Automate Trading using MT5 and Python - Part 2 | Quantra Classroom

Team Quantra <quantra@quantinsti.com>
To: <ejeohejidavid@gmail.com>

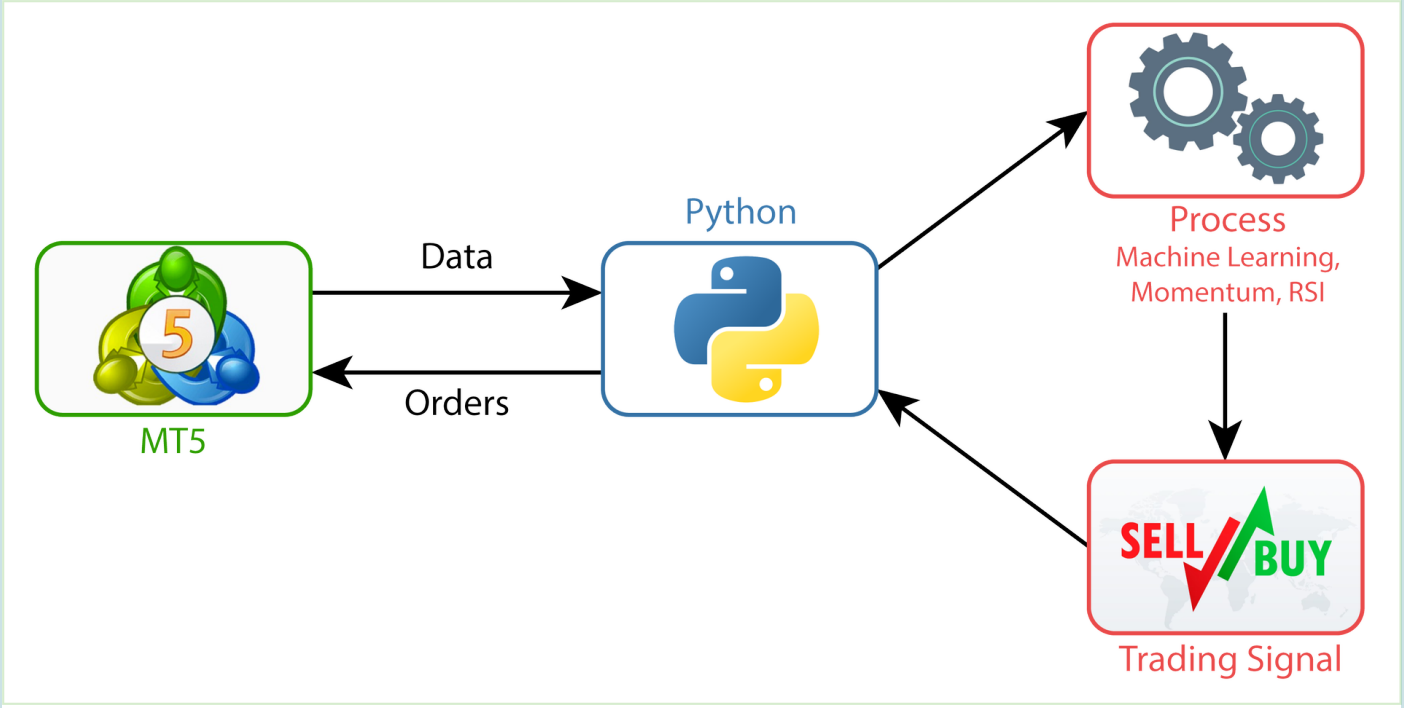
Wed, 28 Jun, 19:02

Hi David Eje,

Today we'll be continuing from the first part of this series, where we covered the necessary setup and how to obtain data. The first part of the content or steps 1-5 can be accessed here.

The idea behind automating trading strategies using Python and MT5 is to use the capability of Python powerful libraries for analyzing data and generating trading signals and use MT5 to place orders automatically. This can improve the quality of signal and reduce the likelihood of errors.

In this part of the series, we'll be covering how to generate trading signals, place orders and close open positions.



Step 6: Trading Signals

To generate trading signals, we need to analyze the data we've collected from the MT5. This can be done using different techniques such as machine learning algorithms, which can be used to identify patterns and trends in the data.

Alternatively, we can use momentum strategies, which rely on the idea that stocks that have performed well in the past will continue to perform well in the future, and vice versa. On the other hand, mean reversion strategies rely on the assumption that the stocks will eventually return to their average price, after deviating from it.

We can also use popular technical indicators such as moving averages, Bollinger bands, and Relative Strength Index (RSI) to generate trading signals. These indicators help us to identify trends, overbought and oversold conditions, and potential entry and exit points.

If you're interested in learning more about these strategies and other trading concepts, Quantra is a great resource. They offer over 50 courses on quantitative and algo trading, covering everything from beginner concepts to advanced strategies.

Explore Courses

Step 8: Place Order

In order to place an order when the entry condition is met, we need to create a dictionary that contains the details for the order placement. This dictionary includes the following parameters:

```
request = {
    "action": mt5.TRADE_ACTION_DEAL,
    "symbol": 'EURUSD',
    "volume": 0.2,
    "type": mt5.ORDER_TYPE_BUY,
    "price": mt5.symbol_info_tick('EURUSD').ask,
    "comment": "Quantra Market Buy Order",
}
```

- "action": This parameter specifies the type of trade action that you want to perform. In this case, we are using "mt5.TRADE_ACTION_DEAL", which means that the trade will be executed immediately at the current market price.
- "symbol": This parameter specifies the trading instrument that you want to trade. You need to replace "symbol" with the symbol name of the trading instrument you want to trade, such as "EURAUD" or "GBPUSD".
- "volume": This parameter specifies the lot size of the trade
- "type": This parameter specifies the type of trade that you want to place, such as a buy/sell market order, limit order, or stop order.

To place a market order, you can set "trade_type" to "mt5.ORDER_TYPE_BUY" or "mt5.ORDER_TYPE_SELL", depending on whether you want to buy or sell the trading

instrument.

To place a limit order, you can set "trade_type" to "mt5.ORDER_TYPE_BUY_LIMIT" or "mt5.ORDER_TYPE_SELL_LIMIT", depending on whether you want to buy or sell the trading instrument.

You also need to specify the "price" parameter, which specifies the price at which you want to enter the trade.

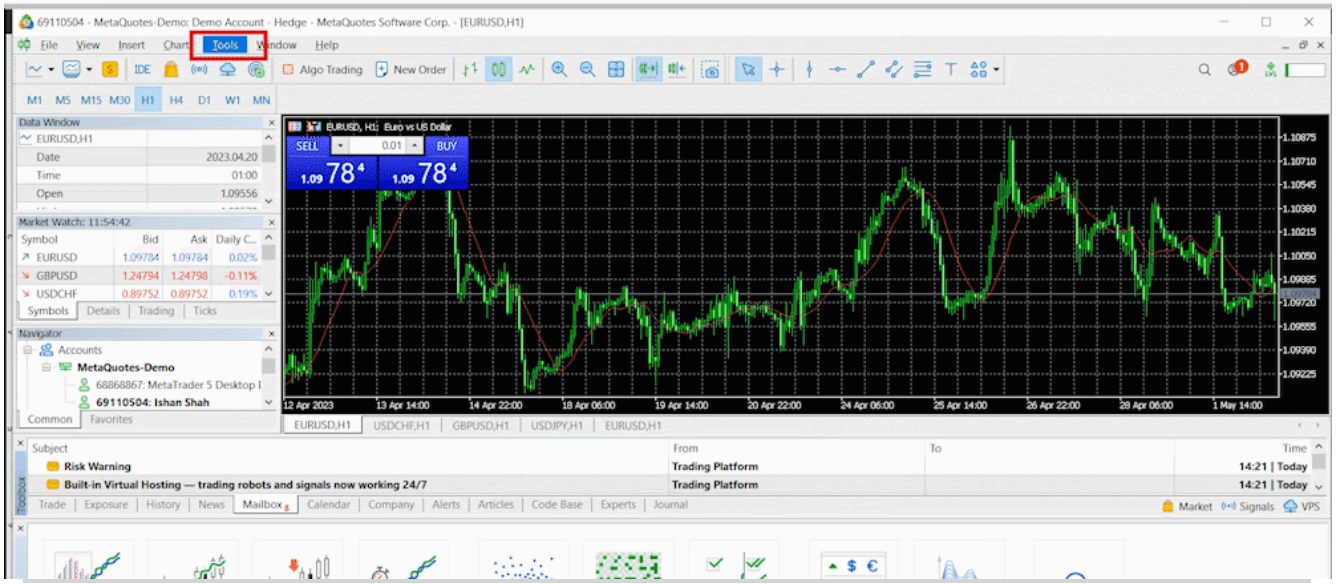
- "comment": This parameter allows you to add a comment to your order. This can be useful for tracking your orders in the future.

Once you have set the parameters, you can use the "mt5.order_send()" function to send the trade request to the MetaTrader5 platform. The function will return a trade ticket number if the trade is executed successfully.

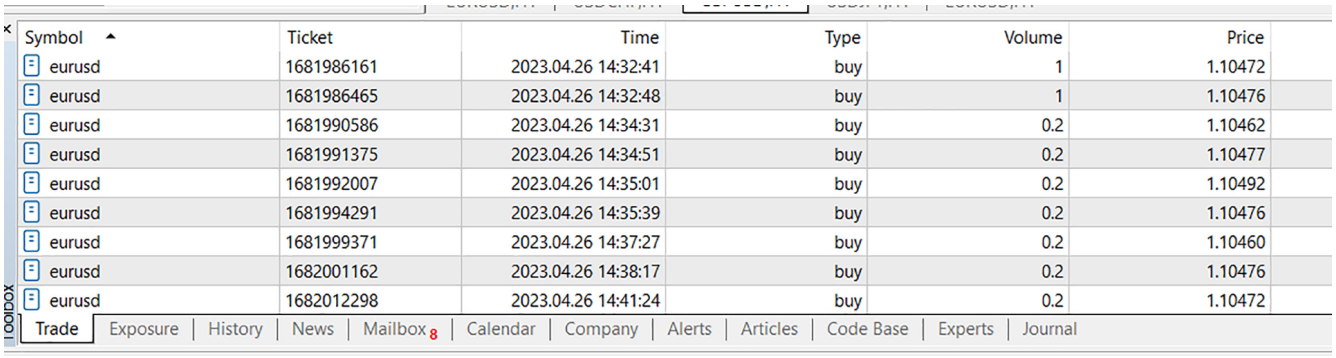
```
# send the order request and check for errors

order_result = mt5.order_send(request)

if order_result.retcode != mt5.TRADE_RETCODE_DONE:

    print("Error placing order: ", order_result.comment)

else:

    print("Order placed successfully, order ticket:", order_result.order)
```

If you receive an error message that says "Automated trading is not enabled" when trying to place orders using the MetaTrader5 package in Python, you can follow these steps:

1. Open the MetaTrader5 desktop application.
2. Click on the "Tools" menu and select "Options."
3. In the "Options" window, click on the "Expert Advisors" tab.
4. Make sure that the "Allow algorithmic trading" checkbox is selected.
5. Click "OK" to save the changes.



Once you've enabled automated trading, you should be able to place orders using the MetaTrader5 package in Python without any issues. You can also verify the execution of order from MetaTrader Terminal.



Symbol	Ticket	Time	Type	Volume	Price
eurusd	1681986161	2023.04.26 14:32:41	buy	1	1.10472
eurusd	1681986465	2023.04.26 14:32:48	buy	1	1.10476
eurusd	1681990586	2023.04.26 14:34:31	buy	0.2	1.10462
eurusd	1681991375	2023.04.26 14:34:51	buy	0.2	1.10477
eurusd	1681992007	2023.04.26 14:35:01	buy	0.2	1.10492
eurusd	1681994291	2023.04.26 14:35:39	buy	0.2	1.10476
eurusd	1681999371	2023.04.26 14:37:27	buy	0.2	1.10460
eurusd	1682001162	2023.04.26 14:38:17	buy	0.2	1.10476
eurusd	1682012298	2023.04.26 14:41:24	buy	0.2	1.10472

Trade | Exposure | History | News | Mailbox 8 | Calendar | Company | Alerts | Articles | Code Base | Experts | Journal

Step 9: Monitor Position

Once the order has been filled, you can monitor open positions and the unrealized profit and loss (P&L) using the following code:

```
# Positions
result = mt5.positions_get()
if result:
    # create a list of dictionaries containing the data for each position
    data = pd.DataFrame([position._asdict() for position in result])
```

```
    print("Unrealized P&L: ", data.profit.sum())
    # print the DataFrame
    display(data.head())
else:
    print("No positions found")
```

This code retrieves the position information and calculates the unrealized P&L for all the open positions. If there are no positions found, the code will print a message indicating that no positions were found.

Step 10: Close Open Position

To close an open position, you can use the order_send function used previously. You also need to specify the ticket of the order or position you are closing. Here's an example code snippet that shows how to close an open position:

```
# connect to MetaTrader 5
mt5.initialize()
# get the ticket number of the position to close
ticket = int(data.iloc[0].ticket)
# check if the position exists and its type
position = mt5.positions_get(ticket=ticket)
if position:
    if position[0].type == mt5.ORDER_TYPE_BUY:
        # if the position is a buy position, send a sell order to close it
        request = {
            "action": mt5.TRADE_ACTION_DEAL,
            "symbol": position[0].symbol,
            "volume": position[0].volume,
            "type": mt5.ORDER_TYPE_SELL,
            "position": position[0].ticket,
            "price": mt5.symbol_info_tick(position[0].symbol).bid,
```

```
        }
    else:
        # if the position is a sell position, send a buy order to close it
        request = {
            "action": mt5.TRADE_ACTION_DEAL,
            "symbol": position[0].symbol,
            "volume": position[0].volume,
            "type": mt5.ORDER_TYPE_BUY,
            "position": position[0].ticket,
            "price": mt5.symbol_info_tick(position[0].symbol).ask,
        }
    # close the position
    result = mt5.order_send(request)
    if result.retcode != mt5.TRADE_RETCODE_DONE:
        print("Error closing position: ", result.comment)
    else:
        print(f"Position {position[0].ticket} closed \
successfully, order ticket: {result.order}")
else:
    print(f"Position {ticket} not found")
```

If the position is closed successfully, the code will print a message indicating that the position was closed successfully. If the position close fails for some reason, the code will print an error message indicating the reason for the failure.

We hope this email was helpful and you found the information about automating your strategy using MT5 and Python useful. Once you get the hang of automation, you'll be able to focus on researching and backtesting strategies that generate alphas. It's an exciting path to explore!

If you're interested in learning more about trading strategies, you might want to check Quantra to help you dive deeper into the world of algorithmic trading.

If you run into any issues while following the steps specified in the email, don't hesitate to reach out to the Quantra community for help.

References:

1. MetaTrader5 Python documentation

2. Algo Trading for Beginners

Author: Ishan Shah

We'd love to have you join our community of learners and traders.

Connect with us to discuss more. We'd be happy to help!