

TRADING AT ALPACA

KERRY BACK AND KEVIN CROTTY



CREATE ALPACA ACCOUNT

- After logging into Alpaca at <https://alpaca.markets>, you DO NOT need to enter name, address, etc. - that is only for real trading accounts.
- By clicking **Home** on the left toolbar, you should get to the screen on the following slide.
- Click on **View API Keys**, then **Generate New Keys** to see your key and your secret key. Copy and save them somewhere.

Paper
PA326D2U1D75

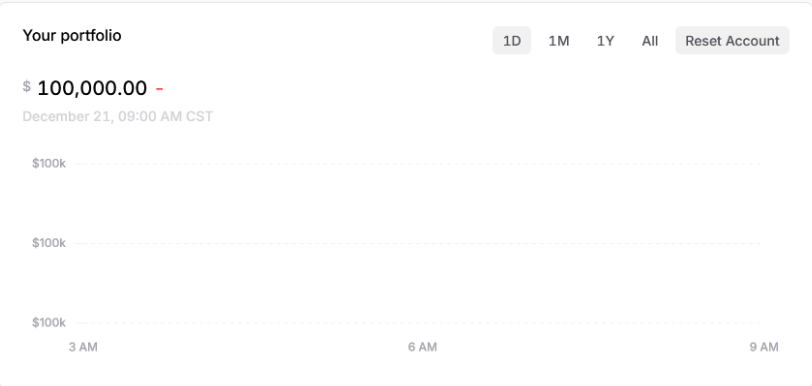
- Home
- Account >
- Alpaca Connect
- Subscription
- Resources >
- Community >
- Support

v3.12.946

You are on Paper Trading, no real money is being used.

Open Live Account

Good morning, Kevin! 😊
Here you can see an overview of your account.



Quick Trade View API Keys

Buy Sell

Symbol Market Price
Enter here \$0.00

Order Type
Market

Quantity Estimated Cost
1 \$0.00

Time in Force
GTC - Good till Canceled

Review Order

Buying Power	Cash	Daily Change
\$200,000	\$100,000	-

Top Positions

Asset Class All Side All View All

Asset	Price	Qty	Market Value	Total P/L (\$)
No open positions. Place some trades to see this table populate				

Watchlist Edit

No assets in watchlist

INSTALL ALPACA-PY

- The module `alpaca-trade-api` is deprecated. Install `alpaca-py` instead.
- The docs are at <https://docs.alpaca.markets/docs/getting-started-with-trading-api>.

CONNECT TO ALPACA

```
1 from alpaca.trading.client import TradingClient
2 from alpaca.trading.requests import MarketOrderRequest
3 from alpaca.trading.enums import OrderSide, TimeInForce
4
5 KEY = "your_key"
6 SECRET_KEY = "your_secret_key"
7
8 trading_client = TradingClient(KEY, SECRET_KEY, paper=True)
```

CHECK YOUR ACCOUNT

```
1 account = trading_client.get_account()
```

```
2 account
```

```
{  'account_blocked': False,  
   'account_number': 'PA326D2U1D75',  
   'accrued_fees': '0',  
   'buying_power': '194344.44',  
   'cash': '96279.78',  
   'created_at': datetime.datetime(2023, 12, 21, 14, 56, 37,  
240661, tzinfo=datetime.timezone.utc),  
   'crypto_status': <AccountStatus.ACTIVE: 'ACTIVE'>,  
   'currency': 'USD',  
   'daytrade_count': 6,  
   'daytrading_buying_power': '0',  
   'equity': '100005.41',
```

CHECK BUYING POWER AND ACCOUNT BALANCE

```
1 print(f'Available buying power:\t${float(account.buying_power)}  
2 print(f'Portfolio value:\t\t${float(account.portfolio_value):,
```

```
Available buying power: $194,344.44  
Portfolio value:          $100,005.41
```

ASSET INFORMATION

```
1 asset_info = trading_client.get_asset('SPY')
2 asset_info
```

```
{ 'asset_class': <AssetClass.US_EQUITY: 'us_equity'>,
  'easy_to_borrow': True,
  'exchange': <AssetExchange.ARCA: 'ARCA'>,
  'fractionable': True,
  'id': UUID('b28f4066-5c6d-479b-a2af-85dc1a8f16fb'),
  'maintenance_margin_requirement': 30.0,
  'marginable': True,
  'min_order_size': None,
  'min_trade_increment': None,
  'name': 'SPDR S&P 500 ETF Trust',
  'price_increment': None,
  'shortable': True,
```


CURRENT QUOTES

```
1 from alpaca.data import StockHistoricalDataClient
2 from alpaca.data.requests import StockLatestQuoteRequest
3 data_client = StockHistoricalDataClient(KEY, SECRET_KEY)
4 params = StockLatestQuoteRequest(symbol_or_symbols=['SPY'])
5 quotes = data_client.get_stock_latest_quote(params)
6 quotes
```

```
{'SPY': {'ask_exchange': 'V',
         'ask_price': 469.6,
         'ask_size': 1.0,
         'bid_exchange': 'V',
         'bid_price': 469.55,
         'bid_size': 2.0,
         'conditions': ['R'],
         'symbol': 'SPY',
         'tape': 'B',
         'timestamp': datetime.datetime(2023, 12, 21, 17, 27, 45,
137877, tzinfo=datetime.timezone.utc)}}
```

BUY SPY

```
1 order = MarketOrderRequest(  
2     symbol="SPY",  
3     qty=10,  
4     side=OrderSide.BUY,  
5     time_in_force=TimeInForce.DAY  
6 )  
7 _ = trading_client.submit_order(order)
```

SHORT AAPL

```
1 order = MarketOrderRequest(  
2     symbol="AAPL",  
3     qty=5,  
4     side=OrderSide.SELL,  
5     time_in_force=TimeInForce.DAY  
6 )  
7 _ = trading_client.submit_order(order)
```

CHECK YOUR POSITIONS

```
1 positions = trading_client.get_all_positions()
2 for p in positions:
3     print("{} shares of {}".format(p.qty, p.symbol))
```

-5 shares of AAPL

10 shares of SPY

```
1 account = trading_client.get_account()
2 print(f'Cash Balance: ${float(account.cash):,.2f}')
```

Cash Balance: \$96,279.78