLossOrderMode().definitions[c.DISABLED]
>>> # all keys
>>> print defaccounts.GuaranteedStopLossOrderMode().definitions.keys()
>>> ...
```

```
ALLOWED = 'ALLOWED'

DISABLED = 'DISABLED'

REQUIRED = 'REQUIRED'

__getitem__ (definitionID)

return description for definitionID.
```

definitions

readonly property holding definition dict.

```
class oandapyV20.definitions.accounts.PositionAggregationMode
    Bases: object
```

Definition representation of PositionAggregationMode

```
ABSOLUTE_SUM = 'ABSOLUTE_SUM'

MAXIMAL_SIDE = 'MAXIMAL_SIDE'

NET_SUM = 'NET_SUM'

__getitem__(definitionID)

return description for definitionID.
```

definitions

readonly property holding definition dict.

4.2 oandapyV20.definitions.instruments

Instruments Definitions.

Definition representation of CandlestickGranularity

```
>>> import oandapyV20.definitions.instruments as definstruments
>>> print definstruments.CandlestickGranularity.H4
H4
>>> c = definstruments.CandlestickGranularity()
>>> print c[c.H4]
4 hour candlesticks, day alignment
>>> # or
>>> print definstruments.CandlestickGranularity().definitions[c.H4]
>>> # all keys
>>> print definstruments.CandlestickGranularity().definitions.keys()
>>> ...
```

```
D = 'D'
H1 = 'H1'
H12 = 'H12'
H2 = 'H2'
H3 = 'H3'
H4 = 'H4'
H6 = 'H6'
H8 = 'H8'
M = 'M'
M1 = 'M1'
M10 = 'M10'
M15 = 'M15'
M2 = 'M2'
M30 = 'M30'
M4 = 'M4'
M5 = 'M5'
S10 = 'S10'
S15 = 'S15'
$30 = '$30'
```

```
S5 = 'S5'
W = 'W'
__getitem__(definitionID)
    return description for definitionID.

definitions
    readonly property holding definition dict.

class oandapyV20.definitions.instruments.WeeklyAlignment
    Bases: object
```

Definition representation of WeeklyAlignment

Definitions used in requests and responses. This class provides the ID and the description of the definitions.

```
>>> import oandapyV20.definitions.instruments as definstruments
>>> print definstruments.WeeklyAlignment.Monday
Monday
>>> c = definstruments.WeeklyAlignment()
>>> print c[c.Monday]
Monday
>>> # or
>>> print definstruments.WeeklyAlignment().definitions[c.Monday]
>>> # all keys
>>> print definstruments.WeeklyAlignment().definitions.keys()
>>> ...
```

```
Friday = 'Friday'
Monday = 'Monday'
Saturday = 'Saturday'
Sunday = 'Sunday'
Thursday = 'Thursday'
Tuesday = 'Tuesday'
Wednesday = 'Wednesday'
__getitem__(definitionID)
    return description for definitionID.
```

definitions

readonly property holding definition dict.

```
class oandapyV20.definitions.instruments.PriceComponents
    Bases: object
```

Definition representation of PriceComponents

Definitions used in requests and responses. This class provides the ID and the description of the definitions.

```
>>> import oandapyV20.definitions.instruments as definstruments
>>> print definstruments.PriceComponents.A
A
>>> c = definstruments.PriceComponents()
>>> print c[c.A]
Ask
>>> # or
>>> print definstruments.PriceComponents().definitions[c.A]
```

```
>>> # all keys
>>> print definstruments.PriceComponents().definitions.keys()
>>> ...

A = 'A'
B = 'B'
M = 'M'
__getitem__(definitionID)
    return description for definitionID.

definitions
    readonly property holding definition dict.
```

4.3 oandapyV20.definitions.orders

Order related definitions.

```
\begin{tabular}{ll} \textbf{class} & \texttt{oandapyV20.definitions.orders.OrderStateFilter} \\ & \textbf{Bases:} & \texttt{object} \end{tabular}
```

Definition representation of OrderStateFilter

Definitions used in requests and responses. This class provides the ID and the description of the definitions.

```
>>> import oandapyV20.definitions.orders as deforders
>>> print deforders.OrderStateFilter.CANCELLED
CANCELLED
>>> c = deforders.OrderStateFilter()
>>> print c[c.CANCELLED]
The orders that have been cancelled
>>> # or
>>> print deforders.OrderStateFilter().definitions[c.CANCELLED]
>>> # all keys
>>> print deforders.OrderStateFilter().definitions.keys()
>>> ...

ALL = 'ALL'
CANCELLED = 'CANCELLED'
FILLED = 'FILLED'
PENDING = 'PENDING'
TRIGGERED = 'TRIGGERED'
```

definitions

readonly property holding definition dict.

return description for definitionID.

```
class oandapyV20.definitions.orders.OrderType
    Bases: object
```

Definition representation of OrderType

<u>__getitem__</u>(definitionID)

Definitions used in requests and responses. This class provides the ID and the description of the definitions.

```
>>> import oandapyV20.definitions.orders as deforders
>>> print deforders.OrderType.MARKET_IF_TOUCHED
MARKET_IF_TOUCHED
>>> c = deforders.OrderType()
>>> print c[c.MARKET_IF_TOUCHED]
A Market-if-touched Order
>>> # or
>>> print deforders.OrderType().definitions[c.MARKET_IF_TOUCHED]
>>> print deforders.OrderType().definitions.keys()
FIXED PRICE = 'FIXED PRICE'
LIMIT = 'LIMIT'
MARKET = 'MARKET'
MARKET_IF_TOUCHED = 'MARKET_IF_TOUCHED'
STOP = 'STOP'
STOP_LOSS = 'STOP_LOSS'
TAKE PROFIT = 'TAKE_PROFIT'
TRAILING_STOP_LOSS = 'TRAILING_STOP_LOSS'
__getitem__(definitionID)
    return description for definitionID.
```

definitions

readonly property holding definition dict.

```
class oandapyV20.definitions.orders.CancellableOrderType
    Bases: object
```

Definition representation of CancellableOrderType

```
>>> import oandapyV20.definitions.orders as deforders
>>> print deforders.CancellableOrderType.MARKET_IF_TOUCHED
MARKET_IF_TOUCHED
>>> c = deforders.CancellableOrderType()
>>> print c[c.MARKET_IF_TOUCHED]
A Market-if-touched Order
>>> print deforders.CancellableOrderType().definitions[c.MARKET_IF_TOUCHED]
>>> # all keys
>>> print deforders.CancellableOrderType().definitions.keys()
>>> ...
```

```
LIMIT = 'LIMIT'
MARKET IF TOUCHED = 'MARKET IF TOUCHED'
STOP = 'STOP'
STOP_LOSS = 'STOP_LOSS'
TAKE PROFIT = 'TAKE PROFIT'
```

```
TRAILING_STOP_LOSS = 'TRAILING_STOP_LOSS'
    __getitem__ (definitionID)
    return description for definitionID.

definitions
    readonly property holding definition dict.

class oandapyV20.definitions.orders.OrderPositionFill
    Bases: object
```

Definition representation of OrderPositionFill

Definitions used in requests and responses. This class provides the ID and the description of the definitions.

```
>>> import oandapyV20.definitions.orders as deforders
>>> print deforders.OrderPositionFill.REDUCE_ONLY
REDUCE_ONLY
>>> c = deforders.OrderPositionFill()
>>> print c[c.REDUCE_ONLY]
When the Order is filled, only reduce an existing Position.
>>> # or
>>> print deforders.OrderPositionFill().definitions[c.REDUCE_ONLY]
>>> # all keys
>>> print deforders.OrderPositionFill().definitions.keys()
>>> ...
```

```
DEFAULT = 'DEFAULT'
OPEN_ONLY = 'OPEN_ONLY'
REDUCE_FIRST = 'REDUCE_FIRST'
REDUCE_ONLY = 'REDUCE_ONLY'
__getitem__(definitionID)
    return description for definitionID.
```

definitions

readonly property holding definition dict.

```
class oandapyV20.definitions.orders.TimeInForce
Bases: object
```

Definition representation of TimeInForce

```
>>> import oandapyV20.definitions.orders as deforders
>>> print deforders.TimeInForce.IOC
IOC
>>> c = deforders.TimeInForce()
>>> print c[c.IOC]
The Order must be "Immediately partially filled Or Killed"
>>> # or
>>> print deforders.TimeInForce().definitions[c.IOC]
>>> # all keys
>>> print deforders.TimeInForce().definitions.keys()
>>> ...
```

```
FOK = 'FOK'

GFD = 'GFD'
```

```
GTC = 'GTC'

GTD = 'GTD'

IOC = 'IOC'

__getitem__ (definitionID)
    return description for definitionID.

definitions
    readonly property holding definition dict.

class oandapyV20.definitions.orders.OrderState
    Bases: object
```

Definition representation of OrderState

Definitions used in requests and responses. This class provides the ID and the description of the definitions.

```
>>> import oandapyV20.definitions.orders as deforders
>>> print deforders.OrderState.CANCELLED
CANCELLED
>>> c = deforders.OrderState()
>>> print c[c.CANCELLED]
The Order has been cancelled
>>> # or
>>> print deforders.OrderState().definitions[c.CANCELLED]
>>> # all keys
>>> print deforders.OrderState().definitions.keys()
```

```
CANCELLED = 'CANCELLED'

FILLED = 'FILLED'

PENDING = 'PENDING'

TRIGGERED = 'TRIGGERED'

__getitem__ (definitionID)

return description for definitionID.
```

definitions

readonly property holding definition dict.

```
class oandapy V20. definitions.orders.OrderTriggerCondition Bases: object
```

Definition representation of OrderTriggerCondition

Definitions used in requests and responses. This class provides the ID and the description of the definitions.

```
>>> print deforders.OrderTriggerCondition().definitions.keys()
>>> ...

ASK = 'ASK'
BID = 'BID'

DEFAULT = 'DEFAULT'

INVERSE = 'INVERSE'

MID = 'MID'

__getitem__ (definitionID)
    return description for definitionID.

definitions
    readonly property holding definition dict.
```

4.4 oandapy V20. definitions. pricing

Pricing related Definitions.

```
\begin{tabular}{ll} \textbf{class} & \texttt{oandapyV20.definitions.pricing.PriceStatus} \\ \textbf{Bases:} & \texttt{object} \end{tabular}
```

Definition representation of PriceStatus

Definitions used in requests and responses. This class provides the ID and the description of the definitions.

```
>>> import oandapyV20.definitions.pricing as defpricing
>>> print defpricing.PriceStatus.non_tradeable
non-tradeable
>>> c = defpricing.PriceStatus()
>>> print c[c.non_tradeable]
The Instrument's price is not tradeable.
>>> # or
>>> print defpricing.PriceStatus().definitions[c.non_tradeable]
>>> # all keys
>>> print defpricing.PriceStatus().definitions.keys()
```

Note: attribute name *non-tradeable* is renamed to *non_tradeable*, value stil is *non-tradeable*. This means that a lookup stil applies.

```
__getitem__ (definitionID)
    return description for definitionID.

definitions
    readonly property holding definition dict.

invalid = 'invalid'

non_tradeable = 'non-tradeable'

tradeable = 'tradeable'
```

4.5 oandapyV20.definitions.trades

Trades definitions.

```
class oandapyV20.definitions.trades.TradePL
    Bases: object
```

Definition representation of TradePL

Definitions used in requests and responses. This class provides the ID and the description of the definitions.

```
NEGATIVE = 'NEGATIVE'
POSITIVE = 'POSITIVE'
ZERO = 'ZERO'
__getitem__(definitionID)
    return description for definitionID.
```

definitions

readonly property holding definition dict.

```
class oandapyV20.definitions.trades.TradeState
    Bases: object
```

Definition representation of TradeState

```
>>> import oandapyV20.definitions.trades as deftrades
>>> print deftrades.TradeState.CLOSE_WHEN_TRADABLE
CLOSE_WHEN_TRADABLE
>>> c = deftrades.TradeState()
>>> print c[c.CLOSE_WHEN_TRADABLE]
The Trade will be closed as soon as the trade's instrument becomes tradeable
>>> # or
>>> print deftrades.TradeState().definitions[c.CLOSE_WHEN_TRADABLE]
>>> # all keys
>>> print deftrades.TradeState().definitions.keys()
>>> ...
```

```
CLOSED = 'CLOSED'
CLOSE_WHEN_TRADABLE = 'CLOSE_WHEN_TRADABLE'
OPEN = 'OPEN'
```

```
__getitem__ (definitionID) return description for definitionID.
```

definitions

readonly property holding definition dict.

```
class oandapyV20.definitions.trades.TradeStateFilter
    Bases: object
```

Definition representation of TradeStateFilter

Definitions used in requests and responses. This class provides the ID and the description of the definitions.

```
>>> import oandapyV20.definitions.trades as deftrades
>>> print deftrades.TradeStateFilter.CLOSE_WHEN_TRADEABLE
CLOSE_WHEN_TRADEABLE
>>> c = deftrades.TradeStateFilter()
>>> print c[c.CLOSE_WHEN_TRADEABLE]
The Trades that will be closed as soon as the trades' instrument becomes tradeable
>>> # or
>>> print deftrades.TradeStateFilter().definitions[c.CLOSE_WHEN_TRADEABLE]
>>> # all keys
>>> print deftrades.TradeStateFilter().definitions.keys()
>>> ...

ALL = 'ALL'
CLOSED = 'CLOSED'
CLOSE_WHEN_TRADEABLE = 'CLOSE_WHEN_TRADEABLE'
OPEN = 'OPEN'
__getitem__(definitionID)
    return description for definitionID.
```

definitions

readonly property holding definition dict.

4.6 oandapyV20.definitions.transactions

Transactions definitions.

Definition representation of MarketOrderMarginCloseoutReason

Definitions used in requests and responses. This class provides the ID and the description of the definitions.

```
>>> print deftransactions.MarketOrderMarginCloseoutReason().definitions.keys()
>>> ...
```

```
MARGIN_CHECK_VIOLATION = 'MARGIN_CHECK_VIOLATION'
```

REGULATORY_MARGIN_CALL_VIOLATION = 'REGULATORY_MARGIN_CALL_VIOLATION'

```
__getitem__ (definitionID) return description for definitionID.
```

definitions

readonly property holding definition dict.

```
class oandapyV20.definitions.transactions.StopLossOrderReason
    Bases: object
```

Definition representation of StopLossOrderReason

Definitions used in requests and responses. This class provides the ID and the description of the definitions.

```
CLIENT_ORDER = 'CLIENT_ORDER'
ON_FILL = 'ON_FILL'

REPLACEMENT = 'REPLACEMENT'

__getitem__ (definitionID)

return description for definitionID.
```

definitions

readonly property holding definition dict.

```
class oandapyV20.definitions.transactions.OrderFillReason
    Bases: object
```

Definition representation of OrderFillReason

Definitions used in requests and responses. This class provides the ID and the description of the definitions.

```
>>> import oandapyV20.definitions.transactions as deftransactions
>>> print deftransactions.OrderFillReason.STOP_ORDER
STOP_ORDER
>>> c = deftransactions.OrderFillReason()
>>> print c[c.STOP_ORDER]
The Order filled was a Stop Order
>>> # or
>>> print deftransactions.OrderFillReason().definitions[c.STOP_ORDER]
>>> # all keys
```

```
>>> print deftransactions.OrderFillReason().definitions.keys()
    >>> . . .
    LIMIT_ORDER = 'LIMIT_ORDER'
    MARKET IF TOUCHED ORDER = 'MARKET IF TOUCHED ORDER'
    MARKET ORDER = 'MARKET ORDER'
    MARKET_ORDER_DELAYED_TRADE_CLOSE = 'MARKET_ORDER_DELAYED_TRADE_CLOSE'
    MARKET ORDER MARGIN CLOSEOUT = 'MARKET ORDER MARGIN CLOSEOUT'
    MARKET_ORDER_POSITION_CLOSEOUT = 'MARKET_ORDER_POSITION_CLOSEOUT'
    MARKET ORDER TRADE CLOSE = 'MARKET ORDER TRADE CLOSE'
    STOP_LOSS_ORDER = 'STOP_LOSS_ORDER'
    STOP_ORDER = 'STOP_ORDER'
    TAKE_PROFIT_ORDER = 'TAKE_PROFIT_ORDER'
    TRAILING_STOP_LOSS_ORDER = 'TRAILING_STOP_LOSS_ORDER'
    __getitem__(definitionID)
        return description for definitionID.
    definitions
        readonly property holding definition dict.
class oandapy V20.definitions.transactions.FundingReason
    Bases: object
    Definition representation of FundingReason
    Definitions used in requests and responses. This class provides the ID and the description of the definitions.
    >>> import oandapyV20.definitions.transactions as deftransactions
    >>> print deftransactions.FundingReason.ACCOUNT_TRANSFER
    ACCOUNT_TRANSFER
    >>> c = deftransactions.FundingReason()
    >>> print c[c.ACCOUNT_TRANSFER]
    Funds are being transfered between two Accounts.
    >>> print deftransactions.FundingReason().definitions[c.ACCOUNT_TRANSFER]
    >>> # all kevs
    >>> print deftransactions.FundingReason().definitions.keys()
    >>> ...
    ACCOUNT_TRANSFER = 'ACCOUNT_TRANSFER'
    ADJUSTMENT = 'ADJUSTMENT'
    CLIENT_FUNDING = 'CLIENT_FUNDING'
    DIVISION MIGRATION = 'DIVISION MIGRATION'
    SITE_MIGRATION = 'SITE_MIGRATION'
    __getitem__(definitionID)
        return description for definitionID.
```

definitions

readonly property holding definition dict.

 $\textbf{class} \ \, \text{oandapy V20.} definitions. \textbf{transactions.} \\ \textbf{MarketIfTouchedOrderReason} \\ \textbf{Bases:} \ \, \text{object} \\$

Definition representation of MarketIfTouchedOrderReason

Definitions used in requests and responses. This class provides the ID and the description of the definitions.

```
>>> import oandapyV20.definitions.transactions as deftransactions
>>> print deftransactions.MarketIfTouchedOrderReason.CLIENT_ORDER
CLIENT_ORDER
>>> c = deftransactions.MarketIfTouchedOrderReason()
>>> print c[c.CLIENT_ORDER]
The Market-if-touched Order was initiated at the request of a client
>>> # or
>>> print deftransactions.MarketIfTouchedOrderReason().definitions[c.CLIENT_ORDER]
>>> # all keys
>>> print deftransactions.MarketIfTouchedOrderReason().definitions.keys()
>>> ...
```

```
CLIENT_ORDER = 'CLIENT_ORDER'

REPLACEMENT = 'REPLACEMENT'

__getitem__(definitionID)

return description for definitionID.
```

definitions

readonly property holding definition dict.

```
class oandapyV20.definitions.transactions.FixedPriceOrderReason
    Bases: object
```

Definition representation of FixedPriceOrderReason

Definitions used in requests and responses. This class provides the ID and the description of the definitions.

```
PLATFORM_ACCOUNT_MIGRATION = 'PLATFORM_ACCOUNT_MIGRATION'
```

```
__getitem__ (definitionID)
return description for definitionID.
```

definitions

readonly property holding definition dict.

```
class oandapyV20.definitions.transactions.MarketOrderReason
    Bases: object
```

Definition representation of MarketOrderReason

Definitions used in requests and responses. This class provides the ID and the description of the definitions.

>>> import oandapyV20.definitions.transactions as deftransactions

```
>>> print deftransactions.MarketOrderReason.TRADE_CLOSE
    TRADE_CLOSE
    >>> c = deftransactions.MarketOrderReason()
    >>> print c[c.TRADE_CLOSE]
    The Market Order was created to close a Trade at the request of a client
    >>> # or
    >>> print deftransactions.MarketOrderReason().definitions[c.TRADE_CLOSE]
    >>> # all kevs
    >>> print deftransactions.MarketOrderReason().definitions.keys()
    CLIENT ORDER = 'CLIENT ORDER'
    DELAYED_TRADE_CLOSE = 'DELAYED_TRADE_CLOSE'
    MARGIN_CLOSEOUT = 'MARGIN_CLOSEOUT'
    POSITION_CLOSEOUT = 'POSITION_CLOSEOUT'
    TRADE_CLOSE = 'TRADE_CLOSE'
    __getitem__(definitionID)
         return description for definitionID.
    definitions
         readonly property holding definition dict.
class oandapyV20.definitions.transactions.StopOrderReason
    Bases: object
    Definition representation of StopOrderReason
    Definitions used in requests and responses. This class provides the ID and the description of the definitions.
    >>> import oandapyV20.definitions.transactions as deftransactions
    >>> print deftransactions.StopOrderReason.CLIENT_ORDER
    CLIENT_ORDER
    >>> c = deftransactions.StopOrderReason()
    >>> print c[c.CLIENT_ORDER]
    The Stop Order was initiated at the request of a client
    >>> # or
    >>> print deftransactions.StopOrderReason().definitions[c.CLIENT_ORDER]
    >>> # all keys
    >>> print deftransactions.StopOrderReason().definitions.keys()
    >>> ...
    CLIENT_ORDER = 'CLIENT_ORDER'
    REPLACEMENT = 'REPLACEMENT'
    __getitem__(definitionID)
        return description for definitionID.
    definitions
         readonly property holding definition dict.
class oandapyV20.definitions.transactions.TransactionType
    Bases: object
```

Definition representation of TransactionType

```
>>> import oandapyV20.definitions.transactions as deftransactions
>>> print deftransactions.TransactionType.STOP_LOSS_ORDER
STOP_LOSS_ORDER
>>> c = deftransactions.TransactionType()
>>> print c[c.STOP_LOSS_ORDER]
Stop Loss Order Transaction
>>> # or
>>> print deftransactions.TransactionType().definitions[c.STOP_LOSS_ORDER]
>>> # all kevs
>>> print deftransactions.TransactionType().definitions.keys()
CLIENT CONFIGURE = 'CLIENT CONFIGURE'
CLIENT_CONFIGURE_REJECT = 'CLIENT_CONFIGURE_REJECT'
CLOSE = 'CLOSE'
CREATE = 'CREATE'
DAILY_FINANCING = 'DAILY_FINANCING'
DELAYED_TRADE_CLOSURE = 'DELAYED_TRADE_CLOSURE'
FIXED_PRICE_ORDER = 'FIXED_PRICE_ORDER'
LIMIT ORDER = 'LIMIT ORDER'
LIMIT_ORDER_REJECT = 'LIMIT_ORDER_REJECT'
MARGIN_CALL_ENTER = 'MARGIN_CALL_ENTER'
MARGIN CALL EXIT = 'MARGIN CALL EXIT'
MARGIN CALL EXTEND = 'MARGIN CALL EXTEND'
MARKET_IF_TOUCHED_ORDER = 'MARKET_IF_TOUCHED_ORDER'
MARKET_IF_TOUCHED_ORDER_REJECT = 'MARKET_IF_TOUCHED_ORDER_REJECT'
MARKET_ORDER = 'MARKET_ORDER'
MARKET ORDER REJECT = 'MARKET ORDER REJECT'
ORDER_CANCEL = 'ORDER_CANCEL'
ORDER_CANCEL_REJECT = 'ORDER_CANCEL_REJECT'
ORDER CLIENT EXTENSIONS MODIFY = 'ORDER CLIENT EXTENSIONS MODIFY'
ORDER_CLIENT_EXTENSIONS_MODIFY_REJECT = 'ORDER_CLIENT_EXTENSIONS_MODIFY_REJECT'
ORDER FILL = 'ORDER FILL'
REOPEN = 'REOPEN'
RESET RESETTABLE PL = 'RESET RESETTABLE PL'
STOP_LOSS_ORDER = 'STOP_LOSS_ORDER'
STOP_LOSS_ORDER_REJECT = 'STOP_LOSS_ORDER_REJECT'
STOP_ORDER = 'STOP_ORDER'
```

```
STOP_ORDER_REJECT = 'STOP_ORDER_REJECT'
    TAKE PROFIT ORDER = 'TAKE PROFIT ORDER'
    TAKE_PROFIT_ORDER_REJECT = 'TAKE_PROFIT_ORDER_REJECT'
    TRADE_CLIENT_EXTENSIONS_MODIFY = 'TRADE_CLIENT_EXTENSIONS_MODIFY'
    TRADE CLIENT EXTENSIONS MODIFY REJECT = 'TRADE CLIENT EXTENSIONS MODIFY REJECT'
    TRAILING STOP LOSS ORDER = 'TRAILING STOP LOSS ORDER'
    TRAILING_STOP_LOSS_ORDER_REJECT = 'TRAILING_STOP_LOSS_ORDER_REJECT'
    TRANSFER_FUNDS = 'TRANSFER_FUNDS'
    TRANSFER FUNDS REJECT = 'TRANSFER FUNDS REJECT'
    __getitem__(definitionID)
        return description for definitionID.
    definitions
        readonly property holding definition dict.
class oandapyV20.definitions.transactions.TakeProfitOrderReason
    Bases: object
    Definition representation of TakeProfitOrderReason
    Definitions used in requests and responses. This class provides the ID and the description of the definitions.
    >>> import oandapyV20.definitions.transactions as deftransactions
    >>> print deftransactions.TakeProfitOrderReason.ON_FILL
    ON_FILL
    >>> c = deftransactions.TakeProfitOrderReason()
    >>> print c[c.ON_FILL]
    The Take Profit Order was initiated automatically when an Order was filled that,
    →opened a new Trade requiring a Take Profit Order.
    >>> # or
    >>> print deftransactions.TakeProfitOrderReason().definitions[c.ON_FILL]
    >>> # all kevs
    >>> print deftransactions.TakeProfitOrderReason().definitions.keys()
    >>> ...
    CLIENT ORDER = 'CLIENT ORDER'
    ON FILL = 'ON FILL'
    REPLACEMENT = 'REPLACEMENT'
```

__getitem__(definitionID)

return description for definitionID.

definitions

readonly property holding definition dict.

class oandapy V20. definitions.transactions.TransactionRejectReason Bases: object

Definition representation of TransactionRejectReason

```
>>> import oandapyV20.definitions.transactions as deftransactions
>>> print deftransactions.TransactionRejectReason.STOP_LOSS_ORDER_GUARANTEED_
→PRICE WITHIN SPREAD
STOP_LOSS_ORDER_GUARANTEED_PRICE_WITHIN_SPREAD
>>> c = deftransactions.TransactionRejectReason()
>>> print c[c.STOP_LOSS_ORDER_GUARANTEED_PRICE_WITHIN_SPREAD]
An attempt to create a guaranteed stop loss order with a price that is within the,
⇔current tradeable spread.
>>> # or
>>> print deftransactions.TransactionRejectReason().definitions[c.STOP_LOSS_ORDER_
→GUARANTEED_PRICE_WITHIN_SPREAD]
>>> # all keys
>>> print deftransactions.TransactionRejectReason().definitions.keys()
ACCOUNT CONFIGURATION LOCKED = 'ACCOUNT CONFIGURATION LOCKED'
ACCOUNT_DEPOSIT_LOCKED = 'ACCOUNT_DEPOSIT_LOCKED'
ACCOUNT_LOCKED = 'ACCOUNT_LOCKED'
ACCOUNT_NOT_ACTIVE = 'ACCOUNT_NOT_ACTIVE'
ACCOUNT_ORDER_CANCEL_LOCKED = 'ACCOUNT_ORDER_CANCEL_LOCKED'
ACCOUNT_ORDER_CREATION_LOCKED = 'ACCOUNT_ORDER_CREATION_LOCKED'
ACCOUNT_WITHDRAWAL_LOCKED = 'ACCOUNT_WITHDRAWAL_LOCKED'
ADMIN CONFIGURE DATA MISSING = 'ADMIN CONFIGURE DATA MISSING'
ALIAS INVALID = 'ALIAS INVALID'
AMOUNT_INVALID = 'AMOUNT_INVALID'
AMOUNT MISSING = 'AMOUNT MISSING'
CLIENT CONFIGURE DATA MISSING = 'CLIENT CONFIGURE DATA MISSING'
CLIENT_EXTENSIONS_DATA_MISSING = 'CLIENT_EXTENSIONS_DATA_MISSING'
CLIENT_ORDER_COMMENT_INVALID = 'CLIENT_ORDER_COMMENT_INVALID'
CLIENT ORDER ID ALREADY EXISTS = 'CLIENT ORDER ID ALREADY EXISTS'
CLIENT ORDER ID INVALID = 'CLIENT ORDER ID INVALID'
CLIENT_ORDER_TAG_INVALID = 'CLIENT_ORDER_TAG_INVALID'
CLIENT_TRADE_COMMENT_INVALID = 'CLIENT_TRADE_COMMENT_INVALID'
CLIENT TRADE ID ALREADY EXISTS = 'CLIENT TRADE ID ALREADY EXISTS'
CLIENT TRADE ID INVALID = 'CLIENT TRADE ID INVALID'
CLIENT TRADE TAG INVALID = 'CLIENT TRADE TAG INVALID'
CLOSEOUT_POSITION_DOESNT_EXIST = 'CLOSEOUT_POSITION_DOESNT_EXIST'
CLOSEOUT POSITION INCOMPLETE SPECIFICATION = 'CLOSEOUT POSITION INCOMPLETE SPECIFICATI
CLOSEOUT POSITION PARTIAL UNITS MISSING = 'CLOSEOUT POSITION PARTIAL UNITS MISSING'
CLOSEOUT POSITION REJECT = 'CLOSEOUT POSITION REJECT'
CLOSEOUT_POSITION_UNITS_EXCEED_POSITION_SIZE = 'CLOSEOUT_POSITION_UNITS_EXCEED_POSITIO
```

```
CLOSE TRADE PARTIAL UNITS MISSING = 'CLOSE TRADE PARTIAL UNITS MISSING'
CLOSE TRADE TYPE MISSING = 'CLOSE TRADE TYPE MISSING'
CLOSE_TRADE_UNITS_EXCEED_TRADE_SIZE = 'CLOSE_TRADE_UNITS_EXCEED_TRADE_SIZE'
FUNDING_REASON_MISSING = 'FUNDING_REASON_MISSING'
INSTRUMENT MISSING = 'INSTRUMENT MISSING'
INSTRUMENT NOT TRADEABLE = 'INSTRUMENT NOT TRADEABLE'
INSTRUMENT_PRICE_UNKNOWN = 'INSTRUMENT_PRICE_UNKNOWN'
INSTRUMENT_UNKNOWN = 'INSTRUMENT_UNKNOWN'
INSUFFICIENT FUNDS = 'INSUFFICIENT FUNDS'
INSUFFICIENT MARGIN = 'INSUFFICIENT MARGIN'
INTERNAL_SERVER_ERROR = 'INTERNAL_SERVER_ERROR'
INVALID_REISSUE_IMMEDIATE_PARTIAL_FILL = 'INVALID_REISSUE_IMMEDIATE_PARTIAL_FILL'
MARGIN RATE INVALID = 'MARGIN RATE INVALID'
MARGIN_RATE_WOULD_TRIGGER_CLOSEOUT = 'MARGIN_RATE_WOULD_TRIGGER_CLOSEOUT'
MARGIN RATE WOULD TRIGGER MARGIN CALL = 'MARGIN RATE WOULD TRIGGER MARGIN CALL'
MARKUP_GROUP_ID_INVALID = 'MARKUP_GROUP_ID_INVALID'
ORDERS ON FILL DUPLICATE CLIENT ORDER IDS = 'ORDERS ON FILL DUPLICATE CLIENT ORDER IDS
ORDER DOESNT EXIST = 'ORDER DOESNT EXIST'
ORDER FILL POSITION_ACTION_INVALID = 'ORDER_FILL_POSITION_ACTION_INVALID'
ORDER FILL POSITION ACTION MISSING = 'ORDER FILL POSITION ACTION MISSING'
ORDER IDENTIFIER INCONSISTENCY = 'ORDER IDENTIFIER INCONSISTENCY'
ORDER_ID_UNSPECIFIED = 'ORDER_ID_UNSPECIFIED'
ORDER_PARTIAL_FILL_OPTION_INVALID = 'ORDER_PARTIAL_FILL_OPTION_INVALID'
ORDER PARTIAL FILL OPTION MISSING = 'ORDER PARTIAL FILL OPTION MISSING'
PENDING ORDERS ALLOWED EXCEEDED = 'PENDING ORDERS ALLOWED EXCEEDED'
POSITION AGGREGATION MODE INVALID = 'POSITION AGGREGATION MODE INVALID'
PRICE_BOUND_INVALID = 'PRICE_BOUND_INVALID'
PRICE BOUND PRECISION EXCEEDED = 'PRICE BOUND PRECISION EXCEEDED'
PRICE DISTANCE INVALID = 'PRICE DISTANCE INVALID'
PRICE_DISTANCE_MAXIMUM_EXCEEDED = 'PRICE_DISTANCE_MAXIMUM_EXCEEDED'
PRICE_DISTANCE_MINIMUM_NOT_MET = 'PRICE_DISTANCE_MINIMUM_NOT_MET'
PRICE_DISTANCE_MISSING = 'PRICE_DISTANCE_MISSING'
PRICE_DISTANCE_PRECISION_EXCEEDED = 'PRICE_DISTANCE_PRECISION_EXCEEDED'
PRICE INVALID = 'PRICE INVALID'
PRICE_MISSING = 'PRICE_MISSING'
PRICE PRECISION EXCEEDED = 'PRICE PRECISION EXCEEDED'
```

```
REPLACING ORDER INVALID = 'REPLACING ORDER INVALID'
REPLACING TRADE ID INVALID = 'REPLACING TRADE ID INVALID'
STOP_LOSS_ON_FILL_CLIENT_ORDER_COMMENT_INVALID = 'STOP_LOSS_ON_FILL_CLIENT_ORDER_COMME
STOP_LOSS_ON_FILL_CLIENT_ORDER_ID_INVALID = 'STOP_LOSS_ON_FILL_CLIENT_ORDER_ID_INVALID
STOP LOSS ON FILL CLIENT ORDER TAG INVALID = 'STOP LOSS ON FILL CLIENT ORDER TAG INVAL
STOP LOSS ON FILL DISTANCE INVALID = 'STOP LOSS ON FILL DISTANCE INVALID'
STOP_LOSS_ON_FILL_DISTANCE_PRECISION_EXCEEDED = 'STOP_LOSS_ON_FILL_DISTANCE_PRECISION |
STOP_LOSS_ON_FILL_GTD_TIMESTAMP_IN_PAST = 'STOP_LOSS_ON_FILL_GTD_TIMESTAMP_IN_PAST'
STOP_LOSS_ON_FILL_GTD_TIMESTAMP_MISSING = 'STOP_LOSS_ON_FILL_GTD_TIMESTAMP_MISSING'
STOP_LOSS_ON_FILL_GUARANTEED_LEVEL_RESTRICTION_EXCEEDED = 'STOP_LOSS_ON_FILL_GUARANTEE
STOP_LOSS_ON_FILL_GUARANTEED_MINIMUM_DISTANCE_NOT_MET = 'STOP_LOSS_ON_FILL_GUARANTEED_!
STOP_LOSS_ON_FILL_GUARANTEED_NOT_ALLOWED = 'STOP_LOSS_ON_FILL_GUARANTEED_NOT_ALLOWED'
STOP_LOSS_ON_FILL_GUARANTEED_REQUIRED = 'STOP_LOSS_ON_FILL_GUARANTEED_REQUIRED'
STOP LOSS ON FILL PRICE AND DISTANCE BOTH MISSING = 'STOP LOSS ON FILL PRICE AND DISTA
STOP_LOSS_ON_FILL_PRICE_AND_DISTANCE_BOTH_SPECIFIED = 'STOP_LOSS_ON_FILL_PRICE_AND_DIS
STOP LOSS ON FILL PRICE DISTANCE MAXIMUM EXCEEDED = 'STOP LOSS ON FILL PRICE DISTANCE !
STOP LOSS ON FILL PRICE INVALID = 'STOP LOSS ON FILL PRICE INVALID'
STOP LOSS ON FILL PRICE MISSING = 'STOP LOSS ON FILL PRICE MISSING'
STOP_LOSS_ON_FILL_PRICE_PRECISION_EXCEEDED = 'STOP_LOSS_ON_FILL_PRICE_PRECISION_EXCEED
STOP LOSS ON FILL REQUIRED FOR PENDING ORDER = 'STOP LOSS ON FILL REQUIRED FOR PENDING
STOP_LOSS_ON_FILL_TIME_IN_FORCE_INVALID = 'STOP_LOSS_ON_FILL_TIME_IN_FORCE_INVALID'
STOP_LOSS_ON_FILL_TIME_IN_FORCE_MISSING = 'STOP_LOSS_ON_FILL_TIME_IN_FORCE_MISSING'
STOP_LOSS_ON_FILL_TRIGGER_CONDITION_INVALID = 'STOP_LOSS_ON_FILL_TRIGGER_CONDITION_INV
STOP LOSS ON FILL TRIGGER CONDITION MISSING = 'STOP LOSS ON FILL TRIGGER CONDITION MIS
STOP LOSS ORDER ALREADY EXISTS = 'STOP LOSS ORDER ALREADY EXISTS'
STOP_LOSS_ORDER_GUARANTEED_HALTED_CREATE_VIOLATION = 'STOP_LOSS_ORDER_GUARANTEED_HALTE
STOP_LOSS_ORDER_GUARANTEED_HALTED_TIGHTEN_VIOLATION = 'STOP_LOSS_ORDER_GUARANTEED_HALT.
STOP LOSS ORDER GUARANTEED HEDGING NOT ALLOWED = 'STOP LOSS ORDER GUARANTEED HEDGING N
STOP LOSS ORDER GUARANTEED LEVEL RESTRICTION EXCEEDED = 'STOP LOSS ORDER GUARANTEED LE
STOP_LOSS_ORDER_GUARANTEED_MINIMUM_DISTANCE_NOT_MET = 'STOP_LOSS_ORDER_GUARANTEED_MINI
STOP_LOSS_ORDER_GUARANTEED_NOT_ALLOWED = 'STOP_LOSS_ORDER_GUARANTEED_NOT_ALLOWED'
STOP_LOSS_ORDER_GUARANTEED_PRICE_WITHIN_SPREAD = 'STOP_LOSS_ORDER_GUARANTEED_PRICE_WIT
STOP LOSS ORDER GUARANTEED REQUIRED = 'STOP LOSS ORDER GUARANTEED REQUIRED'
STOP LOSS ORDER NOT CANCELABLE = 'STOP LOSS ORDER NOT CANCELABLE'
STOP_LOSS_ORDER_NOT_REPLACEABLE = 'STOP_LOSS_ORDER_NOT_REPLACEABLE'
STOP LOSS ORDER PRICE AND DISTANCE BOTH MISSING = 'STOP LOSS ORDER PRICE AND DISTANCE !
```

```
TAKE PROFIT ON FILL CLIENT ORDER COMMENT INVALID = 'TAKE PROFIT ON FILL CLIENT ORDER C
TAKE_PROFIT_ON_FILL_CLIENT_ORDER_ID_INVALID = 'TAKE_PROFIT_ON_FILL_CLIENT_ORDER_ID_INV
TAKE_PROFIT_ON_FILL_CLIENT_ORDER_TAG_INVALID = 'TAKE_PROFIT_ON_FILL_CLIENT_ORDER_TAG_I
TAKE PROFIT ON FILL GTD TIMESTAMP IN PAST = 'TAKE PROFIT ON FILL GTD TIMESTAMP IN PAST
TAKE PROFIT ON FILL GTD TIMESTAMP MISSING = 'TAKE PROFIT ON FILL GTD TIMESTAMP MISSING
TAKE_PROFIT_ON_FILL_PRICE_INVALID = 'TAKE_PROFIT_ON_FILL_PRICE_INVALID'
TAKE PROFIT ON_FILL PRICE_MISSING = 'TAKE_PROFIT_ON_FILL PRICE_MISSING'
TAKE PROFIT ON FILL PRICE PRECISION EXCEEDED = 'TAKE PROFIT ON FILL PRICE PRECISION EX
TAKE PROFIT ON FILL TIME IN FORCE INVALID = 'TAKE PROFIT ON FILL TIME IN FORCE INVALID
TAKE PROFIT ON FILL TIME IN FORCE MISSING = 'TAKE PROFIT ON FILL TIME IN FORCE MISSING
TAKE PROFIT ON FILL TRIGGER CONDITION INVALID = 'TAKE PROFIT ON FILL TRIGGER CONDITION
TAKE PROFIT ON FILL TRIGGER CONDITION MISSING = 'TAKE PROFIT ON FILL TRIGGER CONDITION
TAKE_PROFIT_ORDER_ALREADY_EXISTS = 'TAKE_PROFIT_ORDER_ALREADY_EXISTS'
TIME_IN_FORCE_GTD_TIMESTAMP_IN_PAST = 'TIME_IN_FORCE_GTD_TIMESTAMP_IN_PAST'
TIME IN FORCE GTD TIMESTAMP MISSING = 'TIME IN FORCE GTD TIMESTAMP MISSING'
TIME IN FORCE INVALID = 'TIME IN FORCE INVALID'
TIME IN FORCE MISSING = 'TIME IN FORCE MISSING'
TRADE_DOESNT_EXIST = 'TRADE_DOESNT_EXIST'
TRADE IDENTIFIER INCONSISTENCY = 'TRADE IDENTIFIER INCONSISTENCY'
TRADE_ID_UNSPECIFIED = 'TRADE_ID_UNSPECIFIED'
TRADE_ON_FILL_CLIENT_EXTENSIONS_NOT_SUPPORTED = 'TRADE_ON_FILL_CLIENT_EXTENSIONS_NOT_S
TRAILING_STOP_LOSS_ON_FILL_CLIENT_ORDER_COMMENT_INVALID = 'TRAILING_STOP_LOSS_ON_FILL_
TRAILING_STOP_LOSS_ON_FILL_CLIENT_ORDER_ID_INVALID = 'TRAILING_STOP_LOSS_ON_FILL_CLIEN
TRAILING STOP LOSS ON FILL CLIENT ORDER TAG INVALID = 'TRAILING STOP LOSS ON FILL CLIE
TRAILING_STOP_LOSS_ON_FILL_GTD_TIMESTAMP_IN_PAST = 'TRAILING_STOP_LOSS_ON_FILL_GTD_TIM
TRAILING_STOP_LOSS_ON_FILL_GTD_TIMESTAMP_MISSING = 'TRAILING_STOP_LOSS_ON_FILL_GTD_TIM
TRAILING STOP LOSS ON FILL PRICE DISTANCE INVALID = 'TRAILING STOP LOSS ON FILL PRICE'
TRAILING_STOP_LOSS_ON_FILL_PRICE_DISTANCE_MAXIMUM_EXCEEDED = 'TRAILING_STOP_LOSS_ON_FI
TRAILING_STOP_LOSS_ON_FILL_PRICE_DISTANCE MINIMUM_NOT_MET = 'TRAILING_STOP_LOSS_ON_FIL
TRAILING_STOP_LOSS_ON_FILL_PRICE_DISTANCE_MISSING = 'TRAILING_STOP_LOSS_ON_FILL_PRICE_
TRAILING STOP_LOSS_ON_FILL_PRICE_DISTANCE_PRECISION_EXCEEDED = 'TRAILING_STOP_LOSS_ON_
TRAILING_STOP_LOSS_ON_FILL_TIME_IN_FORCE_INVALID = 'TRAILING_STOP_LOSS_ON_FILL_TIME_IN
TRAILING_STOP_LOSS_ON_FILL_TIME_IN_FORCE_MISSING = 'TRAILING_STOP_LOSS_ON_FILL_TIME_IN
TRAILING_STOP_LOSS_ON_FILL_TRIGGER_CONDITION_INVALID = 'TRAILING_STOP_LOSS_ON_FILL_TRI
TRAILING_STOP_LOSS_ON_FILL_TRIGGER_CONDITION_MISSING = 'TRAILING_STOP_LOSS_ON_FILL_TRI
```

STOP LOSS ORDER PRICE AND DISTANCE BOTH SPECIFIED = 'STOP LOSS ORDER PRICE AND DISTANC

```
TRAILING_STOP_LOSS_ORDERS_NOT_SUPPORTED = 'TRAILING_STOP_LOSS_ORDERS_NOT_SUPPORTED'
    TRAILING STOP LOSS ORDER ALREADY EXISTS = 'TRAILING STOP LOSS ORDER ALREADY EXISTS'
    TRIGGER_CONDITION_INVALID = 'TRIGGER_CONDITION_INVALID'
    TRIGGER_CONDITION_MISSING = 'TRIGGER_CONDITION_MISSING'
    UNITS INVALID = 'UNITS INVALID'
    UNITS LIMIT EXCEEDED = 'UNITS LIMIT EXCEEDED'
    UNITS_MIMIMUM_NOT_MET = 'UNITS_MIMIMUM_NOT_MET'
    UNITS_MISSING = 'UNITS_MISSING'
    UNITS_PRECISION_EXCEEDED = 'UNITS_PRECISION_EXCEEDED'
    <u>__getitem__</u>(definitionID)
        return description for definitionID.
    definitions
        readonly property holding definition dict.
class oandapyV20.definitions.transactions.OrderCancelReason
    Bases: object
    Definition representation of OrderCancelReason
    Definitions used in requests and responses. This class provides the ID and the description of the definitions.
    >>> import oandapyV20.definitions.transactions as deftransactions
    >>> print deftransactions.OrderCancelReason.TAKE_PROFIT_ON_FILL_GTD_TIMESTAMP_IN_
     ⊶PAST
    TAKE_PROFIT_ON_FILL_GTD_TIMESTAMP_IN_PAST
    >>> c = deftransactions.OrderCancelReason()
    >>> print c[c.TAKE_PROFIT_ON_FILL_GTD_TIMESTAMP_IN_PAST]
    Filling the Order would have resulted in the creation of a Take Profit Order with,
    \rightarrowa GTD time in the past.
    >>> # or
    >>> print deftransactions.OrderCancelReason().definitions[c.TAKE_PROFIT_ON_FILL_
    →GTD_TIMESTAMP_IN_PAST]
    >>> # all kevs
    >>> print deftransactions.OrderCancelReason().definitions.keys()
    >>> . . .
    ACCOUNT LOCKED = 'ACCOUNT LOCKED'
    ACCOUNT_NEW_POSITIONS_LOCKED = 'ACCOUNT_NEW_POSITIONS_LOCKED'
    ACCOUNT ORDER CREATION LOCKED = 'ACCOUNT ORDER CREATION LOCKED'
    ACCOUNT ORDER FILL LOCKED = 'ACCOUNT ORDER FILL LOCKED'
    ACCOUNT POSITION VALUE LIMIT EXCEEDED = 'ACCOUNT POSITION VALUE LIMIT EXCEEDED'
    BOUNDS VIOLATION = 'BOUNDS VIOLATION'
    CLIENT_REQUEST = 'CLIENT_REQUEST'
    CLIENT_REQUEST_REPLACED = 'CLIENT_REQUEST_REPLACED'
    CLIENT TRADE ID ALREADY EXISTS = 'CLIENT TRADE ID ALREADY EXISTS'
    FIFO_VIOLATION = 'FIFO_VIOLATION'
    HEDGING_GSLO_VIOLATION = 'HEDGING_GSLO_VIOLATION'
```

```
INSUFFICIENT LIQUIDITY = 'INSUFFICIENT LIQUIDITY'
INSUFFICIENT MARGIN = 'INSUFFICIENT MARGIN'
INTERNAL_SERVER_ERROR = 'INTERNAL_SERVER_ERROR'
LINKED_TRADE_CLOSED = 'LINKED_TRADE_CLOSED'
LOSING TAKE PROFIT = 'LOSING TAKE PROFIT'
MARKET HALTED = 'MARKET HALTED'
MIGRATION = 'MIGRATION'
OPEN_TRADES_ALLOWED_EXCEEDED = 'OPEN_TRADES_ALLOWED_EXCEEDED'
PENDING_ORDERS_ALLOWED_EXCEEDED = 'PENDING_ORDERS_ALLOWED_EXCEEDED'
POSITION CLOSEOUT FAILED = 'POSITION CLOSEOUT FAILED'
POSITION_SIZE_EXCEEDED = 'POSITION_SIZE_EXCEEDED'
STOP_LOSS_ON_FILL_CLIENT_ORDER_ID_ALREADY_EXISTS = 'STOP_LOSS_ON_FILL_CLIENT_ORDER_ID_.
STOP_LOSS_ON_FILL_GTD_TIMESTAMP_IN_PAST = 'STOP_LOSS_ON_FILL_GTD_TIMESTAMP_IN_PAST'
STOP LOSS ON FILL GUARANTEED HEDGING NOT ALLOWED = 'STOP LOSS ON FILL GUARANTEED HEDGI
STOP LOSS ON FILL GUARANTEED LEVEL RESTRICTION EXCEEDED = 'STOP LOSS ON FILL GUARANTEE
STOP LOSS ON FILL GUARANTEED MINIMUM DISTANCE NOT MET = 'STOP LOSS ON FILL GUARANTEED !
STOP LOSS ON FILL GUARANTEED NOT ALLOWED = 'STOP LOSS ON FILL GUARANTEED NOT ALLOWED'
STOP_LOSS_ON_FILL_GUARANTEED_REQUIRED = 'STOP_LOSS_ON_FILL_GUARANTEED_REQUIRED'
STOP_LOSS_ON_FILL_LOSS = 'STOP_LOSS_ON_FILL_LOSS'
STOP LOSS ON FILL PRICE DISTANCE MAXIMUM EXCEEDED = 'STOP LOSS ON FILL PRICE DISTANCE !
STOP_LOSS_ON_FILL_REQUIRED = 'STOP_LOSS_ON_FILL_REQUIRED'
STOP_LOSS_ON_FILL_TIME_IN_FORCE_INVALID = 'STOP_LOSS_ON_FILL_TIME_IN_FORCE_INVALID'
STOP_LOSS_ON_FILL_TRIGGER_CONDITION_INVALID = 'STOP_LOSS_ON_FILL_TRIGGER_CONDITION_INV
TAKE PROFIT ON FILL CLIENT ORDER ID ALREADY EXISTS = 'TAKE PROFIT ON FILL CLIENT ORDER
TAKE PROFIT ON FILL GTD TIMESTAMP IN PAST = 'TAKE PROFIT ON FILL GTD TIMESTAMP IN PAST
TAKE PROFIT ON FILL LOSS = 'TAKE PROFIT ON FILL LOSS'
TAKE_PROFIT_ON_FILL_PRICE_DISTANCE_MAXIMUM_EXCEEDED = 'TAKE_PROFIT_ON_FILL_PRICE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DISTANCE_DIS
TIME IN FORCE EXPIRED = 'TIME IN FORCE EXPIRED'
TRAILING_STOP_LOSS_ON_FILL_CLIENT_ORDER_ID_ALREADY_EXISTS = 'TRAILING_STOP_LOSS_ON_FIL
TRAILING_STOP_LOSS_ON_FILL_GTD_TIMESTAMP_IN_PAST = 'TRAILING_STOP_LOSS_ON_FILL_GTD_TIM
__getitem__(definitionID)
       return description for definitionID.
definitions
       readonly property holding definition dict.
```

 $\textbf{class} \ \ \textbf{oandapy} \ \textbf{V20.} \\ \textbf{definitions.transactions.} \\ \textbf{TrailingStopLossOrderReason} \\ \textbf{Bases:} \ \textbf{object}$

Definition representation of TrailingStopLossOrderReason

Definitions used in requests and responses. This class provides the ID and the description of the definitions.

```
CLIENT_ORDER = 'CLIENT_ORDER'
ON_FILL = 'ON_FILL'
REPLACEMENT = 'REPLACEMENT'
__getitem__(definitionID)
    return description for definitionID.
```

definitions

readonly property holding definition dict.

```
class oandapy V20. definitions.transactions.LimitOrderReason Bases: object
```

Definition representation of LimitOrderReason

Definitions used in requests and responses. This class provides the ID and the description of the definitions.

```
>>> import oandapyV20.definitions.transactions as deftransactions
>>> print deftransactions.LimitOrderReason.CLIENT_ORDER
CLIENT_ORDER
>>> c = deftransactions.LimitOrderReason()
>>> print c[c.CLIENT_ORDER]
The Limit Order was initiated at the request of a client
>>> # or
>>> print deftransactions.LimitOrderReason().definitions[c.CLIENT_ORDER]
>>> # all keys
>>> print deftransactions.LimitOrderReason().definitions.keys()
>>> ...
```

```
CLIENT_ORDER = 'CLIENT_ORDER'

REPLACEMENT = 'REPLACEMENT'

__getitem___(definitionID)
    return description for definitionID.

definitions
```

readonly property holding definition dict.

CHAPTER 5

oandapyV20.types

The oandapyV20.types module contains the types representing the types that are used in the API-specs of OANDA, check developer.oanda.com. These types offer a convenient interface between Python types and the types used in the REST-API.

Take for instance the *PriceValue* type. It is the string representation of a float.

```
from oandapyV20.types import PriceValue

pv1 = PriceValue(122.345)
pv2 = PriceValue("122.345")
pv1.value
"122.345"
pv1.value == pv2.value
True
```

Regardless the value we instantiate it with, a float or a string, the PriceValue instance will allways be a string value.

The types also validate the values passed. Invalid values will raise an exception.

5.1 AccountID

```
class oandapyV20.types.AccountID(accountID) representation of an AccountID, string value of an Account Identifier.
```

Parameters accountID (string (required)) - the accountID of a v20 account

Example

```
>>> print AccountID("001-011-5838423-001").value
```

A ValueError exception is raised in case of an incorrect value.

```
__init__(accountID)
    x.__init__(...) initializes x; see help(type(x)) for signature
value
    value property.
```

5.2 AccountUnits

5.3 ClientComment

5.4 ClientID

5.5 ClientTag

5.6 DateTime

```
class oandapyV20.types.DateTime(dateTime)
    representation of a DateTime as a RFC 3339 string.
```

Parameters

- dateTime(string, datetime instance, dict (required))
 - the dateTime parameter must be:
 - a valid RFC3339 string representing a date-time, or
 - a dict holding the relevant datetime parts, or
 - a datetime.datetime instance
- value property is always RFC3339 datetime string (The) -
- \bullet seconds are in microseconds. This compatible with (\textit{Fractional})
- datetime.datetime. -

Example

A ValueError exception is raised in case of an invalid value

```
__init__(dateTime)
     x.__init__(...) initializes x; see help(type(x)) for signature
value
    value property.
```

5.7 OrderID

```
class oandapyV20.types.OrderID (orderID) representation of an orderID, string value of an integer.
```

Parameters orderID (integer or string (required)) – the orderID as a positive integer or as a string

Example

```
A ValueError exception is raised in case of a negative integer value

__init__(orderID)

x.__init__(...) initializes x; see help(type(x)) for signature
```

5.6. DateTime 105

```
value value property.
```

5.8 Orderldentifier

5.9 OrderSpecifier

```
class oandapyV20.types.OrderSpecifier (specifier)
    representation of the OrderSpecifier.
    __init__ (specifier)
        x.__init__(...) initializes x; see help(type(x)) for signature
    value
        value property.
```

5.10 PriceValue

5.11 TradeID

```
class oandapyV20.types.TradeID (tradeID)
    representation of a tradeID, string value of an integer.

Parameters tradeID (integer or string (required)) - the tradeID as a positive integer or as a string
```

Example

```
>>> print TradeID(1234).value
```

A ValueError exception is raised in case of a negative integer value

```
__init__(tradeID)
            x.__init__(...) initializes x; see help(type(x)) for signature
value
            value property.
```

5.12 Units

```
class oandapyV20.types.Units(units)
    representation Units, string value of an integer.
    __init__(units)
            x.__init__(...) initializes x; see help(type(x)) for signature
    value
            value property.
```

5.12. Units 107

oandapyV20.contrib

6.1 Factories

The oandapyV20.contrib.factories module contains several classes / methods that can be used optionally to generate requests.

6.1.1 InstrumentsCandlesFactory

oandapyV20.contrib.factories.InstrumentsCandlesFactory(instrument, params=None) InstrumentsCandlesFactory - generate InstrumentCandles requests.

InstrumentsCandlesFactory is used to retrieve historical data by automatically generating consecutive requests when the OANDA limit of *count* records is exceeded.

This is known by calculating the number of candles between *from* and *to*. If *to* is not specified *to* will be equal to *now*

The *count* parameter is only used to control the number of records to retrieve in a single request.

The *includeFirst* parameter is forced to make sure that results do no have a 1-record gap between consecutive requests.

Parameters

- instrument (string (required)) the instrument to create the order for
- params (params (optional)) the parameters to specify the historical range, see the REST-V20 docs regarding 'instrument' at developer.oanda.com If no params are specified, just a single InstrumentsCandles request will be generated acting the same as if you had just created it directly.

Example

The *oandapyV20.API* client processes requests as objects. So, downloading large historical batches simply comes down to:

```
>>> import json
>>> from oandapyV20 import API
>>> from oandapyV20.contrib.factories import InstrumentsCandlesFactory
>>> client = API(access_token=...)
>>> instrument, granularity = "EUR_USD", "M15"
>>> _from = "2017-01-01T00:00:00Z"
>>> params = {
      "from": _from,
       "granularity": granularity,
. . .
       "count": 2500,
. . .
. . . }
>>> with open("/tmp/{}.{}".format(instrument, granularity), "w") as OUT:
       # The factory returns a generator generating consecutive
>>>
        # requests to retrieve full history from date 'from' till 'to'
>>>
        for r in InstrumentsCandlesFactory(instrument=instrument,
>>>
                                            params=params)
. . .
            client.request(r)
>>>
            OUT.write(json.dumps(r.response.get('candles'), indent=2))
```

Note: Normally you can't combine *from*, *to* and *count*. When *count* specified, it is used to calculate the gap between *to* and *from*. The *params* passed to the generated request itself does contain the *count* parameter.

6.2 Generic

The oandapy V20. contrib. generic module contains several classes / methods that serve a generic purpose.

6.2.1 granularity_to_time

```
oandapyV20.contrib.generic.granularity_to_time(s) convert a named granularity into seconds.
```

get value in seconds for named granularities: M1, M5 ... H1 etc.

```
>>> print(granularity_to_time("M5"))
300
```

```
oandapyV20.contrib.generic.secs2time(e) secs2time - convert epoch to datetime.
```

```
>>> d = secs2time(1497499200)

>>> d

datetime.datetime(2017, 6, 15, 4, 0)

>>> d.strftime("%Y%m%d-%H:%M:%S")

'20170615-04:00:00'
```

6.3 Order Classes

The oandapy V20.contrib.requests module contains several classes that can be used optionally when creating Order Requests.

When creating an order to create a position, it is possible to create dependant orders that will be triggered when the position gets filled. This goes typically for *Take Profit* and *Stop Loss*.

These order specifications and additional data that goes with these order specifications can be created by the contrib.requests.*Order* classes and the contrib.requests.*Details classes.

6.3.1 LimitOrderRequest

```
class oandapyV20.contrib.requests.LimitOrderRequest (instrument, units, price, posi-
tionFill='DEFAULT', clientEx-
tensions=None, takeProfitOn-
Fill=None, timeInForce='GTC',
gtdTime=None, stopLossOn-
Fill=None, trailingStopLossOn-
Fill=None, tradeClientExten-
sions=None)
```

Bases: oandapyV20.contrib.requests.baserequest.BaseRequest create a LimitOrderRequest.

LimitOrderRequest is used to build the body for a LimitOrder. The body can be used to pass to the OrderCreate endpoint.

```
__init__ (instrument, units, price, positionFill='DEFAULT', clientExtensions=None, takeProfitOn-
Fill=None, timeInForce='GTC', gtdTime=None, stopLossOnFill=None, trailingStopLos-
sOnFill=None, tradeClientExtensions=None)
Instantiate a LimitOrderRequest.
```

Parameters

- instrument (string (required)) the instrument to create the order for
- units (integer (required)) the number of units. If positive the order results in a LONG order. If negative the order results in a SHORT order
- price (float (required)) the price indicating the limit.

Example

```
>>> import json
>>> from oandapyV20 import API
>>> import oandapyV20.endpoints.orders as orders
>>> from oandapyV20.contrib.requests import LimitOrderRequest
>>>
>>> accountID = "..."
>>> client = API(access_token=...)
>>> ordr = LimitOrderRequest(instrument="EUR_USD",
... units=10000, price=1.08)
>>> print(json.dumps(ordr.data, indent=4))
{
    "order": {
        "timeInForce": "GTC",
```

(continues on next page)

6.3. Order Classes 111

data

data property.

return the JSON order body

6.3.2 MarketOrderRequest

```
class oandapyV20.contrib.requests.MarketOrderRequest (instrument,
                                                                                          price-
                                                                    Bound=None,
                                                                                       position-
                                                                    Fill='DEFAULT',
                                                                                          clien-
                                                                    tExtensions=None,
                                                                                           take-
                                                                    ProfitOnFill=None,
                                                                                         timeIn-
                                                                    Force='FOK',
                                                                                    stopLossOn-
                                                                    Fill=None,
                                                                                 trailingStopLos-
                                                                    sOnFill=None, tradeClientEx-
                                                                    tensions=None)
```

 $\pmb{Bases:}\ \texttt{oandapyV20.contrib.requests.baserequest.BaseRequest}$

create a MarketOrderRequest.

MarketOrderRequest is used to build the body for a MarketOrder. The body can be used to pass to the Order-Create endpoint.

__init__ (instrument, units, priceBound=None, positionFill='DEFAULT', clientExtensions=None, takeProfitOnFill=None, timeInForce='FOK', stopLossOnFill=None, trailingStopLossOn-Fill=None, tradeClientExtensions=None)
Instantiate a MarketOrderRequest.

Parameters

- instrument (string (required)) the instrument to create the order for
- units (integer (required)) the number of units. If positive the order results in a LONG order. If negative the order results in a SHORT order

Example

```
>>> import json
>>> from oandapyV20 import API
>>> import oandapyV20.endpoints.orders as orders
>>> from oandapyV20.contrib.requests import MarketOrderRequest
>>>
>>> accountID = "..."
>>> client = API(access_token=...)
```

```
>>> mo = MarketOrderRequest(instrument="EUR_USD", units=10000)
>>> print(json.dumps(mo.data, indent=4))
    "order": {
        "type": "MARKET",
        "positionFill": "DEFAULT",
        "instrument": "EUR_USD",
        "timeInForce": "FOK",
        "units": "10000"
    }
>>> # now we have the order specification, create the order request
>>> r = orders.OrderCreate(accountID, data=mo.data)
>>> # perform the request
>>> rv = client.request(r)
>>> print(rv)
>>> print(json.dumps(rv, indent=4))
    "orderFillTransaction": {
        "reason": "MARKET_ORDER",
        "pl": "0.0000",
        "accountBalance": "97864.8813",
        "units": "10000",
        "instrument": "EUR_USD",
        "accountID": "101-004-1435156-001",
        "time": "2016-11-11T19:59:43.253587917Z",
        "type": "ORDER_FILL",
        "id": "2504",
        "financing": "0.0000",
        "tradeOpened": {
           "tradeID": "2504",
            "units": "10000"
        },
        "orderID": "2503",
        "userID": 1435156,
        "batchID": "2503",
        "price": "1.08463"
    "lastTransactionID": "2504",
    "relatedTransactionIDs": [
        "2503",
        "2504"
    "orderCreateTransaction": {
        "type": "MARKET_ORDER",
        "reason": "CLIENT_ORDER",
        "id": "2503",
        "timeInForce": "FOK",
        "units": "10000",
        "time": "2016-11-11T19:59:43.253587917Z",
        "positionFill": "DEFAULT",
        "accountID": "101-004-1435156-001",
        "instrument": "EUR_USD",
        "batchID": "2503",
        "userID": 1435156
    }
```

(continues on next page)

6.3. Order Classes

```
>>>
```

data

data property.

return the JSON body.

6.3.3 MITOrderRequest

```
class oandapyV20.contrib.requests.MITOrderRequest (instrument,
                                                                                units,
                                                                                           price,
                                                                 priceBound=None.
                                                                                        position-
                                                                 Fill='DEFAULT',
                                                                                          timeIn-
                                                                 Force='GTC',
                                                                                  gtdTime=None,
                                                                 clientExtensions=None,
                                                                                            take-
                                                                 ProfitOnFill=None,
                                                                                     stopLossOn-
                                                                              trailingStopLossOn-
                                                                 Fill=None,
                                                                 Fill=None,
                                                                                 tradeClientExten-
                                                                 sions=None)
```

 $Bases: \verb|oandapyV20.contrib.requests.baserequest.BaseRequest|\\$

create a MarketIfTouched OrderRequest.

MITOrderRequest is used to build the body for a MITOrder. The body can be used to pass to the OrderCreate endpoint.

__init__ (instrument, units, price, priceBound=None, positionFill='DEFAULT', timeInForce='GTC', gtdTime=None, clientExtensions=None, takeProfitOnFill=None, stopLossOnFill=None, trailingStopLossOnFill=None, tradeClientExtensions=None)
Instantiate an MITOrderRequest.

Parameters

- instrument (string (required)) the instrument to create the order for
- units (integer (required)) the number of units. If positive the order results in a LONG order. If negative the order results in a SHORT order
- price (float (required)) the price indicating the limit.

Example

```
>>> import json
>>> from oandapyV20 import API
>>> import oandapyV20.endpoints.orders as orders
>>> from oandapyV20.contrib.requests import MITOrderRequest
>>>
>>> accountID = "..."
>>> client = API(access_token=...)
>>> ordr = MITOrderRequest(instrument="EUR_USD",
... units=10000, price=1.08)
>>> print(json.dumps(ordr.data, indent=4))
{
    "order": {
        "timeInForce": "GTC",
        "instrument": "EUR_USD",
        "units": "10000",
```

```
"price": "1.08000",
    "type": "MARKET_IF_TOUCHED",
    "positionFill": "DEFAULT"
}

>>> r = orders.OrderCreate(accountID, data=ordr.data)
>>> rv = client.request(r)
>>> ...
```

data

data property.

return the JSON order body

6.3.4 PositionCloseRequest

create a PositionCloseRequest.

PositionCloseRequest is used to build the body to close a position. The body can be used to pass to the PositionClose endpoint.

```
__init__(longUnits=None, longClientExtensions=None, shortUnits=None, shortClientExtensions=None)
Instantiate a PositionCloseRequest.
```

Parameters

- longUnits (integer (optional)) the number of long units to close
- longClientExtensions (dict (optional)) dict representing longClientExtensions
- shortUnits (integer (optional)) the number of short units to close
- **shortClientExtensions** (dict (optional)) dict representing shortClientExtensions

One of the parameters or both must be supplied.

Example

```
>>> import json
>>> from oandapyV20 import API
>>> import oandapyV20.endpoints.positions as positions
>>> from oandapyV20.contrib.requests import PositionCloseRequest
>>>
>>> accountID = "..."
>>> client = API(access_token=...)
>>> ordr = PositionCloseRequest(longUnits=10000)
>>> print(json.dumps(ordr.data, indent=4))
{
```

(continues on next page)

6.3. Order Classes 115

6.3.5 StopLossOrderRequest

Bases: oandapyV20.contrib.requests.baserequest.BaseRequest create a StopLossOrderRequest.

StopLossOrderRequest is used to build the body for a StopLossOrder. The body can be used to pass to the OrderCreate endpoint.

```
__init__ (tradeID, price, clientTradeID=None, timeInForce='GTC', gtdTime=None, clientExten-
sions=None)
Instantiate a StopLossOrderRequest.
```

Parameters

- tradeID (string (required)) the tradeID of an existing trade
- price (float (required)) the treshold price indicating the price to close the
 order

Example

```
>>> import json
>>> from oandapyV20 import API
>>> import oandapyV20.endpoints.orders as orders
>>> from oandapyV20.contrib.requests import StopLossOrderRequest
>>>
>>> accountID = "..."
>>> client = API(access_token=...)
>>> ordr = StopLossOrderRequest(tradeID="1234", price=1.07)
>>> print(json.dumps(ordr.data, indent=4))
{
    "order": {
        "type": "STOP_LOSS",
        "tradeID": "1234",
        "price": "1.07000",
        "timeInForce": "GTC",
    }
}
>>> # now we have the order specification, create the order request
```

```
>>> r = orders.OrderCreate(accountID, data=ordr.data)
>>> # perform the request
>>> rv = client.request(r)
>>> print(json.dumps(rv, indent=4))
>>> ...
```

data

data property.

return the JSON body.

6.3.6 StopOrderRequest

Bases: oandapyV20.contrib.requests.baserequest.BaseRequest

create a StopOrderRequest.

StopOrderRequest is used to build the body for an StopOrder. The body can be used to pass to the OrderCreate endpoint.

__init__ (instrument, units, price, priceBound=None, positionFill='DEFAULT', timeInForce='GTC', gtdTime=None, clientExtensions=None, takeProfitOnFill=None, stopLossOnFill=None, trailingStopLossOnFill=None, tradeClientExtensions=None)
Instantiate a StopOrderRequest.

Parameters

- instrument (string (required)) the instrument to create the order for
- units (integer (required)) the number of units. If positive the order results in a LONG order. If negative the order results in a SHORT order
- price (float (required)) the treshold price indicating the price to activate the order

Example

```
>>> import json
>>> from oandapyV20 import API
>>> import oandapyV20.endpoints.orders as orders
>>> from oandapyV20.contrib.requests import StopOrderRequest
>>>
>>> accountID = "..."
>>> client = API(access_token=...)
>>> ordr = StopOrderRequest(instrument="EUR_USD",
... units=10000, price=1.07)
>>> print(json.dumps(ordr.data, indent=4))
```

(continues on next page)

6.3. Order Classes 117

```
"order": {
    "type": "STOP",
    "price": "1.07000",
    "positionFill": "DEFAULT",
    "instrument": "EUR_USD",
    "timeInForce": "GTC",
    "units": "10000"
}

>>> # now we have the order specification, create the order request
>>> r = orders.OrderCreate(accountID, data=ordr.data)
>>> # perform the request
>>> rv = client.request(r)
>>> print(json.dumps(rv, indent=4))
>>> ...
```

data

data property.

return the JSON body.

6.3.7 TakeProfitOrderRequest

Bases: oandapyV20.contrib.requests.baserequest.BaseRequest

create a TakeProfit OrderRequest.

TakeProfitOrderRequest is used to build the body for a TakeProfitOrder. The body can be used to pass to the OrderCreate endpoint.

```
__init__(tradeID, price, clientTradeID=None, timeInForce='GTC', gtdTime=None, clientExtensions=None)
Instantiate a TakeProfitOrderRequest.
```

Parameters

- tradeID (string (required)) the tradeID of an existing trade
- **price** (float (required)) the price indicating the target price to close the order.

Example

```
>>> import json
>>> from oandapyV20 import API
>>> import oandapyV20.endpoints.orders as orders
>>> from oandapyV20.contrib.requests import TakeProfitOrderRequest
>>>
>>> accountID = "..."
>>> client = API(access_token=...)
>>> ordr = TakeProfitOrderRequest(tradeID="1234",
```

```
price=1.22)
>>> print(json.dumps(ordr.data, indent=4))

{
    "order": {
        "timeInForce": "GTC",
        "price": "1.22000",
        "type": "TAKE_PROFIT",
        "tradeID": "1234"
    }
}
>>> r = orders.OrderCreate(accountID, data=ordr.data)
>>> rv = client.request(r)
>>> ...
```

data

data property.

return the JSON order body

6.3.8 TradeCloseRequest

```
class oandapyV20.contrib.requests.TradeCloseRequest (units='ALL')
    Bases: oandapyV20.contrib.requests.baserequest.BaseRequest
    create a TradeCloseRequest.
```

TradeCloseRequest is used to build the body to close a trade. The body can be used to pass to the TradeClose endpoint.

```
__init__ (units='ALL')
Instantiate a TradeCloseRequest.
```

Parameters units (integer (optional)) – the number of units to close. Default it is set to "ALL".

Example

```
>>> import json
>>> from oandapyV20 import API
>>> import oandapyV20.endpoints.trades as trades
>>> from oandapyV20.contrib.requests import TradeCloseRequest
>>>
>>> accountID = "..."
>>> client = API(access_token=...)
>>> ordr = TradeCloseRequest (units=10000)
>>> print(json.dumps(ordr.data, indent=4))
   "units": "10000"
>>> # now we have the order specification, create the order request
>>> r = trades.TradeClose(accountID, tradeID=1234,
                          data=ordr.data)
>>> # perform the request
>>> rv = client.request(r)
>>> print(rv)
```

6.3. Order Classes 119

6.3.9 TrailingStopLossOrderRequest

```
class oandapyV20.contrib.requests.TrailingStopLossOrderRequest (tradeID, dis-
tance, client-
TradeID=None,
timeIn-
Force='GTC',
gtdTime=None,
clientExten-
sions=None)
```

Bases: oandapyV20.contrib.requests.baserequest.BaseRequest create a TrailingStopLossOrderRequest.

TrailingStopLossOrderRequest is used to build the body for a TrailingStopLossOrder. The body can be used to pass to the OrderCreate endpoint.

```
__init__ (tradeID, distance, clientTradeID=None, timeInForce='GTC', gtdTime=None, clientExten-
sions=None)
Instantiate a TrailingStopLossOrderRequest.
```

Parameters

- tradeID (string (required)) the tradeID of an existing trade
- distance (float (required)) the price distance

Example

```
>>> import json
>>> from oandapyV20 import API
>>> import oandapyV20.endpoints.orders as orders
>>> from oandapyV20.contrib.requests import TrailingStopLossOrderRequest
>>> accountID = "..."
>>> client = API(access_token=...)
>>> ordr = TrailingStopLossOrderRequest(tradeID="1234", distance=20)
>>> print(json.dumps(ordr.data, indent=4))
    "order": {
        "type": "TRAILING_STOP_LOSS",
        "tradeID": "1234",
        "timeInForce": "GTC",
        "distance": "20.00000"
>>> # now we have the order specification, create the order request
>>> r = orders.OrderCreate(accountID, data=ordr.data)
>>> # perform the request
>>> rv = client.request(r)
>>> print(json.dumps(rv, indent=4))
>>> ...
```

data

data property.

return the JSON body.

6.4 support classes

The oandapy V20.contrib.requests module contains several classes that can be used optionally when creating Order Requests.

When creating an order to create a position, it is possible to create dependant orders that will be triggered when the position gets filled. This goes typically for *Take Profit* and *Stop Loss*.

These order specifications and additional data that goes with these order specifications can be created by the contrib.requests.*Order* classes and the contrib.requests.*Details classes.

6.4.1 Client Extensions

Client extensions can be used optionally on Order Requests. It allows a client to set a custom ID, Tag and/or Comment.

Parameters

Instantiate ClientExtensions.

- clientID (clientID (required)) the clientID
- clientTag (clientTag (required)) the clientTag
- clientComment (clientComment (required)) the clientComment

Example

```
>>> import json
>>> from oandapyV20 import API
>>> import oandapyV20.endpoints.orders as orders
>>> from oandapyV20.contrib.requests import (
       MarketOrderRequest, TakeProfitDetails, ClientExtensions)
>>>
>>> accountID = "..."
>>> client = API(access_token=...)
>>> # at time of writing EUR_USD = 1.0740
>>> # let us take profit at 1.10, GoodTillCancel (default)
>>> # add clientExtensions to it also
>>> takeProfitOnFillOrder = TakeProfitDetails(
       price=1.10,
       clientExtensions=ClientExtensions(clientTag="mytag").data)
>>> print(takeProfitOnFillOrder.data)
    'timeInForce': 'GTC',
    'price": '1.10000',
    'clientExtensions': {'tag': 'mytag'}
>>> ordr = MarketOrderRequest(
     instrument="EUR_USD",
```

```
units=10000,
takeProfitOnFill=takeProfitOnFillOrder.data

)

>>> # or as shortcut ...
>>> # takeProfitOnFill=TakeProfitDetails(price=1.10).data
>>> print(json.dumps(ordr.data, indent=4))
>>> r = orders.OrderCreate(accountID, data=ordr.data)
>>> rv = client.request(r)
>>> ...
```

6.4.2 StopLossDetails

```
class oandapy V20.contrib.requests. StopLossDetails (price, time In Force='GTC', gtdTime=None, client Extensions=None)
```

 $Bases: \verb|oandapyV20.contrib.requests.onfill.OnFill| \\$

Representation of the specification for a StopLossOrder.

It is typically used to specify 'stop loss details' for the 'stopLossOnFill' parameter of an OrderRequest. This way one can create the Stop Loss Order as a dependency when an order gets filled.

The other way to create a StopLossOrder is to create it afterwards on an existing trade. In that case you use StopLossOrderRequest on the trade.

```
__init__ (price, timeInForce='GTC', gtdTime=None, clientExtensions=None)
Instantiate StopLossDetails.
```

Parameters

- price (float or string (required)) the price to trigger take profit order
- timeInForce (TimeInForce (required), default TimeInForce.GTC)
 the time in force
- gtdTime (DateTime (optional)) gtdTime is required in case timeInForce == TimeInForce.GTD
- clientExtensions (ClientExtensions (optional)) -

Example

```
>>> import json
>>> from oandapyV20 import API
>>> import oandapyV20.endpoints.orders as orders
>>> from oandapyV20.contrib.requests import (
>>> MarketOrderRequest, StopLossDetails)
>>>
>>> accountID = "..."
>>> client = API(access_token=...)
>>> # at time of writing EUR_USD = 1.0740
>>> # let us take profit at 1.10, GoodTillCancel (default)
>>> stopLossOnFill = StopLossDetails(price=1.06)
>>> print(stopLossOnFill)
{
    "timeInForce": "GTC",
```

6.4.3 TakeProfitDetails

Bases: oandapyV20.contrib.requests.onfill.OnFill

Representation of the specification for a TakeProfitOrder.

It is typically used to specify 'take profit details' for the 'takeProfitOnFill' parameter of an OrderRequest. This way one can create the Take Profit Order as a dependency when an order gets filled.

The other way to create a TakeProfitOrder is to create it afterwards on an existing trade. In that case you use TakeProfitOrderRequest on the trade.

```
__init__ (price, timeInForce='GTC', gtdTime=None, clientExtensions=None)
Instantiate TakeProfitDetails.
```

Parameters

- price (float or string (required)) the price to trigger take profit order
- timeInForce (TimeInForce (required), default TimeInForce.GTC)
 the time in force
- gtdTime (DateTime (optional)) gtdTime is required in case timeInForce == TimeInForce.GTD

Example

```
>>> import json
>>> from oandapyV20 import API
>>> import oandapyV20.endpoints.orders as orders
>>> from oandapyV20.contrib.requests import (
>>> MarketOrderRequest, TakeProfitDetails)
>>>
>>> accountID = "..."
>>> client = API(access_token=...)
>>> # at time of writing EUR_USD = 1.0740
>>> # let us take profit at 1.10, GoodTillCancel (default)
>>> takeProfitOnFillOrder = TakeProfitDetails(price=1.10)
>>> print(takeProfitOnFillOrder.data)
```

```
"timeInForce": "GTC",
    "price": "1.10000"
>>> ordr = MarketOrderRequest(
       instrument="EUR_USD",
       units=10000,
       takeProfitOnFill=takeProfitOnFillOrder.data
>>> )
>>> # or as shortcut ...
>>> # takeProfitOnFill=TakeProfitDetails(price=1.10).data
>>> print(json.dumps(ordr.data, indent=4))
    "order": {
        "timeInForce": "FOK",
        "instrument": "EUR_USD",
        "units": "10000",
        "positionFill": "DEFAULT",
        "type": "MARKET",
        "takeProfitOnFill": {
            "timeInForce": "GTC",
            "price": "1.10000"
>>> r = orders.OrderCreate(accountID, data=ordr.data)
>>> rv = client.request(r)
>>> ...
```

6.4.4 TrailingStopLossDetails

```
class oandapyV20.contrib.requests.TrailingStopLossDetails (distance, timeIn-
Force='GTC', gtd-
Time=None, clientEx-
tensions=None)
```

 $Bases: \verb|oandapyV20.contrib.requests.onfill.OnFill| \\$

Representation of the specification for a TrailingStopLossOrder.

It is typically used to specify 'trailing stop loss details' for the 'trailingStopLossOnFill' parameter of an Order-Request. This way one can create the Trailing Stop Loss Order as a dependency when an order gets filled.

The other way to create a TrailingStopLossOrder is to create it afterwards on an existing trade. In that case you use TrailingStopLossOrderRequest on the trade.

```
__init__ (distance, timeInForce='GTC', gtdTime=None, clientExtensions=None)
Instantiate TrailingStopLossDetails.
```

Parameters

- distance (float or string (required)) the price to trigger trailing stop loss order
- timeInForce (TimeInForce (required), default TimeInForce.GTC)
 the time in force
- gtdTime (DateTime (optional)) gtdTime is required in case timeInForce == TimeInForce.GTD

• clientExtensions (ClientExtensions (optional)) -

Example

```
>>> import json
>>> from oandapyV20 import API
>>> import oandapyV20.endpoints.orders as orders
>>> from oandapyV20.contrib.requests import (
       MarketOrderRequest, TrailingStopLossDetails)
>>>
>>> accountID = "..."
>>> client = API(access_token=...)
>>> # at time of writing EUR_USD = 1.0740
>>> # add a trailing stoploss, at 50 pips GoodTillCancel (default)
>>> sld = 1.0740 - 1.0690
>>> trailingStopLossOnFill = TrailingStopLossDetails(distance=sld)
>>> print(trailingStopLossOnFill)
    "timeInForce": "GTC",
    "distance": "0.00500"
>>> ordr = MarketOrderRequest(
>>>
       instrument="EUR_USD",
>>>
        units=10000,
>>>
        \verb|trailingStopLossOnFill=trailingStopLossOnFill.data|\\
>>> )
>>> print(json.dumps(ordr.data, indent=4))
>>> r = orders.OrderCreate(accountID, data=ordr.data)
>>> rv = client.request(r)
>>> ...
```

125

CHAPTER 7

Examples

Examples can be found in the examples repositiony on github: examples reposition.

7.1 Example for trades-endpoints

Take the script below and name it 'trades.py'. From the shell:

```
hootnot@dev:~/test$ python trades.py list
hootnot@dev:~/test$ python trades.py open
hootnot@dev:~/test$ python trades.py details <id1> [<id2> ...]
hootnot@dev:~/test$ python trades.py close <id1> <numunits> [<id2> <numunits>...]
hootnot@dev:~/test$ python trades.py clext <id1> [<id2> ...]
hootnot@dev:~/test$ python trades.py crc_do <id1> <takeprofit> <stoploss> [<id2> ...]
```

```
if chc == 'open':
  r = trades.OpenTrades(accountID)
  rv = api.request(r)
  print("RESP:\n{} ".format(json.dumps(rv, indent=2)))
  tradeIDs = [o["id"] for o in rv["trades"]]
  print("TRADE IDS: {}".format(tradeIDs))
if chc == 'details':
  for 0 in sys.argv[2:]:
      r = trades.TradeDetails(accountID, tradeID=0)
      rv = api.request(r)
      print("RESP:\n{} ".format(json.dumps(rv, indent=2)))
if chc == 'close':
  X = iter(sys.argv[2:])
  for 0 in X:
      cfg = { "units": X.next() }
       r = trades.TradeClose(accountID, tradeID=0, data=cfg)
      rv = api.request(r)
      print("RESP:\n{} ".format(json.dumps(rv, indent=2)))
if chc == 'cltext':
   for 0 in sys.argv[2:]: # tradeIDs
       cfg = { "clientExtensions": {
               "id": "myID{}".format(0),
               "comment": "myComment",
         }
       r = trades.TradeClientExtensions(accountID, tradeID=0, data=cfg)
       rv = api.request(r)
      print("RESP:\n{} ".format(json.dumps(rv, indent=2)))
if chc == 'crc_do':
  X = iter(sys.argv[2:])
   for 0 in X:
      cfg = {
               "takeProfit": {
                 "timeInForce": "GTC",
                 "price": X.next(),
               "stopLoss": {
                 "timeInForce": "GTC",
                 "price": X.next()
       r = trades.TradeCRCDO(accountID, tradeID=0, data=cfg)
       rv = api.request(r)
      print("RESP:\n{} ".format(json.dumps(rv, indent=2)))
```

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Python Module Index

0

```
oandapyV20.contrib.factories,109
oandapyV20.contrib.generic,110
oandapyV20.definitions.accounts,79
oandapyV20.definitions.instruments,81
oandapyV20.definitions.orders,83
oandapyV20.definitions.pricing,87
oandapyV20.definitions.trades,88
oandapyV20.definitions.transactions,89
```

132 Python Module Index

Symbols	method), 86
getitem() (oan9 dapyV20.definitions.accounts.AccountFinancingMode method), 79	getitem() (oan- dapyV20.definitions.pricing.PriceStatus method), 87
getitem() (oang dapyV20.definitions.accounts.GuaranteedStopLossOre method), 80	getitem() (oan- derModapyV20.definitions.trades.TradePL method), 88
getitem() (oan9 dapyV20.definitions.accounts.PositionAggregationMod	de dapyV20.definitions.trades.TradeState method), 88 (oan-
getitem() (oan9 dapyV20.definitions.instruments.CandlestickGranular.method), 82	method), 89
getitem() (oan9 dapyV20.definitions.instruments.PriceComponents method), 83	getitem() (oan-dapyV20.definitions.transactions.FixedPriceOrderReason method), 92
dapyV20.definitions.instruments.WeeklyAlignment method), 82	getitem() (oan-dapyV20.definitions.transactions.FundingReason method), 91
dapyV20.definitions.orders.CancellableOrderType method), 85	getitem() (oan- dapyV20.definitions.transactions.LimitOrderReason method), 102
getitem() (oan9 dapyV20.definitions.orders.OrderPositionFill method), 85	<pre>getitem()</pre>
getitem() (oan9 dapyV20.definitions.orders.OrderState method), 86	getitem() (oan- dapyV20.definitions.transactions.MarketOrderMarginCloseous method), 90
getitem() (oan9 dapyV20.definitions.orders.OrderStateFilter method), 83	getitem() (oan- dapyV20.definitions.transactions.MarketOrderReason method), 93
getitem() (oan9 dapyV20.definitions.orders.OrderTriggerCondition method), 87	getitem() (oan- dapyV20.definitions.transactions.OrderCancelReason method), 101
getitem() (oan9 dapyV20.definitions.orders.OrderType method), 84	getitem() (oan- dapyV20.definitions.transactions.OrderFillReason method), 91
getitem() (oan9 dapyV20.definitions.orders.TimeInForce	getitem() (oan-dapyV20.definitions.transactions.StopLossOrderReason method), 90

getitem() (oan- method), 20
dapyV20.definitions.transactions.StopOrderReason_init() (oandapyV20.endpoints.forexlabs.Autochartist
method), 93 method), 21
getitem() (oaninit() (oandapyV20.endpoints.forexlabs.Calendar
dapyV20.definitions.transactions.TakeProfitOrderReason method), 23
method), 95init() (oandapyV20.endpoints.forexlabs.CommitmentsOfTraders
getitem() (oan- method), 23
$dapy V20. definitions. transactions. Trailing Stop Loss \underline{Oritler Re} \underline{aso} (n) \ (o and apy V20. endpoints. for extlabs. Historical Position Ratios \underline{Oritler Re} \underline{aso} (n) \ (o and apy V20. endpoints. for extlabs. Historical Position Ratios \underline{Oritler Re} \underline{Oritler Re} \underline{Oritler} \underline{Oritler Re} \underline{Oritler} Oritler$
method), 102 method), 25
getitem() (oaninit() (oandapyV20.endpoints.forexlabs.OrderbookData
$dapy V20. definitions. transactions. Transaction Reject Reason\ method),\ 32$
method), 100init() (oandapyV20.endpoints.forexlabs.Spreads
getitem() (oan- method), 34
$dapy V20. definitions. transactions. Transaction Type \verb \init\() (o and apy V20. endpoints. instruments. Instruments C and less than the context of the $
method), 95 method), 35
init() (oandapyV20.API method), 7
init() (oandapyV20.V20Error method), 8 method), 37
init() (oandapyV20.contrib.requests.ClientExtensionsinit() (oandapyV20.endpoints.instruments.InstrumentsPositionBoo
method), 121 method), 40
init() (oandapyV20.contrib.requests.LimitOrderRequestnit() (oandapyV20.endpoints.orders.OrderCancel
method), 111 method), 43
init() (oandapyV20.contrib.requests.MITOrderRequestnit() (oandapyV20.endpoints.orders.OrderClientExtensions
method), 114 method), 44
init() (oandapyV20.contrib.requests.MarketOrderRequestt() (oandapyV20.endpoints.orders.OrderCreate
method), 112 method), 45
init() (oandapyV20.contrib.requests.PositionClose <u>Req</u> uest() (oandapyV20.endpoints.orders.OrderDetails
method), 115 method), 46
init() (oandapyV20.contrib.requests.StopLossDeta <u>ils</u> init() (oandapyV20.endpoints.orders.OrderList
method), 122 method), 47
init() (oandapyV20.contrib.requests.StopLossOrde <u>rR</u> equest() (oandapyV20.endpoints.orders.OrderReplace
method), 116 method), 48 in it () (agadamyV20 contrib requests StapOrderPagaeita it () (agadamyV20 and points orders Orders Pending
init() (oandapyV20.contrib.requests.StopOrderRequestnit() (oandapyV20.endpoints.orders.OrdersPending method), 117
init() (oandapyV20.contrib.requests.TakeProfitDetailsinit() (oandapyV20.endpoints.positions.OpenPositions
method), 123 method), 51
init() (oandapyV20.contrib.requests.TakeProfitOrderRequests_() (oandapyV20.endpoints.positions.PositionClose
method), 118 method), 52
init() (oandapyV20.contrib.requests.TradeCloseRequests_it() (oandapyV20.endpoints.positions.PositionDetails
method), 119 method), 54
init() (oandapyV20.contrib.requests.TrailingStopLossDetails_() (oandapyV20.endpoints.positions.PositionList
method), 124 method), 55
init() (oandapyV20.contrib.requests.TrailingStopLossOnderRequestsondapyV20.endpoints.pricing.PricingInfo
method), 120 method), 57
init() (oandapyV20.endpoints.accounts.AccountChangesit() (oandapyV20.endpoints.pricing.PricingStream
method), 11 $method$), 60
init() (oandapyV20.endpoints.accounts.AccountConfiguration () (oandapyV20.endpoints.trades.OpenTrades
method), 13 method), 63
init() (oandapyV20.endpoints.accounts.AccountDetailsnit() (oandapyV20.endpoints.trades.TradeCRCDO
method), 14 method), 64
init() (oandapyV20.endpoints.accounts.AccountInstruments() (oandapyV20.endpoints.trades.TradeClientExtensions
method), 17 method), 66
init() (oandapyV20.endpoints.accounts.AccountList_init() (oandapyV20.endpoints.trades.TradeClose
method), 19 method), 67
init() (oandapyV20.endpoints.accounts.AccountSummiarryt() (oandapyV20.endpoints.trades.TradeDetails

method), 69	ACCOUNT_NEW_POSITIONS_LOCKED (oan-			
init() (oandapyV20.endpoints.trades.TradesList	dapy V20. definitions. transactions. Order Cancel Reason			
method), 70	attribute), 100			
init() (oandapyV20.endpoints.transactions.Transa				
method), 71	dapy V20. definitions. transactions. Transaction Reject Reason			
init() (oandapyV20.endpoints.transactions.Transa				
method), 72	ACCOUNT_ORDER_CANCEL_LOCKED (oan-			
method), 74	actionList dapyV20.definitions.transactions.TransactionRejectReason attribute), 96			
init() (oandapyV20.endpoints.transactions.Transactions				
method), 75	dapy V20. definitions. transactions. Order Cancel Reason			
init() (oandapyV20.endpoints.transactions.Transa				
method), 77	ACCOUNT_ORDER_CREATION_LOCKED (oan-			
init() (oandapyV20.types.AccountID method), 103	dapyV20.definitions.transactions.TransactionRejectReason attribute), 96			
	ACCOUNT_ORDER_FILL_LOCKED (oan-			
method), 104	dapy V20. definitions. transactions. Order Cancel Reason			
init() (oandapyV20.types.ClientComment	attribute), 100			
method), 104	ACCOUNT_POSITION_VALUE_LIMIT_EXCEEDED			
init() (oandapyV20.types.ClientID method), 104	(oandapyV20.definitions.transactions.OrderCancelReason attribute), 100			
	ACCOUNT_TRANSFER (oan-			
104	dapy V20. definitions. transactions. Funding Reason			
init() (oandapyV20.types.DateTime method),	attribute), 91			
105	ACCOUNT_WITHDRAWAL_LOCKED (oan-			
init() (oandapyV20.types.OrderID method), 105	dapyV20.definitions.transactions.TransactionRejectReason attribute), 96			
init() (oandapyV20.types.OrderIdentifier	AccountChanges (class in oan-			
method), 106	dapyV20.endpoints.accounts), 11			
init() (oandapyV20.types.OrderSpecifier method), 106	AccountConfiguration (class in oan-dapyV20.endpoints.accounts), 13			
init() (oandapyV20.types.PriceValue method),	AccountDetails (class in oan-			
106	dapyV20.endpoints.accounts), 14			
init() (oandapyV20.types.TradeID method), 106	AccountFinancingMode (class in oan-			
init() (oandapyV20.types.Units method), 107	dapy V20. definitions. accounts), 79			
۸	Account ID (class in oandapyV20.types), 103			
A	AccountInstruments (class in oan-			
${\tt A}\ (o and apy V20. definitions. instruments. Price Components$	dapyV20.endpoints.accounts), 17			
attribute), 83	AccountList (class in oan-			
ABSOLUTE_SUM (oan-	dapyV20.endpoints.accounts), 19			
dapyV20.definitions.accounts.PositionAggregation	on Mode unt Summary (class in oan-			
attribute), 80	dapyV20.endpoints.accounts), 20 AccountUnits (class in oandapyV20.types), 104			
ACCOUNT_CONFIGURATION_LOCKED (oan-	ectReasonMENT (oandapyV20.definitions.transactions.FundingReason			
attribute), 96	attribute), 91			
ACCOUNT_DEPOSIT_LOCKED (oan-	ADMIN_CONFIGURE_DATA_MISSING (oan-			
	ectReason dapyV20.definitions.transactions.TransactionRejectReason			
attribute), 96	attribute), 96 ALIAS_INVALID (oan-			
ACCOUNT_LOCKED (oan-				
dapyV20.definitions.transactions.OrderCancelRe	attribute), 96			
attribute), 100 ACCOUNT_LOCKED (oan-	ALL (oandapyV20.definitions.orders.OrderStateFilter at-			
dapyV20.definitions.transactions.TransactionRej				
attribute), 96	ALL (oandapyV20.definitions.trades.TradeStateFilter at-			

tribute), 89		C	TENT	_ORDER	(oan-
**	ns accounts Guaran			_ONDER d kapy&& .definitions.transactions.Mar	\
attribute), 80	rs.accounts.Guaran	ιεεασιορ		attribute), 92	kenjioucheuOraerKeason
	((0.00
AMOUNT_INVALID	,			_ORDER	(oan-
- ·	nsactions.1ransactio	мкејеси		dapyV20.definitions.transactions.Mar	keiOraerReason
attribute), 96	/	0.7		attribute), 93	
AMOUNT_MISSING	,	pan- Cl			(oan-
- · · · · · · · · · · · · · · · · · · ·	nsactions.1ransactio	опкејеси		dapyV20.definitions.transactions.Stop	LossOraerKeason
attribute), 96		0.7		attribute), 90	
API (class in oandapyV20), 5	0.1.77: 0			_ORDER	(oan-
ASK (oandapyV20.definitions.ord	ers.Order1riggerCo	ndition		dapyV20.definitions.transactions.Stop	OrderReason
attribute), 87				attribute), 93	
Autochartist (class		oan- Cl		_ORDER	(oan-
dapyV20.endpoints.fore	xlabs), 21			dapyV20.definitions.transactions.Take	eProfitOrderReason
D				attribute), 95	
В		C		_ORDER	(oan-
$\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	nents.PriceCompon	ents		dapy V20. definitions. transactions. Training a property of the property of	lingStopLossOrderReason
attribute), 83				attribute), 102	
${\tt BID}(o and apy V20. definitions. order$	ers.OrderTriggerCo	ndition	LIENT_	_ORDER_COMMENT_INVALID	(oan-
attribute), 87				dapy V20. definitions. transactions. Transactions. Transactions and the property of the prop	nsactionRejectReason
BOUNDS_VIOLATION		oan-		attribute), 96	
dapyV20.definitions.tra	nsactions.OrderCan	celReas	MENT_	_ORDER_ID_ALREADY_EXISTS	(oan-
attribute), 100				dapy V20. definitions. transactions. Transactions. Transactions and the property of the prop	nsactionRejectReason
<i>"</i>				attribute), 96	
C		C	LIENT_	_ORDER_ID_INVALID	(oan-
Calendar (class in oandapyV	20 andpoints foravla	abe)		dapyV20.definitions.transactions.Tran	nsactionRejectReason
22	20.enapoinis.jorexia	103),		attribute), 96	
CancellableOrderType	(class in c	oan- CI	LIENT_	_ORDER_TAG_INVALID	(oan-
dapyV20.definitions.ord	`	Jun-		dapyV20.definitions.transactions.Tran	nsactionRejectReason
CANCELLED (oandapyV20.defini		tata		attribute), 96	-
_ ·	nions.oraers.Oraers	ciaie C	LIENT_	_REQUEST	(oan-
attribute), 86 CANCELLED (oandapyV20.defini	tions orders Orders			dapyV20.definitions.transactions.Ord	erCancelReason
_ ·	nons.oraers.oraers	шегше	,	attribute), 100	
attribute), 83	(alass in	CI		_REQUEST_REPLACED	(oan-
CandlestickGranularity	`	oan-		dapyV20.definitions.transactions.Ord	erCancelReason
dapyV20.definitions.ins				attribute), 100	
CLIENT_CONFIGURE	((oan- 		_TRADE_COMMENT_INVALID	(oan-
adpy v 20. definitions. trai	nsactions.1ransactio	эптуре		dapyV20.definitions.transactions.Tran	nsactionRejectReason
attribute), 94	ATCCINC (attribute), 96	v
CLIENT_CONFIGURE_DATA_N	MISSING (6	oan- 			(oan-
dapyV20.definitions.trai	nsactions.1ransactio	мкејеси	Keason-	dapyV20.definitions.transactions.Ord	erCancelReason
attribute), 96	T. (attribute), 100	
CLIENT_CONFIGURE_REJECT	1 (d	oan- T Cl		_TRADE_ID_ALREADY_EXISTS	(oan-
aapyv 20.aejinitions.trai	nsactions.1ransactio	onType		dapyV20.definitions.transactions.Tran	`
attribute), 94				attribute), 96	,
CLIENT_EXTENSIONS_DATA_	_MISSING (d	oan-			(oan-
dapyV20.definitions.tra	nsactions.Transactio	onReječti	Rēāsòħ-	dapyV20.definitions.transactions.Tran	`
attribute), 96				attribute), 96	
CLIENT_FUNDING	(0	oan-			(oan-
dapyV20.definitions.tra	nsactions.FundingR	eason 🗀		_TRADE_TAG_INVALID dapyV20.definitions.transactions.Tran	,
attribute), 91				attribute), 96	isaciionixejeenxeason
CLIENT_ORDER	(6	oan-	lien+(Comment (class in pandamy V20 types	104
dapyV20.definitions.tra	nsactions.LimitOrde	erReasŏn	lion+I	Comment (class in oandapyV20.types Extensions (class in	oan-
attribute), 102		C.	LICITOI	dapyV20.contrib.requests), 121	oun-
				uupy , 20.commo.requesis), 121	

```
ClientID (class in oandapyV20.types), 104
                                                        data (oandapy V20.contrib.requests.MarketOrderRequest
ClientTag (class in oandapyV20.types), 104
                                                                  attribute), 114
                                                                (oandapyV20.contrib.requests.MITOrderRequest
CLOSE (oandapy V20.definitions.transactions.TransactionTypeta
         attribute), 94
                                                                  attribute), 115
CLOSE_TRADE_PARTIAL_UNITS_MISSING (oan- data(oandapyV20.contrib.requests.StopLossOrderRequest
         dapyV20.definitions.transactions.TransactionRejectReason attribute), 117
         attribute), 96
                                                        data (oandapyV20.contrib.requests.StopOrderRequest
CLOSE_TRADE_TYPE_MISSING
                                                                  attribute), 118
                                                 (oan-
         dapyV20.definitions.transactions.TransactionRejectReas(mandapyV20.contrib.requests.TakeProfitOrderRequest
         attribute), 97
                                                                  attribute), 119
CLOSE_TRADE_UNITS_EXCEED_TRADE_SIZE
                                                        \verb+data+ (o and a py V20. contrib. requests. Trailing Stop Loss Order Request
         (oandapyV20.definitions.transactions.TransactionRejectReasumibute), 120
                                                        DateTime (class in oandapyV20.types), 105
         attribute), 97
CLOSE_WHEN_TRADABLE
                                                        DEFAULT (oandapyV20.definitions.orders.OrderPositionFill
                                                 (oan-
         dapy V20. definitions. trades. Trade State\\
                                                                  attribute), 85
                                                   at-
         tribute), 88
                                                        DEFAULT (oandapyV20.definitions.orders.OrderTriggerCondition
CLOSE_WHEN_TRADEABLE
                                                                  attribute), 87
                                                 (oan-
         dapyV20.definitions.trades.TradeStateFilter
                                                        definitions (oandapy V20. definitions. accounts. Account Financing Mode
         attribute), 89
                                                                  attribute), 80
CLOSED (oandapyV20.definitions.trades.TradeState at-
                                                        definitions (oandapyV20.definitions.accounts.GuaranteedStopLossOr
         tribute), 88
                                                                  attribute), 80
CLOSED (oandapy V20. definitions. trades. TradeStateFilter definitions (oandapy V20. definitions. accounts. Position Aggregation Mo
         attribute), 89
                                                                  attribute), 80
CLOSEOUT POSITION DOESNT EXIST
                                                 (oan- definitions (oandapy V20. definitions. instruments. Candle stick Granular
         dapyV20.definitions.transactions.TransactionRejectReason attribute), 82
         attribute), 96
                                                        definitions (oandapy V20. definitions. instruments. Price Components
CLOSEOUT_POSITION_INCOMPLETE_SPECIFICATION
                                                                  attribute), 83
         (oandapyV20.definitions.transactions.TransactionRejEatReasionns (oandapyV20.definitions.instruments.WeeklyAlignment
                                                                 attribute), 82
CLOSEOUT_POSITION_PARTIAL_UNITS_MISSING definitions (oandapy V20. definitions. orders. Cancellable Order Type
         (oandapyV20.definitions.transactions.TransactionRejectReastmibute), 85
         attribute), 96
                                                        definitions (oandapy V20. definitions. orders. Order Position Fill
CLOSEOUT_POSITION_REJECT
                                                 (oan-
                                                                  attribute), 85
         dapyV20.definitions.transactions.TransactionRejectRéasontions (oandapyV20.definitions.orders.OrderState
         attribute), 96
                                                                  attribute), 86
CLOSEOUT_POSITION_UNITS_EXCEED_POSITION_& ££ initions (oandapy V20. definitions. orders. Order State Filter
         (oandapyV20.definitions.transactions.TransactionRejectReastmibute), 83
         attribute), 96
                                                        definitions (oandapy V20. definitions. orders. Order Trigger Condition
CommitmentsOfTraders
                                 (class
                                                                  attribute), 87
                                                 oan-
         dapyV20.endpoints.forexlabs), 23
                                                        definitions (oandapy V20. definitions. orders. Order Type
CREATE (oandapyV20.definitions.transactions.TransactionType
                                                                  attribute), 84
         attribute), 94
                                                        definitions (oandapy V20. definitions. orders. TimeInForce
                                                                  attribute), 86
D
                                                        definitions (oandapy V20. definitions. pricing. Price Status
                                                                  attribute), 87
D (oandapyV20.definitions.instruments.CandlestickGranularity
                                                        definitions (oandapyV20.definitions.trades.TradePL
         attribute), 81
                                                                 attribute), 88
DAILY (oandapy V20. definitions. accounts. Account Financing Mode
                                                        definitions (oandapy V20. definitions. trades. TradeState
         attribute), 79
                                                                  attribute), 89
DAILY_FINANCING
                                                 (oan-
         dapy V20. definitions. \textit{transactionS.TransactionType} \texttt{definitions}. \textit{(oandapy V20. definitions.trades.TradeStateFilter)} \\
                                                                  attribute), 89
         attribute), 94
                                                        definitions (oandapy V20. definitions. transactions. Fixed Price Order Rec
data (oandapyV20.contrib.requests.LimitOrderRequest
                                                                  attribute), 92
         attribute), 112
                                                        definitions (oandapy V20. definitions. transactions. Funding Reason
```

```
attribute), 91
                                                                                                                                                                                ENDPOINT (oandapyV20.endpoints.forexlabs.CommitmentsOfTraders
definitions (oandapy V20. definitions. transactions. Limit Order Reast tribute), 23
                                                                                                                                                                                ENDPOINT (oandapy V20. endpoints. for exlabs. Historical Position Ratios
                             attribute), 102
{\tt definitions.} \textit{(oandapy V20. definitions. transactions. Market If Touched Dibblete Re2 Son}
                             attribute), 92
                                                                                                                                                                                ENDPOINT (oandapyV20.endpoints.forexlabs.OrderbookData
definitions (oandapy V20. definitions. transactions. Market Order Matteibulk) se@utReason
                                                                                                                                                                                ENDPOINT
                                                                                                                                                                                                                             (oandapyV20.endpoints.forexlabs.Spreads
                             attribute), 90
definitions (oandapy V20. definitions. transactions. Market Order Reatts obtailed, 34
                             attribute), 93
                                                                                                                                                                                ENDPOINT (oandapyV20.endpoints.instruments.InstrumentsCandles
definitions (oandapy V20. definitions. transactions. Order Cancel Reasonbute), 35
                             attribute), 101
                                                                                                                                                                                ENDPOINT (oandapy V20. endpoints. instruments. Instruments Order Book
definitions (oandapy V20. definitions. transactions. Order Fill Reas attribute), 37
                                                                                                                                                                                {\tt ENDPOINT}\ (o and apy V20. endpoints. instruments. Instruments Position Book
                            attribute), 91
definitions (oandapy V20. definitions. transactions. Stop Loss Order Reciboute), 40
                             attribute), 90
                                                                                                                                                                                ENDPOINT (oandapyV20.endpoints.orders.OrderCancel
definitions (oandapy V20. definitions. transactions. Stop Order Reasonaribute), 43
                                                                                                                                                                                ENDPOINT (oandapy V20.endpoints.orders.OrderClientExtensions
                             attribute), 93
definitions (oandapy V20. definitions. transactions. Take Profit Ordear Reidustra), 44
                                                                                                                                                                                ENDPOINT (oandapyV20.endpoints.orders.OrderCreate
                             attribute), 95
definitions (oandapy V20. definitions. transactions. Trailing Stop Loss Wild telegration
                                                                                                                                                                                {\tt ENDPOINT}\ (o and apy V20. endpoints. orders. Order Details
                             attribute), 102
definitions (oandapy V20. definitions. transactions. Transaction Rejectiffleuts) p. 46
                             attribute), 100
                                                                                                                                                                                ENDPOINT (oandapyV20.endpoints.orders.OrderList at-
definitions (oandapy V20. definitions. transactions. Transaction Type ibute), 47
                             attribute), 95
                                                                                                                                                                                ENDPOINT (oandapyV20.endpoints.orders.OrderReplace
DELAYED TRADE CLOSE
                                                                                                                                                         (oan-
                                                                                                                                                                                                              attribute), 48
                             dapyV20.definitions.transactions.MarketOrderReasnnPOINT (oandapyV20.endpoints.orders.OrdersPending
                            attribute), 93
                                                                                                                                                                                                              attribute), 50
DELAYED_TRADE_CLOSURE
                                                                                                                                                         (oan- ENDPOINT (oandapy V20. endpoints. positions. Open Positions
                            dapy V20. definitions. transactions. Transaction Type
                                                                                                                                                                                                              attribute), 51
                            attribute), 94
                                                                                                                                                                                ENDPOINT (oandapyV20.endpoints.positions.PositionClose
DISABLED (oandapyV20.definitions.accounts.GuaranteedStopLossOndeniMade, 52
                                                                                                                                                                                ENDPOINT (oandapy V20. endpoints. positions. Position Details
                            attribute), 80
DIVISION_MIGRATION
                                                                                                                                                         (oan-
                                                                                                                                                                                                              attribute), 54
                             dapyV20.definitions.transactions.FundingReason ENDPOINT (oandapyV20.endpoints.positions.PositionList
                            attribute), 91
                                                                                                                                                                                                             attribute), 55
                                                                                                                                                                                ENDPOINT (oandapyV20.endpoints.pricing.PricingInfo
F
                                                                                                                                                                                                              attribute), 57
{\tt ENDPOINT}\ (o and apy V20. endpoints. accounts. Account Chan {\tt ENDPOINT}\ (o and apy V20. endpoints. pricing. Pricing Stream) {\tt ENDPOINT}\ (o and apy V20. endpoints. pricing. Pricing Stream) {\tt ENDPOINT}\ (o and apy V20. endpoints. pricing. Pricing Stream) {\tt ENDPOINT}\ (o and apy V20. endpoints. pricing. Pricing Stream) {\tt ENDPOINT}\ (o and apy V20. endpoints. pricing. Pricing Stream) {\tt ENDPOINT}\ (o and apy V20. endpoints. pricing. Pricing Stream) {\tt ENDPOINT}\ (o and apy V20. endpoints. pricing. Pricing Stream) {\tt ENDPOINT}\ (o and apy V20. endpoints. pricing. Pricing Stream) {\tt ENDPOINT}\ (o and apy V20. endpoints. pricing. Pricing Stream) {\tt ENDPOINT}\ (o and apy V20. endpoints. pricing. Pricing Stream) {\tt ENDPOINT}\ (o and apy V20. endpoints. pricing. Pricing Stream) {\tt ENDPOINT}\ (o and apy V20. endpoints. pricing. Pricing Stream) {\tt ENDPOINT}\ (o and apy V20. endpoints. pricing. Pricing Stream) {\tt ENDPOINT}\ (o and apy V20. endpoints. pricing. Pricing Stream) {\tt ENDPOINT}\ (o and apy V20. endpoints. pricing. Pricing Stream) {\tt ENDPOINT}\ (o and apy V20. endpoints. pricing. pric
                                                                                                                                                                                                              attribute), 60
                             attribute), 11
\verb|Endpoint| (o and a py V20. endpoints. accounts. Account Configuration INT (o and a py V20. endpoints. trades. Open Trades of the property 
                                                                                                                                                                                                              attribute), 63
                            attribute), 13
{\tt ENDPOINT}\ (o and apy V20. endpoints. accounts. Account Detail {\tt ENDPOINT}\ (o and apy V20. endpoints. trades. Trade Client {\tt Extensions}\ (o and apy V20. endpoints. trades. Trade {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trade {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trade {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trade {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trade {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trade {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trade {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trade {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trade {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trade {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trade {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trade {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trade {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trade {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trade {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trades {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trades {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trades {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trades {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trades {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trades {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trades {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trades {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trades {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trades {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trades {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trades {\tt Client Extensions}\ (o and apy V20. endpoints. trades. Trades {\tt Client Extensions}\ (o and apy V20. endpoints)\ (o and apy V20. endpoints. trades. Trades {\tt Client Extensions}\ (o
                                                                                                                                                                                                              attribute), 66
                             attribute), 14
                                                                                                                                                                                                                           (oandapyV20.endpoints.trades.TradeClose
{\tt ENDPOINT} \ (o and apy V20. endpoints. accounts. Account Instruments {\tt OINT}
                                                                                                                                                                                                              attribute), 67
                             attribute), 17
attribute), 64
                             attribute), 19
{\tt ENDPOINT}\ (o and apy V20. endpoints. accounts. Account Summer {\tt PPOINT}\ (o and apy V20. endpoints. trades. Trade Details {\tt PPOINT}\ (o and {\tt PPOINT}\ (o {\tt PPOIN
                                                                                                                                                                                                              attribute), 69
                            attribute), 20
                                                                                                                                                                                                                              (o and a py V20. endpoints. trades. Trades List\\
ENDPOINT (oandapyV20.endpoints.forexlabs.Autochartist ENDPOINT
                                                                                                                                                                                                              attribute), 70
                             attribute), 21
\verb|ENDPOINT| (o and apy V20. endpoints. for exlabs. Calendar | \verb|ENDPOINT| (o and apy V20. endpoints. transactions. Transaction Details)|
                                                                                                                                                                                                              attribute), 71
                            attribute), 22
```

${ t ENDPOINT}$ ($oandapy V20.endpoints.transactions.Transac$	ctionIDRang e lapyV20.endpoints.orders.OrderCancel
attribute), 72	attribute), 43
${ t ENDPOINT}$ (oandapy ${ t V20}. { t endpoints.transactions.Transactions}$	ction Later CTED_STATUS (oan-
attribute), 74	dapy V20. endpoints. orders. Order Client Extensions
${ t ENDPOINT}$ (oandapy ${ t V20}. { t endpoints.transactions.Transactions}$	ctionsSinceIDattribute), 44
attribute), 75	EXPECTED_STATUS (oan-
ENDPOINT (oandapyV20.endpoints.transactions.Transac attribute), 77	ctionsStreamdapyV20.endpoints.orders.OrderCreate at- tribute), 45
EXPECTED_STATUS (oan-	EXPECTED_STATUS (oan-
dapyV20.endpoints.accounts.AccountChanges	dapyV20.endpoints.orders.OrderDetails
attribute), 11	attribute), 46
EXPECTED_STATUS (oan-	EXPECTED_STATUS (oan-
dapyV20.endpoints.accounts.AccountConfigural attribute), 13	_
EXPECTED_STATUS (oan-	• •
dapyV20.endpoints.accounts.AccountDetails	dapyV20.endpoints.orders.OrderReplace
attribute), 14	attribute), 48
EXPECTED_STATUS (oan- dapyV20.endpoints.accounts.AccountInstrumen.	
* * · · · · · · · · · · · · · · · · · ·	**
attribute), 17	attribute), 50
EXPECTED_STATUS (oan-	_
dapyV20.endpoints.accounts.AccountList attribute), 19	dapyV20.endpoints.positions.OpenPositions attribute), 51
EXPECTED_STATUS (oan-	EXPECTED_STATUS (oan-
dapyV20.endpoints.accounts.AccountSummary attribute), 20	dapyV20.endpoints.positions.PositionClose attribute), 52
EXPECTED_STATUS (oan-	
dapyV20.endpoints.forexlabs.Autochartist	dapyV20.endpoints.positions.PositionDetails
attribute), 21	attribute), 54
EXPECTED_STATUS (oan-	
dapyV20.endpoints.forexlabs.Calendar at-	· · · · · · · · · · · · · · · · · · ·
tribute), 22	attribute), 55
EXPECTED_STATUS (oan-	_
dapyV20.endpoints.forexlabs.CommitmentsOfTr attribute), 23	raders dapyV20.endpoints.pricing.PricingInfo at- tribute), 57
EXPECTED_STATUS (oan-	EXPECTED_STATUS (oan-
dapyV20.endpoints.forexlabs.HistoricalPosition attribute), 25	nRatios dapyV20.endpoints.pricing.PricingStream attribute), 60
EXPECTED_STATUS (oan-	
dapyV20.endpoints.forexlabs.OrderbookData	dapyV20.endpoints.trades.OpenTrades at-
attribute), 32	tribute), 63
EXPECTED_STATUS (oan-	_
dapyV20.endpoints.forexlabs.Spreads at-	dapy V20. endpoints. trades. Trade Client Extensions
tribute), 34	attribute), 66
EXPECTED_STATUS (oan-	_
dapyV20.endpoints.instruments.InstrumentsCan attribute), 35	ndles dapyV20.endpoints.trades.TradeClose at- tribute), 67
<i>"</i>	EXPECTED_STATUS (oan-
dapyV20.endpoints.instruments.InstrumentsOrd	
attribute), 37	attribute), 64
EXPECTED_STATUS (oan-	_
dapyV20.endpoints.instruments.InstrumentsPost	
attribute), 40	tribute), 69
EXPECTED_STATUS (oan-	EXPECTED_STATUS (oan-

	dapyV20.endpoints.trades.TradesList	at-	Н
	tribute), 70		${\tt H1}\ (o and a py V20. definitions. instruments. Candle stick Granularity$
EXPE	CTED_STATUS	(oan-	attribute), 81
	dapyV20.endpoints.transactions.Transa	actionDet	ails 2 (oandapyV20.definitions.instruments.CandlestickGranularity
	attribute), /1		attribute), 81
EXPE	CTED_STATUS	(oan-	H2 (oandapyV20.definitions.instruments.CandlestickGranularity
	dapyV20.endpoints.transactions.Transa	actionIDR	Range attribute), 81
	attribute), 72		H3 (oandapyV20.definitions.instruments.CandlestickGranularity
EXPE	CTED_STATUS	(oan-	attribute), 81
	dapyV20.endpoints.transactions.Transa attribute), 74	actionList	H4 (oandapyV20.definitions.instruments.CandlestickGranularity attribute), 81
EXPE	CTED_STATUS	(oan-	H6 (oandapyV20.definitions.instruments.CandlestickGranularity
	dapyV20.endpoints.transactions.Transa	actionsSin	iceID attribute), 81
	attribute), 75		H8 (oandapyV20.definitions.instruments.CandlestickGranularity
EXPE	CTED_STATUS	(oan-	attribute) 81
	dapyV20.endpoints.transactions.Transa	actionsStr	requeance, 61
	attribute), 77		attribute), 13
_			HEADERS (oandapyV20.endpoints.orders.OrderClientExtensions
F			attribute), 44
FIFC	_VIOLATION	(oan-	HEADERS (oandapyV20.endpoints.orders.OrderCreate
	- dapyV20.definitions.transactions.Order	`	
	attribute), 100		HEADERS (oandapyV20.endpoints.orders.OrderReplace
FILL	ED (oandapyV20.definitions.orders.OrderS	State at-	attribute), 48
	tribute), 86		HEADERS (oandapyV20.endpoints.positions.PositionClose
FILI	ED (oandapyV20.definitions.orders.OrderS	tateFilter	
	attribute), 83		HEADERS (oandapyV20.endpoints.trades.TradeClientExtensions
FIXE	D_PRICE (oandapyV20.definitions.orders.	OrderTyp	
	attribute), 84	J _F	HEADERS (oandapyV20.endpoints.trades.TradeClose at-
FIXE	D_PRICE_ORDER	(oan-	tribute), 67
		`	Peheaders (oandapyV20.endpoints.trades.TradeCRCDO
	attribute), 94	JI	attribute), 64
Fixe	edPriceOrderReason (class in	oan-	HEDGING_GSLO_VIOLATION (oan-
	dapyV20.definitions.transactions), 92		dapyV20.definitions.transactions.OrderCancelReason
FOK	(oandapyV20.definitions.orders.TimeInFo	rce at-	attribute), 100
	tribute), 85		HistoricalPositionRatios (class in oan-
Fric	lay (oandapyV20.definitions.instruments.Wa attribute), 82	eeklyAligi	
FUND	ING_REASON_MISSING	(oan-	
	dapyV20.definitions.transactions.Trans		iegt Regro ment Missing (oan-
	attribute), 97	v	dapyV20.definitions.transactions.TransactionRejectReasor
Func	lingReason (class in	oan-	attribute), 97
	dapyV20.definitions.transactions), 91		INSTRUMENT_NOT_TRADEABLE (oan-
			dapyV20.definitions.transactions.TransactionRejectReasor
G			attribute), 97
GFD	(oandapyV20.definitions.orders.TimeInFo	rce at-	INSTRUMENT_PRICE_UNKNOWN (oan-
01.5	tribute), 85		dapyV20.definitions.transactions.TransactionRejectReasor
arar	ularity_to_time() (in module	oan-	attribute), 97
gran.	dapyV20.contrib.generic), 110	oun	INSTRUMENT_UNKNOWN (oan-
GTC	(oandapyV20.definitions.orders.TimeInFo	rce at-	dapyV20.definitions.transactions.TransactionRejectReasor
510	tribute), 85	ui	attribute), 97
GTD	(oandapyV20.definitions.orders.TimeInFo	rce at-	InstrumentsCandles (class in oan-
J 1 D	tribute), 86		dapyV20.endpoints.instruments), 35
Guar	ranteedStopLossOrderMode (class	in oan-	InstrumentsCandlesFactory() (in module oan-
2441	dapyV20.definitions.accounts), 80		dapyV20.contrib.factories), 109

```
M
InstrumentsOrderBook
                                                                                                       (class
                                                                                                                                       in
                                                                                                                                                            oan-
                             dapyV20.endpoints.instruments), 37
                                                                                                                                                                                 M (oandapyV20.definitions.instruments.CandlestickGranularity
InstrumentsPositionBook
                                                                                                                (class
                                                                                                                                                            oan-
                                                                                                                                                                                                                attribute), 81
                             dapyV20.endpoints.instruments), 40
                                                                                                                                                                                 M (oandapyV20.definitions.instruments.PriceComponents
INSUFFICIENT FUNDS
                                                                                                                                                          (oan-
                                                                                                                                                                                                               attribute), 83
                             \frac{-}{dapy V20. definitions. transactions. Transaction Reject Reason lapsy V20. definitions. instruments. Candle stick Granularity and the state of the state of
                             attribute), 97
                                                                                                                                                                                                                attribute), 81
INSUFFICIENT LIQUIDITY
                                                                                                                                                          (oan-
                                                                                                                                                                                 M10 (oandapyV20.definitions.instruments.CandlestickGranularity
                             dapyV20.definitions.transactions.OrderCancelReason
                                                                                                                                                                                                               attribute), 81
                             attribute), 100
                                                                                                                                                                                 M15 (oandapyV20.definitions.instruments.CandlestickGranularity
INSUFFICIENT_MARGIN
                                                                                                                                                          (oan-
                                                                                                                                                                                                                attribute), 81
                             dapy V20. definitions. transactions. Order Cancel Reason (o and apy V20. definitions. instruments. Candle stick Granularity) and the state of the contraction of th
                             attribute), 101
                                                                                                                                                                                                               attribute), 81
INSUFFICIENT_MARGIN
                                                                                                                                                          (oan-
                                                                                                                                                                                 {\tt M30}\ (o and apy V20. definitions. instruments. Candle stick Granularity
                             dapyV20.definitions.transactions.TransactionRejectReason attribute), 81
                             attribute), 97
                                                                                                                                                                                 M4 (oandapyV20.definitions.instruments.CandlestickGranularity
INTERNAL_SERVER_ERROR
                                                                                                                                                          (oan-
                                                                                                                                                                                                                attribute), 81
                             dapy V20. definitions. transactions. Order Cancel Reason (o and apy V20. definitions. instruments. Candle stick Granularity) and the state of the contraction of th
                             attribute), 101
                                                                                                                                                                                                               attribute), 81
INTERNAL SERVER ERROR
                                                                                                                                                          (oan-
                                                                                                                                                                                 MARGIN_CALL_ENTER
                                                                                                                                                                                                                                                                                                                                            (oan-
                             \frac{-}{dapy} V20. definitions. transactions. Transaction Reject Reason \ dapy V20. definitions. transactions. Transaction Type \ dapy V20. definitions. Transaction Type \ dapy V20. definition Type \ dapy V20. def
                                                                                                                                                                                                                attribute), 94
invalid (oandapyV20.definitions.pricing.PriceStatus
                                                                                                                                                                                 MARGIN_CALL_EXIT
                                                                                                                                                                                                                                                                                                                                            (oan-
                             attribute), 87
                                                                                                                                                                                                               dapyV20.definitions.transactions.TransactionType
INVALID REISSUE IMMEDIATE PARTIAL FILL
                                                                                                                                                                                                                attribute), 94
                              (oandapyV20.definitions.transactions.TransactionRejectReaseALL EXTEND
                                                                                                                                                                                                                                                                                                                                            (oan-
                             attribute), 97
                                                                                                                                                                                                                dapyV20.definitions.transactions.TransactionType
INVERSE (oandapyV20.definitions.orders.OrderTriggerCondition
                                                                                                                                                                                                                attribute), 94
                             attribute), 87
                                                                                                                                                                                  MARGIN_CHECK_VIOLATION
                     (oandapyV20.definitions.orders.TimeInForce at-
                                                                                                                                                                                                                dapyV20.definitions.transactions.MarketOrderMarginCloseoutRe
                              tribute), 86
                                                                                                                                                                                                               attribute), 90
                                                                                                                                                                                 MARGIN CLOSEOUT
                                                                                                                                                                                                                                                                                                                                            (oan-
                                                                                                                                                                                                                dapyV20.definitions.transactions.MarketOrderReason
LIMIT (oandapyV20.definitions.orders.CancellableOrderType
                                                                                                                                                                                                                attribute), 93
                             attribute), 84
                                                                                                                                                                                  MARGIN_RATE_INVALID
                                                                                                                                                                                                                                                                                                                                            (oan-
LIMIT
                           (oandapyV20.definitions.orders.OrderType at-
                                                                                                                                                                                                               dapy V20. definitions. transactions. Transaction Reject Reason\\
                             tribute), 84
                                                                                                                                                                                                                attribute), 97
LIMIT_ORDER (oandapy V20.definitions.transactions.Order Fit Reason Ate_would_trigger_closeout (oan-
                             attribute), 91
                                                                                                                                                                                                               dapyV20.definitions.transactions.TransactionRejectReason
LIMIT_ORDER (oandapy V20. definitions. transactions. Transaction Typetribute), 97
                             attribute), 94
                                                                                                                                                                                 MARGIN RATE WOULD TRIGGER MARGIN CALL
LIMIT_ORDER_REJECT
                                                                                                                                                          (oan-
                                                                                                                                                                                                                (oandapyV20.definitions.transactions.TransactionRejectReason
                             dapyV20.definitions.transactions.TransactionType
                                                                                                                                                                                                               attribute), 97
                             attribute), 94
                                                                                                                                                                                 MARKET (oandapyV20.definitions.orders.OrderType at-
LimitOrderReason
                                                                                            (class
                                                                                                                                  in
                                                                                                                                                            oan-
                                                                                                                                                                                                               tribute), 84
                             dapyV20.definitions.transactions), 102
                                                                                                                                                                                 MARKET_HALTED
                                                                                                                                                                                                                                                                                                                                            (oan-
LimitOrderRequest
                                                                                               (class
                                                                                                                                                            oan-
                                                                                                                                                                                                               dapy V20. definitions. transactions. Order Cancel Reason\\
                              dapyV20.contrib.requests), 111
                                                                                                                                                                                                                attribute), 101
LINKED TRADE CLOSED
                                                                                                                                                          (oan-
                                                                                                                                                                                 MARKET_IF_TOUCHED
                                                                                                                                                                                                                                                                                                                                            (oan-
                             dapyV20.definitions.transactions.OrderCancelReason
                                                                                                                                                                                                               dapyV20.definitions.orders.CancellableOrderType
                             attribute), 101
                                                                                                                                                                                                                attribute), 84
LOSING TAKE PROFIT
                                                                                                                                                          (oan- MARKET_IF_TOUCHED
                                                                                                                                                                                                                                                                                                                                            (oan-
                              dapyV20.definitions.transactions.OrderCancelReason
                                                                                                                                                                                                                dapyV20.definitions.orders.OrderType
                                                                                                                                                                                                                                                                                                                                                    at-
                             attribute), 101
                                                                                                                                                                                                                tribute), 84
```

```
MARKET_IF_TOUCHED_ORDER
                                               (oan- METHOD (oandapy V20. endpoints. accounts. Account Summary
         dapyV20.definitions.transactions.OrderFillReason
                                                                attribute), 20
        attribute), 91
                                                      METHOD (oandapyV20.endpoints.forexlabs.Autochartist
MARKET_IF_TOUCHED_ORDER
                                                               attribute), 21
                                               (oan-
         dapyV20.definitions.transactions.TransactionTypeMETHOD (oandapyV20.endpoints.forexlabs.Calendar at-
        attribute), 94
                                                               tribute), 23
MARKET IF TOUCHED ORDER REJECT
                                               (oan- METHOD (oandapyV20.endpoints.forexlabs.CommitmentsOfTraders
         dapyV20.definitions.transactions.TransactionType
                                                                attribute), 23
        attribute), 94
                                                      METHOD (oandapyV20.endpoints.forexlabs.HistoricalPositionRatios
                                                                attribute), 25
MARKET_ORDER
                                               (oan-
         dapyV20.definitions.transactions.OrderFillReasonMETHOD (oandapyV20.endpoints.forexlabs.OrderbookData
        attribute), 91
                                                                attribute), 32
MARKET ORDER
                                               (oan-
                                                      METHOD
                                                               (oandapyV20.endpoints.forexlabs.Spreads at-
        dapyV20.definitions.transactions.TransactionType
                                                                tribute), 34
        attribute), 94
                                                      METHOD (oandapyV20.endpoints.instruments.InstrumentsCandles
MARKET_ORDER_DELAYED_TRADE_CLOSE
                                               (oan-
                                                                attribute), 35
         dapyV20.definitions.transactions.OrderFillReasonMETHOD (oandapyV20.endpoints.instruments.InstrumentsOrderBook
        attribute), 91
                                                               attribute), 37
MARKET ORDER MARGIN CLOSEOUT
                                               (oan- METHOD (oandapyV20.endpoints.instruments.InstrumentsPositionBook
         dapyV20.definitions.transactions.OrderFillReason
                                                                attribute), 40
        attribute), 91
                                                      METHOD (oandapyV20.endpoints.orders.OrderCancel at-
MARKET ORDER POSITION CLOSEOUT
                                               (oan-
                                                               tribute), 43
         dapyV20.definitions.transactions.OrderFillReasonMETHOD (oandapyV20.endpoints.orders.OrderClientExtensions
        attribute), 91
                                                                attribute), 44
MARKET ORDER REJECT
                                               (oan- METHOD (oandapyV20.endpoints.orders.OrderCreate at-
        dapyV20.definitions.transactions.TransactionType
                                                               tribute), 45
        attribute), 94
                                                      METHOD (oandapyV20.endpoints.orders.OrderDetails at-
MARKET_ORDER_TRADE_CLOSE
                                               (oan-
                                                               tribute), 46
        dapyV20.definitions.transactions.OrderFillReasonMETHOD
                                                                (oandapyV20.endpoints.orders.OrderList at-
        attribute), 91
                                                                tribute), 47
MarketIfTouchedOrderReason (class in oan-
                                                      METHOD
                                                                 (oandapyV20.endpoints.orders.OrderReplace
         dapyV20.definitions.transactions), 92
                                                                attribute), 48
                                                      METHOD (oandapyV20.endpoints.orders.OrdersPending
MarketOrderMarginCloseoutReason (class in
         oandapyV20.definitions.transactions), 89
                                                                attribute), 50
MarketOrderReason
                             (class
                                        in
                                                      METHOD (oandapyV20.endpoints.positions.OpenPositions
                                                oan-
                                                                attribute), 51
        dapyV20.definitions.transactions), 92
MarketOrderRequest
                              (class
                                                oan-
                                                      METHOD (oandapyV20.endpoints.positions.PositionClose
         dapyV20.contrib.requests), 112
                                                                attribute), 52
MARKUP GROUP ID INVALID
                                               (oan-
                                                      METHOD (oandapyV20.endpoints.positions.PositionDetails
         dapyV20.definitions.transactions.TransactionRejectReason attribute), 54
        attribute), 97
                                                      METHOD
                                                                (oandapyV20.endpoints.positions.PositionList
MAXIMAL_SIDE
                                                                attribute), 55
                                               (oan-
         dapyV20.definitions.accounts.PositionAggregationMixibiOD (oandapyV20.endpoints.pricing.PricingInfo at-
                                                                tribute), 57
        attribute), 80
METHOD (oandapy V20.endpoints.accounts.AccountChangesMETHOD (oandapy V20.endpoints.pricing.PricingStream
         attribute), 11
                                                                attribute), 60
METHOD (oandapy V20.endpoints.accounts.AccountConfiguiMHTTHOD (oandapy V20.endpoints.trades.OpenTrades at-
         attribute), 13
                                                                tribute), 63
METHOD (oandapyV20.endpoints.accounts.AccountDetails METHOD (oandapyV20.endpoints.trades.TradeClientExtensions
         attribute), 14
                                                                attribute), 66
METHOD (oandapyV20.endpoints.accounts.AccountInstrumeMESTHOD (oandapyV20.endpoints.trades.TradeClose at-
        attribute), 17
                                                                tribute), 67
METHOD
          (oandapyV20.endpoints.accounts.AccountList METHOD
                                                                 (oandapyV20.endpoints.trades.TradeCRCDO
         attribute), 19
                                                                attribute), 64
```

METHOD (oandapyV20.endpoints.trades.TradeDetails attribute), 69	OPEN	(oandapyV20.definitions.trades.TradeState attribute), 88
METHOD (oandapyV20.endpoints.trades.TradesList attribute), 70	OPEN	(oandapyV20.definitions.trades.TradeStateFilter attribute), 89
METHOD (oandapyV20.endpoints.transactions.Transactio attribute), 71	onD@RAIS_C	
METHOD (oandapyV20.endpoints.transactions.Transactio attribute), 72	onIDRang <u>e</u> I	CRADES_ALLOWED_EXCEEDED (oan-dapyV20.definitions.transactions.OrderCancelReason
$\verb"METHOD" (o and a py V20. endpoints. transactions. Transaction for the property of the prop$		attribute), 101
attribute), 74	_	ositions (class in oan-
METHOD (oandapy V20.endpoints.transactions.Transactio		
attribute), 75 METHOD (oandapy V20.endpoints.transactions.Transactio		rades (class in oandapyV20.endpoints.trades), 63
attribute), 77		CANCEL (oan-
MID (oandapyV20.definitions.orders.OrderTriggerCondit attribute), 87		dapyV20.definitions.transactions.TransactionType attribute), 94
MIGRATION (oandapyV20.definitions.transactions.Order	rC@raetRe <u>a</u>	
attribute), 101 MITOrderRequest (class in oan-		dapyV20.definitions.transactions.TransactionType attribute), 94
dapyV20.contrib.requests), 114	ORDER_	_CLIENT_EXTENSIONS_MODIFY (oan-
Monday (oandapy V20. definitions. instruments. Weekly Aligatribute), 82	-	dapyV20.definitions.transactions.TransactionType attribute), 94
N	ORDER_	_CLIENT_EXTENSIONS_MODIFY_REJECT
		(oandapyV20.definitions.transactions.TransactionType
NEGATIVE (oandapyV20.definitions.trades.TradePL at-		attribute), 94
tribute), 88		DOESNT_EXIST (oan-
attribute), 80		odapyV20.definitions.transactions.TransactionRejectReaso attribute), 97
NO_FINANCING (oan-		_FILL (oandapyV20.definitions.transactions.TransactionTy
dapyV20.definitions.accounts.AccountFinancin		attribute), 94 _FILL_POSITION_ACTION_INVALID (oan-
attribute), 79 non_tradeable (oan-		dapyV20.definitions.transactions.TransactionRejectReaso
non_tradeable (oan- dapyV20.definitions.pricing.PriceStatus at-		attribute), 97
tribute), 87		FILL_POSITION_ACTION_MISSING (oan-
0		dapyV20.definitions.transactions.TransactionRejectReaso attribute), 97
oandapyV20.contrib.factories (module), 109		_ID_UNSPECIFIED (oan-
oandapyV20.contrib.generic (module), 110		dapyV20.definitions.transactions.TransactionRejectReaso
oandapyV20.definitions.accounts (module),	,	attribute), 97
79	ORDER_	_IDENTIFIER_INCONSISTENCY (oan-
oandapy V20. definitions.instruments ($mod-ule$), 81		dapyV20.definitions.transactions.TransactionRejectReaso attribute), 97
$\verb oandapyV20.definitions.orders (\textit{module}), 83$	ORDER_	_PARTIAL_FILL_OPTION_INVALID (oan-
oandapyV20.definitions.pricing (module), 87	,	dapyV20.definitions.transactions.TransactionRejectReasonattribute), 97
oandapyV20.definitions.trades (module), 88	ORDER_	PARTIAL_FILL_OPTION_MISSING (oan- dapyV20.definitions.transactions.TransactionRejectReaso
oandapyV20.definitions.transactions		attribute), 97
(module), 89	_oOrderh	
ON_FILL (oandapyV20.definitions.transactions.StopLos. attribute), 90		uupy v 20.e uupoinis.jorexiuosj, 52
ON_FILL (oandapyV20.definitions.transactions.TakePro attribute), 95	fitOrderRed	Cancel (class in oandapyV20.endpoints.orders),
ON_FILL (oandapyV20.definitions.transactions.Trailing.	StopEdsstr	Sangel Reason (class in oan- dapyV20.definitions.transactions), 100

dapyV20.endpoints.orders), 44	POSITION_SIZE_EXCEEDED (oan-dapyV20.definitions.transactions.OrderCancelReason
OrderCreate (class in oandapyV20.endpoints.orders), 45	attribute), 101 PositionAggregationMode (class in oan-
OrderDetails (class in oan-	dapyV20.definitions.accounts), 80
dapyV20.endpoints.orders), 46	PositionClose (class in oan-
OrderFillReason (class in oan-	dapyV20.endpoints.positions), 52
dapyV20.definitions.transactions), 90	PositionCloseRequest (class in oan-
OrderID (class in oandapyV20.types), 105	dapyV20.contrib.requests), 115
OrderIdentifier (class in oandapyV20.types), 106	PositionDetails (class in oan-
OrderList (class in oandapyV20.endpoints.orders), 47	dapyV20.endpoints.positions), 54
OrderPositionFill (class in oan-dapyV20.definitions.orders), 85	PositionList (class in oan-dapyV20.endpoints.positions), 55
OrderReplace (class in oan-	POSITIVE (oandapyV20.definitions.trades.TradePL at-
dapyV20.endpoints.orders), 48	tribute), 88
ORDERS_ON_FILL_DUPLICATE_CLIENT_ORDER_I	
(oandapyV20.definitions.transactions.Transactio attribute), 97	nRejectRea dap yV20.definitions.transactions.TransactionRejectReason attribute), 97
OrderSpecifier (class in oandapyV20.types), 106	PRICE_BOUND_PRECISION_EXCEEDED (oan-
OrdersPending (class in oan-dapyV20.endpoints.orders), 50	$dapy V20. definitions. transactions. Transaction Reject Reason \\ attribute), 97$
OrderState (class in oandapyV20.definitions.orders),	PRICE_DISTANCE_INVALID (oan-
86 OrderStateFilter (class in oan-	dapyV20.definitions.transactions.TransactionRejectReason attribute), 97
dapyV20.definitions.orders), 83	PRICE_DISTANCE_MAXIMUM_EXCEEDED (oan-
OrderTriggerCondition (class in oan-dapyV20.definitions.orders), 86	dapyV20.definitions.transactions.TransactionRejectReason attribute), 97
OrderType (class in oandapyV20.definitions.orders),	
83	dapyV20.definitions.transactions.TransactionRejectReason attribute), 97
P 83	
	attribute), 97
PENDING (oandapyV20.definitions.orders.OrderState attribute), 86 PENDING (oandapyV20.definitions.orders.OrderStateFilte	attribute), 97 PRICE_DISTANCE_MISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 rPRICE_DISTANCE_PRECISION_EXCEEDED (oan-
PENDING (oandapyV20.definitions.orders.OrderState attribute), 86 PENDING (oandapyV20.definitions.orders.OrderStateFilterattribute), 83	attribute), 97 PRICE_DISTANCE_MISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 rPRICE_DISTANCE_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason
PENDING (oandapyV20.definitions.orders.OrderState attribute), 86 PENDING (oandapyV20.definitions.orders.OrderStateFilte attribute), 83 PENDING_ORDERS_ALLOWED_EXCEEDED (oan-	attribute), 97 PRICE_DISTANCE_MISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 rPRICE_DISTANCE_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97
PENDING (oandapyV20.definitions.orders.OrderState attribute), 86 PENDING (oandapyV20.definitions.orders.OrderStateFilte attribute), 83 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.OrderCancelRef	attribute), 97 PRICE_DISTANCE_MISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 rPRICE_DISTANCE_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 rabbace_INVALID (oan-
PENDING (oandapyV20.definitions.orders.OrderState attribute), 86 PENDING (oandapyV20.definitions.orders.OrderStateFilte attribute), 83 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.OrderCancelReattribute), 101	attribute), 97 PRICE_DISTANCE_MISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 rPRICE_DISTANCE_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97
PENDING (oandapyV20.definitions.orders.OrderState attribute), 86 PENDING (oandapyV20.definitions.orders.OrderStateFilte attribute), 83 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.OrderCancelRoattribute), 101 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.OrderCancelRoattribute), 101	attribute), 97 PRICE_DISTANCE_MISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 rPRICE_DISTANCE_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 easbace_INVALID (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97
PENDING (oandapyV20.definitions.orders.OrderState attribute), 86 PENDING (oandapyV20.definitions.orders.OrderStateFilte attribute), 83 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.OrderCancelReattribute), 101 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.TransactionRej	attribute), 97 PRICE_DISTANCE_MISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 rPRICE_DISTANCE_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 easbace_INVALID (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97
PENDING (oandapyV20.definitions.orders.OrderState attribute), 86 PENDING (oandapyV20.definitions.orders.OrderStateFilte attribute), 83 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.OrderCancelReattribute), 101 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.TransactionRejattribute), 97	attribute), 97 PRICE_DISTANCE_MISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 rPRICE_DISTANCE_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 earrow Coan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 earrow Coan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97
PENDING (oandapyV20.definitions.orders.OrderState attribute), 86 PENDING (oandapyV20.definitions.orders.OrderStateFilte attribute), 83 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.OrderCancelReattribute), 101 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.TransactionRejattribute), 97 PLATFORM_ACCOUNT_MIGRATION (oandapyV20.definitions.definitions.transactionRejattribute), 97	attribute), 97 PRICE_DISTANCE_MISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 rPRICE_DISTANCE_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 edsBnICE_INVALID (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 eerregsson dapyV20.definitions.transactions.TransactionRejectReason attribute), 97
PENDING (oandapyV20.definitions.orders.OrderState attribute), 86 PENDING (oandapyV20.definitions.orders.OrderStateFilte attribute), 83 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.OrderCancelRattribute), 101 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.TransactionRejattribute), 97 PLATFORM_ACCOUNT_MIGRATION (oandapyV20.definitions.transactions.FixedPriceOrd	attribute), 97 PRICE_DISTANCE_MISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 rPRICE_DISTANCE_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 edsBACE_INVALID (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 eetreesson dapyV20.definitions.transactions.TransactionRejectReason attribute), 97
PENDING (oandapyV20.definitions.orders.OrderState attribute), 86 PENDING (oandapyV20.definitions.orders.OrderStateFilte attribute), 83 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.OrderCancelRoattribute), 101 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.TransactionRejattribute), 97 PLATFORM_ACCOUNT_MIGRATION (oandapyV20.definitions.transactions.FixedPriceOrdattribute), 92	attribute), 97 PRICE_DISTANCE_MISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 rPRICE_DISTANCE_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 eRBACE_INVALID (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 eRREASONMISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 erreason_attribute), 97 erreason_precision_exceeded (oan-
PENDING (oandapyV20.definitions.orders.OrderState attribute), 86 PENDING (oandapyV20.definitions.orders.OrderStateFilte attribute), 83 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.OrderCancelRoattribute), 101 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.TransactionRejattribute), 97 PLATFORM_ACCOUNT_MIGRATION (oandapyV20.definitions.transactions.FixedPriceOrdattribute), 92	attribute), 97 PRICE_DISTANCE_MISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 rPRICE_DISTANCE_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 earbaice_INVALID (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 eerreasonMISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 eerreason_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97
PENDING (oandapyV20.definitions.orders.OrderState attribute), 86 PENDING (oandapyV20.definitions.orders.OrderStateFilte attribute), 83 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.OrderCancelRoattribute), 101 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.TransactionRejattribute), 97 PLATFORM_ACCOUNT_MIGRATION (oandapyV20.definitions.transactions.FixedPriceOrdattribute), 92 POSITION_AGGREGATION_MODE_INVALID (oandapyV21)	attribute), 97 PRICE_DISTANCE_MISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 rPRICE_DISTANCE_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 earbaice_INVALID (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 eerreasonMISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 eerreason_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97
PENDING (oandapyV20.definitions.orders.OrderState attribute), 86 PENDING (oandapyV20.definitions.orders.OrderStateFilte attribute), 83 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.OrderCancelRoatribute), 101 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.TransactionRejattribute), 97 PLATFORM_ACCOUNT_MIGRATION (oandapyV20.definitions.transactions.FixedPriceOrdattribute), 92 POSITION_AGGREGATION_MODE_INVALID (oandapyV20.definitions.transactions.TransactionRejattribute), 97	attribute), 97 PRICE_DISTANCE_MISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 rPRICE_DISTANCE_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 earthice_invalid (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 earthicasonMISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 earthicason_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 earthicason_omponents (class in oan-
PENDING (oandapyV20.definitions.orders.OrderState attribute), 86 PENDING (oandapyV20.definitions.orders.OrderStateFilte attribute), 83 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.OrderCancelRoatribute), 101 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.TransactionRejattribute), 97 PLATFORM_ACCOUNT_MIGRATION (oandapyV20.definitions.transactions.FixedPriceOrdattribute), 92 POSITION_AGGREGATION_MODE_INVALID (oandapyV20.definitions.transactions.TransactionRejattribute), 97	attribute), 97 PRICE_DISTANCE_MISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 rPRICE_DISTANCE_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 eRBMCE_INVALID (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 ePTREGSONMISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 ePTREGSON_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 ePTREGSON_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 ePTREGSONOMPONENTS (class in oan- dapyV20.definitions.instruments), 82 PriceStatus (class in oan- eason dapyV20.definitions.pricing), 87
PENDING (oandapyV20.definitions.orders.OrderState attribute), 86 PENDING (oandapyV20.definitions.orders.OrderStateFilte attribute), 83 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.OrderCancelReattribute), 101 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.TransactionRegattribute), 97 PLATFORM_ACCOUNT_MIGRATION (oandapyV20.definitions.transactions.FixedPriceOrdattribute), 92 POSITION_AGGREGATION_MODE_INVALID (oandapyV20.definitions.transactions.TransactionRegattribute), 97 POSITION_CLOSEOUT (oandapyV20.definitions.transactions.MarketOrderRegattribute), 93	attribute), 97 PRICE_DISTANCE_MISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 rPRICE_DISTANCE_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 eRBMICE_INVALID (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 ePTREGEONMISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 ePTREGEON_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 ePTREGEONOMPONENTS (class in oan- dapyV20.definitions.instruments), 82 PriceStatus (class in oan- eason dapyV20.definitions.pricing), 87 PriceValue (class in oandapyV20.types), 106
PENDING (oandapyV20.definitions.orders.OrderState attribute), 86 PENDING (oandapyV20.definitions.orders.OrderStateFilte attribute), 83 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.OrderCancelRoattribute), 101 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.TransactionRegattribute), 97 PLATFORM_ACCOUNT_MIGRATION (oandapyV20.definitions.transactions.FixedPriceOrdattribute), 92 POSITION_AGGREGATION_MODE_INVALID (oandapyV20.definitions.transactions.TransactionRegattribute), 97 POSITION_CLOSEOUT (oandapyV20.definitions.transactions.MarketOrderRegattribute), 93 POSITION_CLOSEOUT_FAILED (oandapyV20.definitions.transactions.MarketOrderRegattribute), 93	attribute), 97 PRICE_DISTANCE_MISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 rPRICE_DISTANCE_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 eartribute),
PENDING (oandapyV20.definitions.orders.OrderState attribute), 86 PENDING (oandapyV20.definitions.orders.OrderStateFilte attribute), 83 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.OrderCancelReattribute), 101 PENDING_ORDERS_ALLOWED_EXCEEDED (oandapyV20.definitions.transactions.TransactionReseattribute), 97 PLATFORM_ACCOUNT_MIGRATION (oandapyV20.definitions.transactions.FixedPriceOrdattribute), 92 POSITION_AGGREGATION_MODE_INVALID (oandapyV20.definitions.transactions.TransactionReseattribute), 97 POSITION_CLOSEOUT (oandapyV20.definitions.transactions.MarketOrderReattribute), 93	attribute), 97 PRICE_DISTANCE_MISSING (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 rPRICE_DISTANCE_PRECISION_EXCEEDED (oan- dapyV20.definitions.transactions.TransactionRejectReason attribute), 97 earthibute),

R	SITE_M	IGRATION	(oan-
REDUCE_FIRST (oan	_	dapyV20.definitions.tra	nsactions.FundingReason
dapyV20.definitions.orders.OrderPositionFill		attribute), 91	
attribute) 85	Spread	s (class in oandapyV20.	endpoints.forexlabs), 34
REDUCE_ONLY (oandapyV20.definitions.orders.OrderPo	ositionPillo	andapyV20.definitions.o.	rders.CancellableOrderType
attribute), 85		attribute), 84	
REGULATORY_MARGIN_CALL_VIOLATION (oan		(oandapyV20.definitions	c.orders.OrderType at-
$dapy V20. definitions. transactions. Market Order \\ attribute), 90$	STOP_L	OSS (oandapyV20.defin	itions.orders.CancellableOrderType
REOPEN (oandapy V20. definitions. transactions. Transacti attribute), 94	STOP_L		nitions.orders.OrderType
REPLACEMENT (oandapyV20.definitions.transactions.Lt attribute), 102	210P_L	O22_ON_t TTT_CTTF	NT_ORDER_COMMENT_INVALID
REPLACEMENT (oandapyV20.definitions.transactions.M attribute), 92		auribuie), 98	
REPLACEMENT (oandapyV20.definitions.transactions.St	top£ossØ rde	Preuson_FILL_CLIE	NT_ORDER_ID_ALREADY_EXISTS
attribute), 90		(oanaapyv20.aejiniiior	s.transactions.OrderCancelReason
REPLACEMENT (oandapyV20.definitions.transactions.St	topOrderRed	_{us} affribute), 101	MT ODDED ID IMMALID
attributa) 03	SIOP	ODD ON LITT CTIF	NT_ORDER_ID_INVALID
REPLACEMENT (oandapyV20.definitions.transactions.To	akeProfitOra	erReason v 20.aejimmor attribute), 98	s.transactions.1ransactionRejectReason
attribute), 95 REPLACEMENT (oandapyV20.definitions.transactions.Tr	:1S.T.@PV	* * * * * * * * * * * * * * * * * * *	NT ORDER TAG INVALID
attribute), 102	ranngsto p E	oss vaerkeason =°=== (oandapvV20.definitior	as.transactions.TransactionRejectReason
REPLACING_ORDER_INVALID (oan	_	attribute), 98	,
dapyV20.definitions.transactions.TransactionR	- PeieStROBSOH		ANCE_INVALID (oan-
attribute), 97	ejecireuson	dapyV20.definitions.tra	nsactions.TransactionRejectReason
REPLACING TRADE ID INVALID (oan	_	attribute), 98	
dapyV20.definitions.transactions.TransactionR	eje&Reas o n	OSS_ON_FILL_DIST	ANCE_PRECISION_EXCEEDED
attribute), 98		(oanaapyv 20.aejinitior	s.transactions.TransactionRejectReason
request() (oandapyV20.API method), 8	aman .	attribute), 98	
request_params (oandapyV20.API attribute), 8	STOP_L	OSS_ON_FILL_GTD_	TIMESTAMP_IN_PAST
REQUIRED (oandapy V20. definitions. accounts. Guarante	edStopLoss(үдүрүүүүү 20.аезишог attribute), 101	is.transactions.OraerCancetReason
attribute), 80	STOP I.	* *	TIMESTAMP_IN_PAST
RESET_RESETTABLE_PL (oan			as.transactions.TransactionRejectReason
dapyV20.definitions.transactions.TransactionT ₂ attribute), 94	ype	attribute), 98	sansacrions.11ansacrion11ejeenteuson
unnome), 54	STOP_L	* *	TIMESTAMP_MISSING
S			as.transactions.TransactionRejectReason
\$10 (oandapyV20.definitions.instruments.CandlestickGr	anularity	attribute), 98	
attribute), 81	STOP_L		ANTEED_HEDGING_NOT_ALLOWED
S15 (oandapyV20.definitions.instruments.CandlestickGr	anularity		s.transactions.OrderCancelReason
	•	attribute), 101	
attribute), 81 S30 (oandapyV20.definitions.instruments.CandlestickGr	anularity_L	OSS_ON_FILL_GUAR	ANTEED_LEVEL_RESTRICTION_EXC
attribute) 81		$(\circ \cdots \circ f)$	
S5 (oandapyV20.definitions.instruments.CandlestickGradattribute), 81	nularity	occon etti cuad	NUTEED IEWEI DECTRICTION EVO
		(oandanyV20 definition	an teed_bevel_kestriction_exci as.transactions.TransactionRejectReason
Saturday (oandapy V20. definitions. instruments. Weekly	Alignment	attribute), 98	ыл анынстоны.11 анынстотке jee ixeuson
attribute), 82	STOP T		ANTEED_MINIMUM_DISTANCE_NOT_I
SECOND_BY_SECOND (oan	-		as.transactions.OrderCancelReason
dapyV20.definitions.accounts.AccountFinancin	igMode	attribute), 101	
attribute), 79 secs2time() (in module oan	STOP_L		ANTEED_MINIMUM_DISTANCE_NOT_N
secs2time() (in module oan- dapyV20.contrib.generic), 110	=	(oandapyV20.definition	s.transactions.TransactionRejectReason
aupy , 20.00111110.50110110), 110		attribute), 98	

- STOP_LOSS_ON_FILL_GUARANTEED_NOT_ALLOWEDSTOP_LOSS_ON_FILL_TRIGGER_CONDITION_INVALID (oandapyV20.definitions.transactions.OrderCancelReason (oandapyV20.definitions.transactions.TransactionRejectReason attribute), 101 attribute), 98
- STOP_LOSS_ON_FILL_GUARANTEED_NOT_ALLOWEDSTOP_LOSS_ON_FILL_TRIGGER_CONDITION_MISSING (oandapyV20.definitions.transactions.TransactionRejectReason attribute), 98 attribute), 98
- STOP_LOSS_ON_FILL_GUARANTEED_REQUIRED STOP_LOSS_ORDER (oan-doandapyV20.definitions.transactions.OrderCancelReason dapyV20.definitions.transactions.OrderFillReason attribute), 101 attribute), 91
- STOP_LOSS_ON_FILL_GUARANTEED_REQUIRED STOP_LOSS_ORDER (oan-doandapyV20.definitions.transactions.TransactionRejectReadapyV20.definitions.transactions.TransactionType attribute), 98 attribute), 94
- STOP_LOSS_ON_FILL_LOSS (oan- STOP_LOSS_ORDER_ALREADY_EXISTS (oan-dapyV20.definitions.transactions.OrderCancelReason attribute), 101 dayV20.definitions.transactions.TransactionRejectReason attribute), 98
- STOP_LOSS_ON_FILL_PRICE_AND_DISTANCE_BOT8_TME_SOUS_ORDER_GUARANTEED_HALTED_CREATE_VIOLATION (oandapyV20.definitions.transactions.TransactionRejectRea(soundapyV20.definitions.transactions.TransactionRejectReason attribute), 98

 attribute), 98
- STOP_LOSS_ON_FILL_PRICE_AND_DISTANCE_BOT8_TOPECOD5\$_EORDER_GUARANTEED_HALTED_TIGHTEN_VIOLATION (oandapyV20.definitions.transactions.TransactionRejectRea(coundapyV20.definitions.transactions.TransactionRejectReason attribute), 98

 attribute), 98
- STOP_LOSS_ON_FILL_PRICE_DISTANCE_MAXIMUMSEXPEEDED_ORDER_GUARANTEED_HEDGING_NOT_ALLOWED (oandapyV20.definitions.transactions.OrderCancelReason (oandapyV20.definitions.transactions.TransactionRejectReason attribute), 101 attribute), 98
- STOP_LOSS_ON_FILL_PRICE_DISTANCE_MAXIMUMSEXPEEDED_ORDER_GUARANTEED_LEVEL_RESTRICTION_EXCEED (oandapyV20.definitions.transactions.TransactionRejectRea(xoundapyV20.definitions.transactions.TransactionRejectReason attribute), 98 attribute), 98
- STOP_LOSS_ON_FILL_PRICE_INVALID (oan- STOP_LOSS_ORDER_GUARANTEED_MINIMUM_DISTANCE_NOT_ME' dapyV20.definitions.transactions.TransactionRejectReason (oandapyV20.definitions.transactions.TransactionRejectReason attribute), 98 attribute), 98
- STOP_LOSS_ON_FILL_PRICE_MISSING (oan- STOP_LOSS_ORDER_GUARANTEED_NOT_ALLOWED dapyV20.definitions.transactions.TransactionRejectReason (oandapyV20.definitions.transactions.TransactionRejectReason attribute), 98 attribute), 98
- STOP_LOSS_ON_FILL_PRICE_PRECISION_EXCEED&DOP_LOSS_ORDER_GUARANTEED_PRICE_WITHIN_SPREAD (oandapyV20.definitions.transactions.TransactionRejectRea(commdapyV20.definitions.transactions.TransactionRejectReason attribute), 98 attribute), 98
- STOP_LOSS_ON_FILL_REQUIRED (oan- STOP_LOSS_ORDER_GUARANTEED_REQUIRED dapyV20.definitions.transactions.OrderCancelReason attribute), 101 (oandapyV20.definitions.transactions.TransactionRejectReason attribute), 98
- STOP_LOSS_ON_FILL_REQUIRED_FOR_PENDING_ORDOR_LOSS_ORDER_NOT_CANCELABLE (oan(oandapyV20.definitions.transactions.TransactionRejectReadapyV20.definitions.transactions.TransactionRejectReason
 attribute), 98
 attribute), 98
- STOP_LOSS_ON_FILL_TIME_IN_FORCE_INVALID STOP_LOSS_ORDER_NOT_REPLACEABLE (oan(oandapyV20.definitions.transactions.OrderCancelReason dapyV20.definitions.transactions.TransactionRejectReason
 attribute), 101 attribute), 98
- STOP_LOSS_ON_FILL_TIME_IN_FORCE_INVALID STOP_LOSS_ORDER_PRICE_AND_DISTANCE_BOTH_MISSING (oandapyV20.definitions.transactions.TransactionRejectRea(womndapyV20.definitions.transactions.TransactionRejectReason attribute), 98

 attribute), 98
- STOP_LOSS_ON_FILL_TIME_IN_FORCE_MISSING STOP_LOSS_ORDER_PRICE_AND_DISTANCE_BOTH_SPECIFIED (oandapyV20.definitions.transactions.TransactionRejectReason attribute), 98 attribute), 98

```
STOP ORDER (oandapy V20. definitions. transactions. Order Fill Reason (oandapy V20. definitions. transactions. Order Cancel Reason
                           attribute), 91
                                                                                                                                                                                             attribute), 101
STOP_ORDER (oandapy V20. definitions. transactions. Transa
                          attribute), 94
                                                                                                                                                                                            dapyV20.definitions.transactions.TransactionRejectReason
STOP ORDER REJECT
                                                                                                                                            (oan-
                                                                                                                                                                                             attribute), 99
                          dapyV20.definitions.transactions.TransactionTypeTAKE PROFIT ON FILL PRICE MISSING (oan-
                          attribute), 94
                                                                                                                                                                                            dapyV20.definitions.transactions.TransactionRejectReason
StopLossDetails
                                                                                 (class
                                                                                                                     in
                                                                                                                                              oan-
                          dapyV20.contrib.requests), 122
                                                                                                                                                                 TAKE PROFIT ON FILL PRICE PRECISION EXCEEDED
StopLossOrderReason
                                                                                                                                                                                            (oandapyV20.definitions.transactions.TransactionRejectReason
                                                                                           (class
                                                                                                                          in
                                                                                                                                              oan-
                          dapyV20.definitions.transactions), 90
                                                                                                                                                                                             attribute), 99
                                                                                                                                                                 TAKE_PROFIT_ON_FILL_TIME_IN_FORCE_INVALID
StopLossOrderRequest
                                                                                              (class
                                                                                                                           in
                                                                                                                                              oan-
                          dapyV20.contrib.requests), 116
                                                                                                                                                                                             (oandapyV20.definitions.transactions.TransactionRejectReason
StopOrderReason
                                                                                 (class
                                                                                                                                                                                             attribute), 99
                                                                                                                     in
                                                                                                                                              oan-
                          dapyV20.definitions.transactions), 93
                                                                                                                                                                 TAKE_PROFIT_ON_FILL_TIME_IN_FORCE_MISSING
StopOrderRequest
                                                                                   (class
                                                                                                                                                                                             (oandapyV20.definitions.transactions.TransactionRejectReason
                                                                                                                                              oan-
                          dapyV20.contrib.requests), 117
                                                                                                                                                                                             attribute), 99
STREAM (oandapyV20.endpoints.pricing.PricingStream
                                                                                                                                                                 TAKE PROFIT ON FILL TRIGGER CONDITION INVALID
                          attribute), 60
                                                                                                                                                                                            (oandapyV20.definitions.transactions.TransactionRejectReason
STREAM (oandapy V20.endpoints.transactions.TransactionsStream
                                                                                                                                                                                            attribute), 99
                          attribute), 77
                                                                                                                                                                 TAKE_PROFIT_ON_FILL_TRIGGER_CONDITION_MISSING
Sunday (oandapy V20. definitions. instruments. Weekly Alignment
                                                                                                                                                                                             (oandapyV20.definitions.transactions.TransactionRejectReason
                                                                                                                                                                                             attribute), 99
                          attribute), 82
                                                                                                                                                                 TAKE PROFIT ORDER
Т
                                                                                                                                                                                             dapyV20.definitions.transactions.OrderFillReason
{\tt TAKE\_PROFIT} (o and apy V20. definitions. orders. Cancellable Order Type tribute), 91
                                                                                                                                                                 TAKE_PROFIT_ORDER
                                                                                                                                                                                                                                                                                                              (oan-
                          attribute), 84
                                                                                                                                                                                            dapyV20.definitions.transactions.TransactionType
TAKE PROFIT (oandapy V20.definitions.orders.OrderType
                                                                                                                                                                                            attribute), 95
                          attribute), 84
TAKE_PROFIT_ON_FILL_CLIENT_ORDER_COMMENT_AKUALROFIT_ORDER_ALREADY_EXISTS
                                                                                                                                                                                                                                                                                                              (oan-
                           (o and apy V20. definitions. transactions. Transaction Reject Readony V20. definitions. transactions. Transaction Reject Reason Reject Readony V20. definitions. Transaction Reject Readony V20. definition Reject Reado
                                                                                                                                                                                            attribute), 99
                          attribute), 99
TAKE_PROFIT_ON_FILL_CLIENT_ORDER_ID_ALREAAKE_ERRGFST_ORDER_REJECT
                           (o and apy V20. definitions. transactions. Order Cancel Reason\ dapy V20. definitions. transactions. Transaction Type
                                                                                                                                                                                             attribute), 95
                          attribute), 101
TAKE_PROFIT_ON_FILL_CLIENT_ORDER_ID_INVATabeProfitDetails
                                                                                                                                                                                                                                                        (class
                                                                                                                                                                                                                                                                                         in
                                                                                                                                                                                                                                                                                                                oan-
                          (o and apy V20. definitions. transactions. Transaction Reject Read \textit{eny} V20. contrib. requests), 123
                                                                                                                                                                 TakeProfitOrderReason
                                                                                                                                                                                                                                                                  (class
                                                                                                                                                                                                                                                                                              in
                                                                                                                                                                                                                                                                                                                oan-
                          attribute), 99
                                                                                                                                                                                            dapyV20.definitions.transactions), 95
TAKE_PROFIT_ON_FILL_CLIENT_ORDER_TAG_INVALID
                          (o and a py V20. definitions. transactions. Transaction {\it Reject Reason} {\it tOrderRequest}
                                                                                                                                                                                                                                                                     (class
                                                                                                                                                                                                                                                                                                                oan-
                                                                                                                                                                                            dapyV20.contrib.requests), 118
                          attribute), 99
{\tt TAKE\_PROFIT\_ON\_FILL\_GTD\_TIMESTAMP\_IN\_PAS} \\ {\tt Terminate()} \ (oandapy V20. endpoints. pricing. Pricing Stream and the property of the pr
                          (o and a py V20. definitions. transactions. Order Cancel Reason\ method), 63
                                                                                                                                                                 terminate()(oandapyV20.endpoints.transactions.TransactionsStream
                          attribute), 101
                                                                                                                                                                                            method), 78
TAKE PROFIT ON FILL GTD TIMESTAMP IN PAST
                           (o and apy V20. definitions. transactions. Transaction \ref{Reposition} \cite{Reposition} \cite{Reposition} \cite{V20}. definitions. instruments. Weekly Alignment
                                                                                                                                                                                            attribute), 82
                          attribute), 99
TAKE_PROFIT_ON_FILL_GTD_TIMESTAMP_MISSIN&IME_IN_FORCE_EXPIRED
                                                                                                                                                                                                                                                                                                              (oan-
                           (o and apy V20. definitions. transactions. Transaction Reject Readony V20. definitions. transactions. Order Cancel Reason the Control of th
                                                                                                                                                                                             attribute), 101
                          attribute), 99
                                                                                                                                            (oan- TIME_IN_FORCE_GTD_TIMESTAMP_IN_PAST
TAKE PROFIT ON FILL LOSS
                                                                                                                                                                                            (oandapyV20.definitions.transactions.TransactionRejectReason
                          dapyV20.definitions.transactions.OrderCancelReason
                                                                                                                                                                                             attribute), 99
                          attribute), 101
TAKE_PROFIT_ON_FILL_PRICE_DISTANCE_MAXIMUM_EXENEBEBCE_GTD_TIMESTAMP_MISSING
```

```
(oandapyV20.definitions.transactions.TransactionRejectReadapyV20.definitions.orders.OrderType
        attribute), 99
                                                            tribute), 84
TIME IN FORCE INVALID
                                            (oan- TRAILING STOP LOSS ON FILL CLIENT ORDER COMMENT IN
        dapyV20.definitions.transactions.TransactionRejectReason (oandapyV20.definitions.transactions.TransactionRejectReason
        attribute), 99
                                                            attribute), 99
TIME IN FORCE MISSING
                                            (oan- TRAILING STOP LOSS ON FILL CLIENT ORDER ID ALREADY
        dapyV20.definitions.transactions.TransactionRejectReason (oandapyV20.definitions.transactions.OrderCancelReason
        attribute), 99
                                                            attribute), 101
TimeInForce
                      (class
                                   in
                                             oan-
                                                   TRAILING STOP LOSS ON FILL CLIENT ORDER ID INVALID
        dapyV20.definitions.orders), 85
                                                            (oandapyV20.definitions.transactions.TransactionRejectReason
TRADE_CLIENT_EXTENSIONS_MODIFY
                                            (oan-
                                                            attribute), 99
        dapyV20.definitions.transactions.TransactionTypeTRAILING_STOP_LOSS_ON_FILL_CLIENT_ORDER_TAG_INVALID
                                                            (o and a py V20. definitions. transactions. Transaction Reject Reason\\
        attribute), 95
TRADE_CLIENT_EXTENSIONS_MODIFY_REJECT
                                                            attribute), 99
        (oandapyV20.definitions.transactions.TransactionType:ILING_STOP_LOSS_ON_FILL_GTD_TIMESTAMP_IN_PAST
        attribute), 95
                                                            (oandapyV20.definitions.transactions.OrderCancelReason
TRADE_CLOSE (oandapyV20.definitions.transactions.MarketOrderReatsrobute), 101
        attribute), 93
                                                   TRAILING STOP LOSS ON FILL GTD TIMESTAMP IN PAST
TRADE DOESNT EXIST
                                                            (oandapyV20.definitions.transactions.TransactionRejectReason
                                            (oan-
        dapyV20.definitions.transactions.TransactionRejectReason attribute), 99
        attribute), 99
                                                   TRAILING_STOP_LOSS_ON_FILL_GTD_TIMESTAMP_MISSING
TRADE ID UNSPECIFIED
                                            (oan-
                                                            (oandapyV20.definitions.transactions.TransactionRejectReason
        dapyV20.definitions.transactions.TransactionRejectReason attribute), 99
        attribute), 99
                                                   TRAILING STOP LOSS ON FILL PRICE DISTANCE INVALID
TRADE_IDENTIFIER_INCONSISTENCY
                                                            (oandapyV20.definitions.transactions.TransactionRejectReason
                                            (oan-
        dapyV20.definitions.transactions.TransactionRejectReason attribute), 99
        attribute), 99
                                                   TRAILING_STOP_LOSS_ON_FILL_PRICE_DISTANCE_MAXIMUM_1
TRADE_ON_FILL_CLIENT_EXTENSIONS_NOT_SUPPORTED (oandapyV20.definitions.transactions.TransactionRejectReason
        (oandapyV20.definitions.transactions.TransactionRejectReastmibute), 99
        attribute), 99
                                                   TRAILING_STOP_LOSS_ON_FILL_PRICE_DISTANCE_MINIMUM_1
tradeable (oandapyV20.definitions.pricing.PriceStatus
                                                            (oandapyV20.definitions.transactions.TransactionRejectReason
        attribute), 87
                                                            attribute), 99
                                                   TRAILING_STOP_LOSS_ON_FILL_PRICE_DISTANCE_MISSING
TradeClientExtensions
                                             oan-
                                                            (oandapyV20.definitions.transactions.TransactionRejectReason
        dapyV20.endpoints.trades), 66
TradeClose (class in oandapyV20.endpoints.trades),
                                                            attribute), 99
                                                   TRAILING_STOP_LOSS_ON_FILL_PRICE_DISTANCE_PRECISION
TradeCloseRequest
                           (class
                                      in
                                             oan-
                                                            (oandapyV20.definitions.transactions.TransactionRejectReason
        dapyV20.contrib.requests), 119
TradeCRCDO (class in oandapyV20.endpoints.trades),
                                                   TRAILING_STOP_LOSS_ON_FILL_TIME_IN_FORCE_INVALID
                                                            (oandapyV20.definitions.transactions.TransactionRejectReason
        64
TradeDetails
                                                            attribute), 99
                       (class
                                             oan-
        dapyV20.endpoints.trades), 69
                                                   TRAILING STOP LOSS ON FILL TIME IN FORCE MISSING
TradeID (class in oandapyV20.types), 106
                                                            (oandapyV20.definitions.transactions.TransactionRejectReason
TradePL (class in oandapyV20.definitions.trades), 88
                                                            attribute), 99
TradesList (class in oandapyV20.endpoints.trades),
                                                   TRAILING_STOP_LOSS_ON_FILL_TRIGGER_CONDITION_INVAL
                                                            (oandapyV20.definitions.transactions.TransactionRejectReason
TradeState (class in oandapyV20.definitions.trades),
                                                            attribute), 99
                                                   TRAILING_STOP_LOSS_ON_FILL_TRIGGER_CONDITION_MISSI
        88
TradeStateFilter
                          (class
                                     in
                                                            (oandapyV20.definitions.transactions.TransactionRejectReason
                                             oan-
        dapyV20.definitions.trades), 89
                                                            attribute), 99
TRAILING_STOP_LOSS
                                                   TRAILING_STOP_LOSS_ORDER
                                                                                                (oan-
                                            (oan-
        dapyV20.definitions.orders.CancellableOrderType
                                                            dapyV20.definitions.transactions.OrderFillReason
        attribute), 84
                                                            attribute), 91
TRAILING_STOP_LOSS
                                            (oan- TRAILING_STOP_LOSS_ORDER
                                                                                                (oan-
```

```
dapyV20.definitions.transactions.TransactionType
                                                                attribute), 100
        attribute), 95
                                                      UNITS LIMIT EXCEEDED
                                                                                                      (oan-
TRAILING STOP LOSS ORDER ALREADY EXISTS
                                                                dapyV20.definitions.transactions.TransactionRejectReason
         (oandapyV20.definitions.transactions.TransactionRejectReastmibute), 100
         attribute), 100
                                                      UNITS MIMIMUM NOT MET
                                                                                                      (oan-
TRAILING STOP LOSS ORDER REJECT
                                                                dapyV20.definitions.transactions.TransactionRejectReason
                                               (oan-
        dapyV20.definitions.transactions.TransactionType
                                                                attribute), 100
                                                      UNITS MISSING
         attribute), 95
                                                                                                      (oan-
TRAILING STOP LOSS ORDERS NOT SUPPORTED
                                                                dapyV20.definitions.transactions.TransactionRejectReason
         (oandapyV20.definitions.transactions.TransactionRejectReastmibute), 100
         attribute), 99
                                                      UNITS_PRECISION_EXCEEDED
                                                                                                      (oan-
TrailingStopLossDetails
                                  (class
                                                                dapyV20.definitions.transactions.TransactionRejectReason
                                                oan-
         dapyV20.contrib.requests), 124
                                                                attribute), 100
TrailingStopLossOrderReason (class in oan-
                                                      V
         dapyV20.definitions.transactions), 101
TrailingStopLossOrderRequest (class in oan-
                                                      V20Error (class in oandapyV20), 8
         dapyV20.contrib.requests), 120
                                                      value (oandapyV20.types.AccountID attribute), 104
TransactionDetails
                                         in
                                                oan-
                                                      value (oandapyV20.types.AccountUnits attribute), 104
         dapyV20.endpoints.transactions), 71
                                                      value (oandapyV20.types.ClientComment attribute),
TransactionIDRange
                              (class
                                         in
                                                oan-
         dapyV20.endpoints.transactions), 72
                                                      value (oandapyV20.types.ClientID attribute), 104
TransactionList
                           (class
                                       in
                                                oan-
                                                      value (oandapyV20.types.ClientTag attribute), 104
         dapyV20.endpoints.transactions), 74
                                                      value (oandapyV20.types.DateTime attribute), 105
TransactionRejectReason
                                  (class
                                           in
                                                oan-
                                                      value (oandapyV20.types.OrderID attribute), 105
         dapyV20.definitions.transactions), 95
                                                               (oandapyV20.types.OrderIdentifier attribute),
                                                      value
TransactionsSinceID
                              (class
                                         in
                                                oan-
         dapyV20.endpoints.transactions), 75
                                                      value (oandapyV20.types.OrderSpecifier attribute), 106
TransactionsStream
                              (class
                                         in
                                                oan-
                                                      value (oandapyV20.types.PriceValue attribute), 106
         dapyV20.endpoints.transactions), 77
                                                      value (oandapyV20.types.TradeID attribute), 107
TransactionType
                           (class
                                       in
                                                oan-
                                                      value (oandapyV20.types.Units attribute), 107
         dapyV20.definitions.transactions), 93
TRANSFER_FUNDS
                                               (oan-
        - dapy V20. definitions. transactions. Transaction Type_{\mathbb{W}\ (o and apy V20. definitions. instruments. Candle stick Granularity)}
        attribute), 95
                                                                attribute), 82
TRANSFER FUNDS REJECT
                                               (oan-
                                                      Wednesday (oandapy V20. definitions. instruments. Weekly Alignment
        dapyV20.definitions.transactions.TransactionType
                                                                attribute), 82
        attribute), 95
                                                      WeeklyAlignment
                                                                                  (class
                                                                                              in
                                                                                                       oan-
TRIGGER_CONDITION_INVALID
                                               (oan-
                                                                dapyV20.definitions.instruments), 82
         dapyV20.definitions.transactions.TransactionRejectReason
                                                      Z
        attribute), 100
TRIGGER_CONDITION MISSING
                                               (oan-
        dapyV20.definitions.transactions.TransactionRejectReason
                                                                (oandapyV20.definitions.trades.TradePL
                                                                                                        at-
                                                                tribute), 88
        attribute), 100
TRIGGERED (oandapyV20.definitions.orders.OrderState
         attribute), 86
TRIGGERED (oandapyV20.definitions.orders.OrderStateFilter
         attribute), 83
Tuesday (oandapy V20. definitions. instruments. Weekly Alignment
         attribute), 82
U
Units (class in oandapy V20.types), 107
UNITS_INVALID
                                               (oan-
         dapy V20. definitions. transactions. Transaction Reject Reason
```