Copy Trading Bot Code Explanation

1. **Importing Required Libraries:**

pythonCopy code

import alpaca_trade_api as tradeapi

This line imports the **alpaca_trade_api** library, which provides access to the Alpaca trading platform through its API.

2. **API Keys and Base URL:**

pythonCopy code

MASTER_API_KEY = 'YOUR_MASTER_API_KEY'

MASTER_SECRET_KEY = 'YOUR_MASTER_SECRET_KEY'

CLIENT_API_KEY = 'YOUR_CLIENT_API_KEY'

CLIENT_SECRET_KEY = 'YOUR_CLIENT_SECRET_KEY'

BASE_URL = 'https://paper-api.alpaca.markets'

These variables should be replaced with your actual Alpaca API keys. The **BASE_URL** specifies the Alpaca Paper API URL, which is used for testing and development purposes.

3. **Risk Management Configuration:**

pythonCopy code

MAX_POSITION_SIZE_PERCENT = 10

This variable sets the maximum allowed position size as a percentage of the client account's portfolio equity. In this example, it's set to 10%, meaning that no individual position should exceed 10% of the account's equity.

4. **Initializing API Clients:**

pythonCopy code

master_api = tradeapi.REST(MASTER_API_KEY, MASTER_SECRET_KEY, BASE_URL, api_version='v2')

client_api = tradeapi.REST(CLIENT_API_KEY, CLIENT_SECRET_KEY, BASE_URL, api_version='v2')

These lines initialize two Alpaca API clients: **master_api** for the master account and **client_api** for the client account. These clients are used to interact with the Alpaca API.

5. **Main Execution:**

pythonCopy code

try:

 # Get the open positions from the master account

```
    master_positions = master_api.list_positions()

    # Get the client account's portfolio equity
    client_portfolio = client_api.get_account()
    client_equity = float(client_portfolio.equity)

    # Open the same positions in the client account
    for position in master_positions:
        try:
            # Calculate maximum allowed position size based on risk management
            max_position_size = client_equity * MAX_POSITION_SIZE_PERCENT / 100

            if position.market_value <= max_position_size:
                # Submit order to open position in client account
                client_api.submit_order(
                    symbol=position.symbol,
                    qty=int(position.qty),
                    side=position.side.lower(),
                    type='market',
                    time_in_force='gtc'
                )
                print(f"Successfully copied {position.qty} shares of {position.symbol} to client account.")
            else:
                print(f"Skipping {position.symbol} due to excessive position size risk.")
        except tradeapi.rest.APIError as e:
            print(f"Error copying {position.symbol} to client account: {e}")
except tradeapi.rest.APIError as e:
    print(f"Error accessing master account positions: {e}")
```

Here's what the main execution part of the code does:

- It retrieves the open positions from the master account using **master_api.list_positions()**.

- It fetches the client account's portfolio equity using **client_api.get_account()** and converts it to a floating-point number.
- It iterates through each position in the master account and calculates the maximum allowed position size based on the risk management configuration.
- If the position's market value is within the allowed size, it submits an order to open the same position in the client account.
- If the position's market value exceeds the allowed size, it prints a message indicating that the position is skipped due to excessive risk.
- If any errors occur during the process, they are caught and handled with appropriate error messages.

Remember to replace the placeholders (**'YOUR_MASTER_API_KEY'**, etc.) with your actual API keys, and thoroughly test any trading code in a safe environment before using it with real funds.