

SUM3API: Using Rust, ZeroMQ, and MetaQuotes Language (MQL5) API Combination to Extract, Communicate, and Externally Project Financial Data from MetaTrader 5 (MT5)

Author: Albeos Rembrant **ORCID:** <https://orcid.org/0009-0006-8743-4419>

A GUIDE

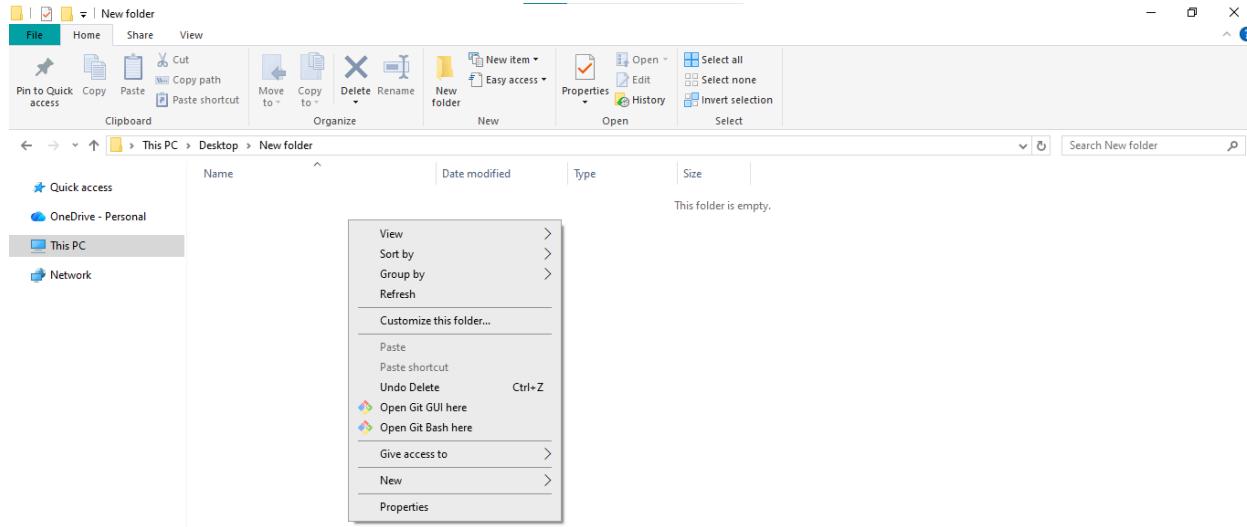
In my case, I've already installed applications such as Git, Rust, VisualStudio C++ Build tools, MetaTrader5, and MetaEditor5. Overall, I'm working on Google Antigravity (you may use any code editor) as long as we have a terminal. Any version would do.

The implementation is done inside a 7-year old Windows 10 Pro 64-bit laptop with 8GB RAM and Intel Core i5-7200U CPU @ 2.50GHz 2.71 GHz, NVIDIA GeForce 940MX, and 256GB SSD
(I'm broke but can stop learning)

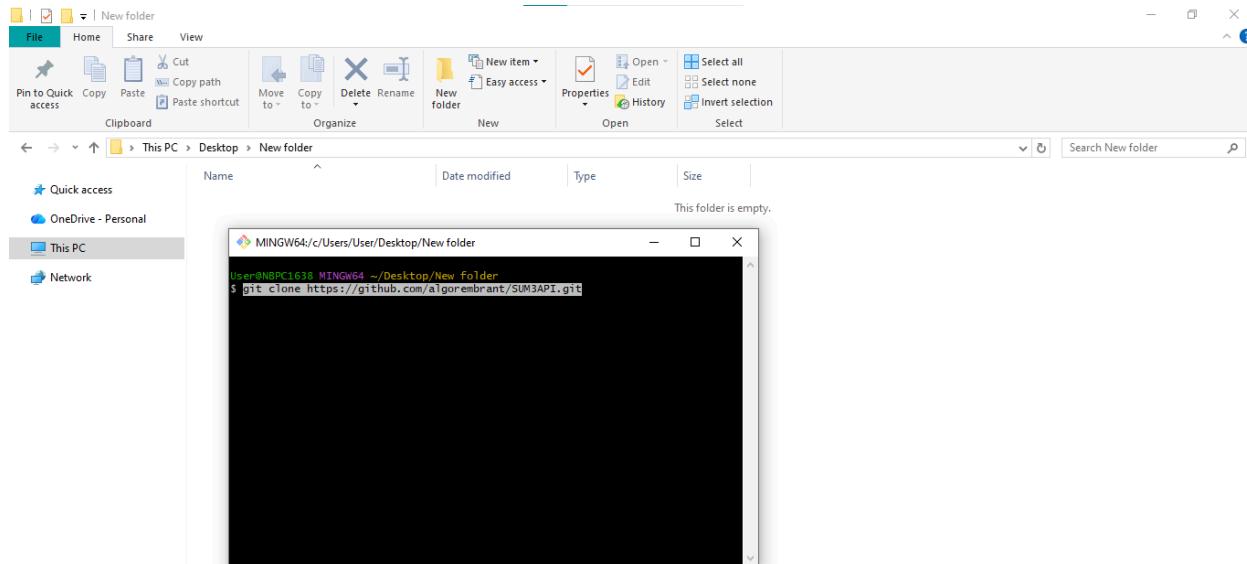
So you wanted to open the SUM3API software (trading terminal)? Thank you for being here. I will guide you as simply as possible.

Follow as we go,

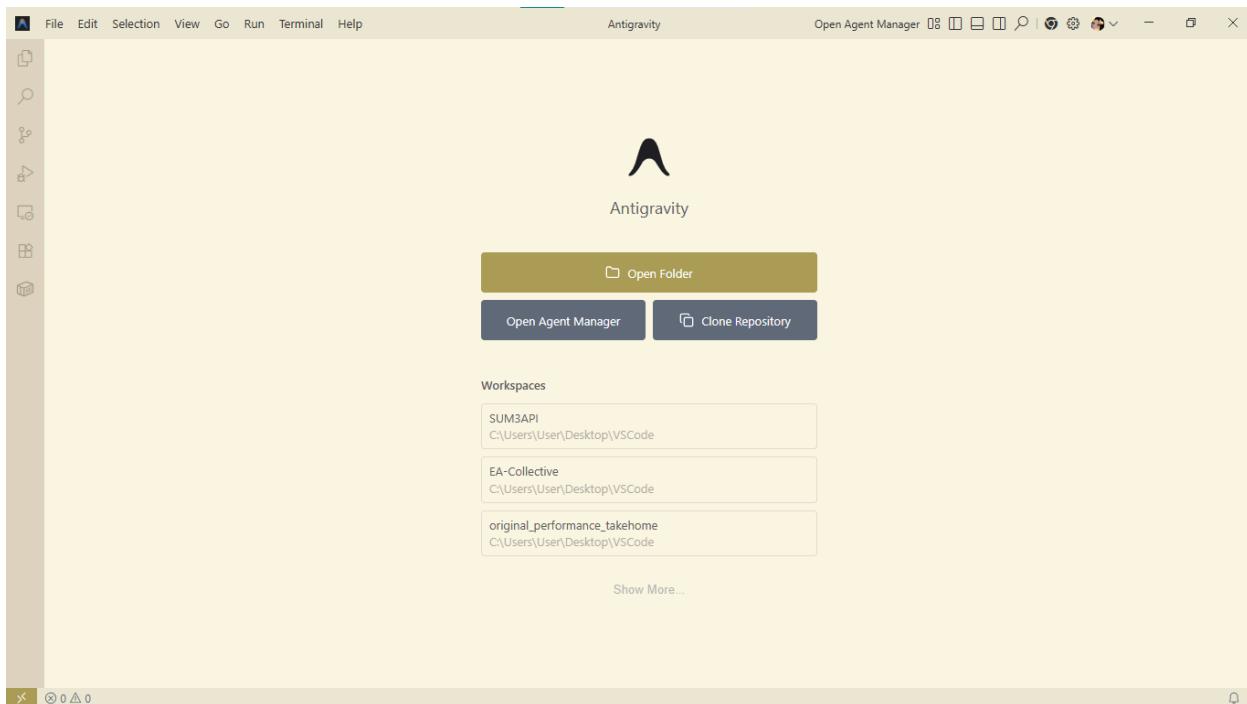
Select your preferred directory from your local computer. Right-click and press '*Open Git Bash here*' (it will show you the terminal with the designated path).



Clone my GitHub repository, bash: git clone <https://github.com/algorembrant/SUM3API.git>
(copy-n-paste that to the terminal)



Open your preferred code editor (in my case, I use Google's Antigravity)

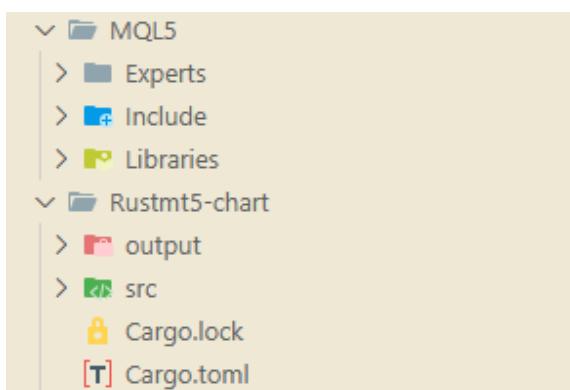


Select the 'SUM3API' Folder to edit (it's the cloned repo)

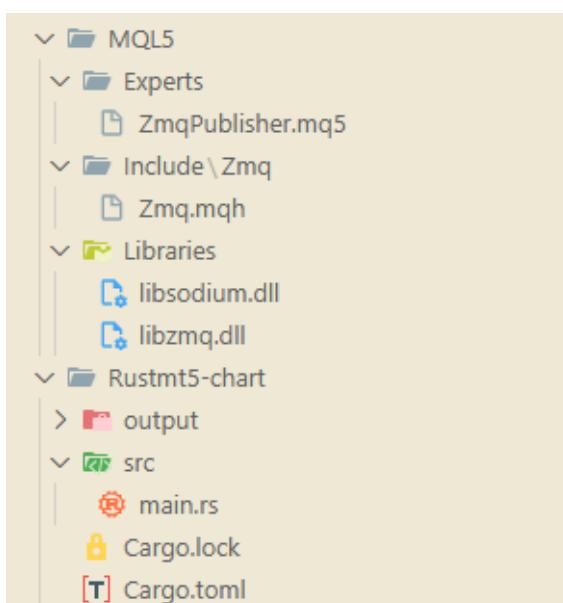
You will notice these 2 main folders for the [SUM3API: Using Rust, ZeroMQ, and MetaQuotes Language \(MQL5\) API Combination to Extract, Communicate, and Externally Project Financial Data from MetaTrader 5 \(MT5\)](#) project. Yes, you only need those two for codes and software (trading terminal).



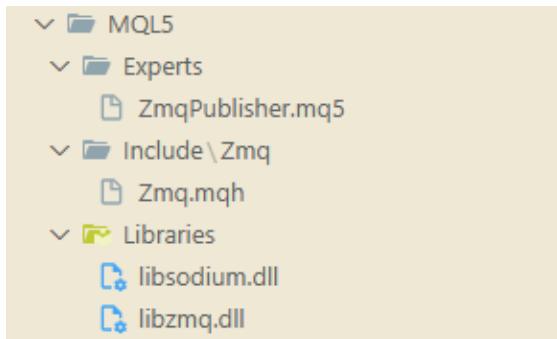
If you open it, it will look like this (btw, I use 'Material Icon Theme' Extension for folder/file icons)



And to look for all necessary files, just open those folders. Both [libsodium.dll](#) file and [libzmq.dll](#) file are downloadable on the internet.



We need to relocate the MQL5 Folder's files directory. Currently, we have it inside our repository. Imagine how the file path would look because it's really crucial for the proceeding steps. (eg., **MQL5\Experts\ZmqPublisher.mq5**)



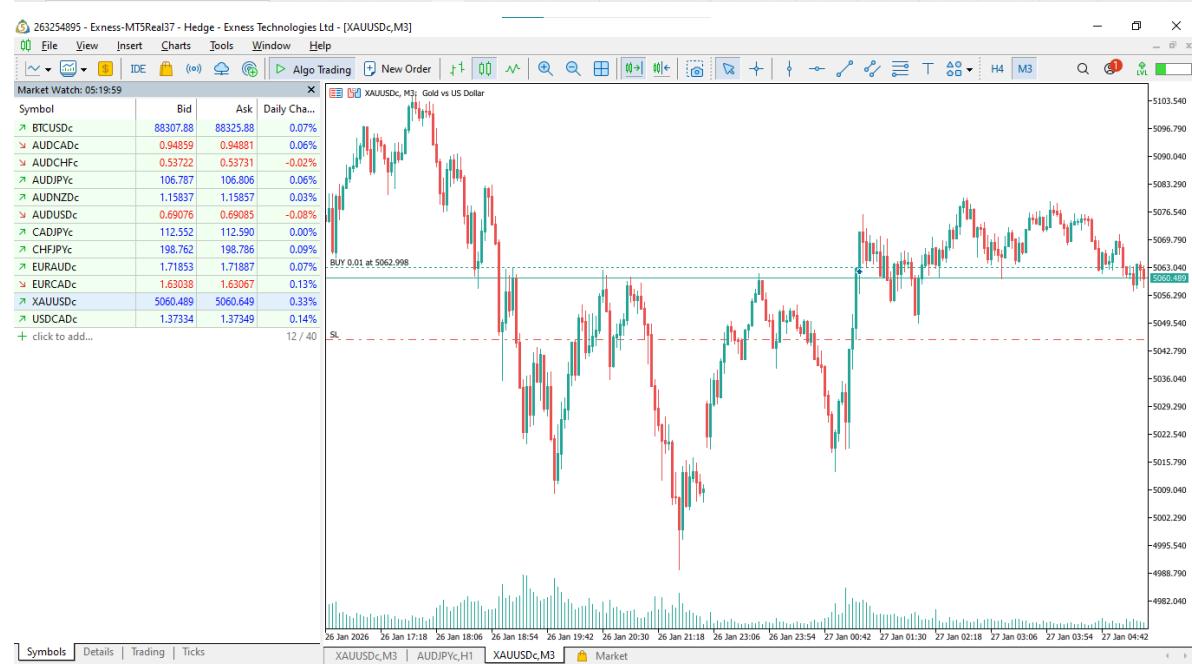
Now we need these 2 apps, MetaTrader5 and MetaEditor5.



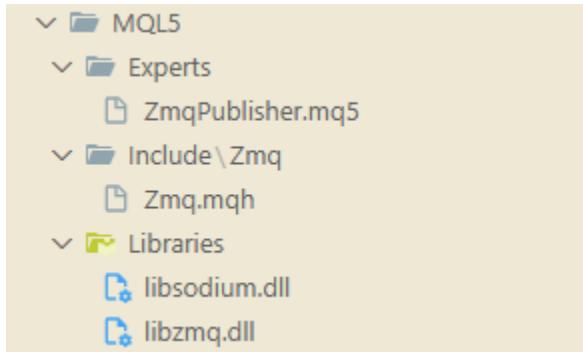
Brace yourself, I will try my best to be as simple and comprehensible as possible.

Open MetaTrader5 and log in to your *trading account* on your preferred Exchange/Broker. In my case, I'm using **Exness** and a **Standard-cent live trading account**. Since I'm using a cent-based account, all tradable assets will now have a 'c' suffix (eg., XAUUSDC for gold, and BTCUSDC for Bitcoin, etc.).

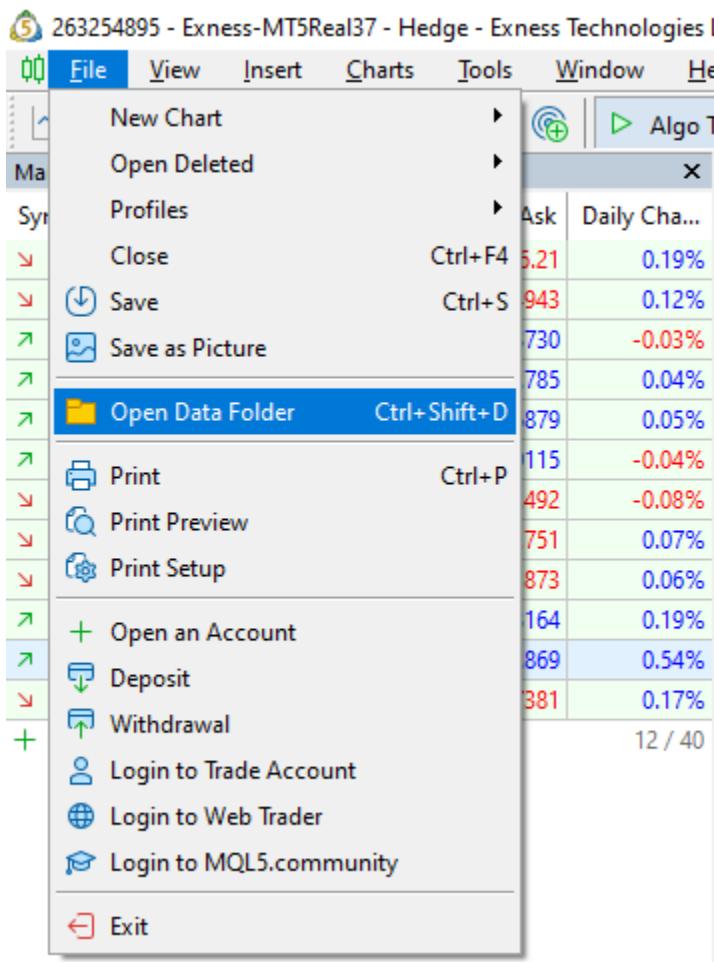
Inside MT5, open '**Market Watch**' (ctrl+m). Right-click 'XAUUSDC' (I'm using gold-forex for this implementation demonstration) and select '**Chart Window**'. It will look like this (don't mind my open test-order, I turned off my grid, you could change the chart background, not black too)



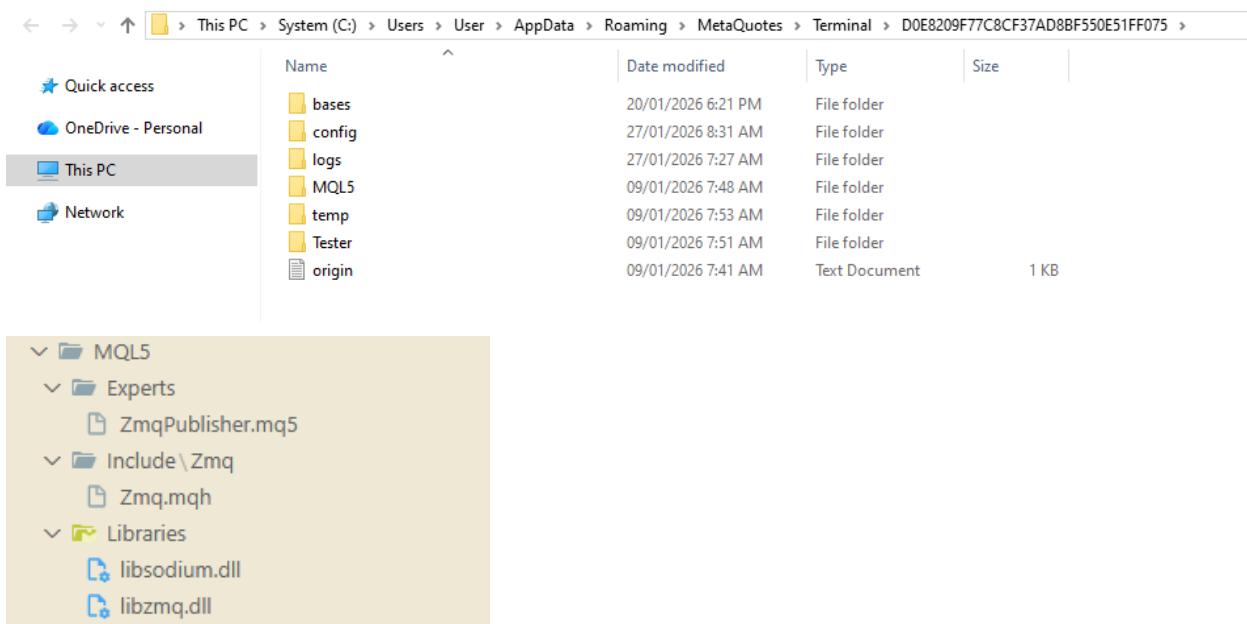
Now, remember these files from our repository? We need to relocate that to the legitimate MT5 folder.



Open ‘Open Data Folder’ (ctrl+shift+d).



In my case, I will see these. Notice we also have an MQL5 Folder (That is the legitimate path, unlike inside our repository).



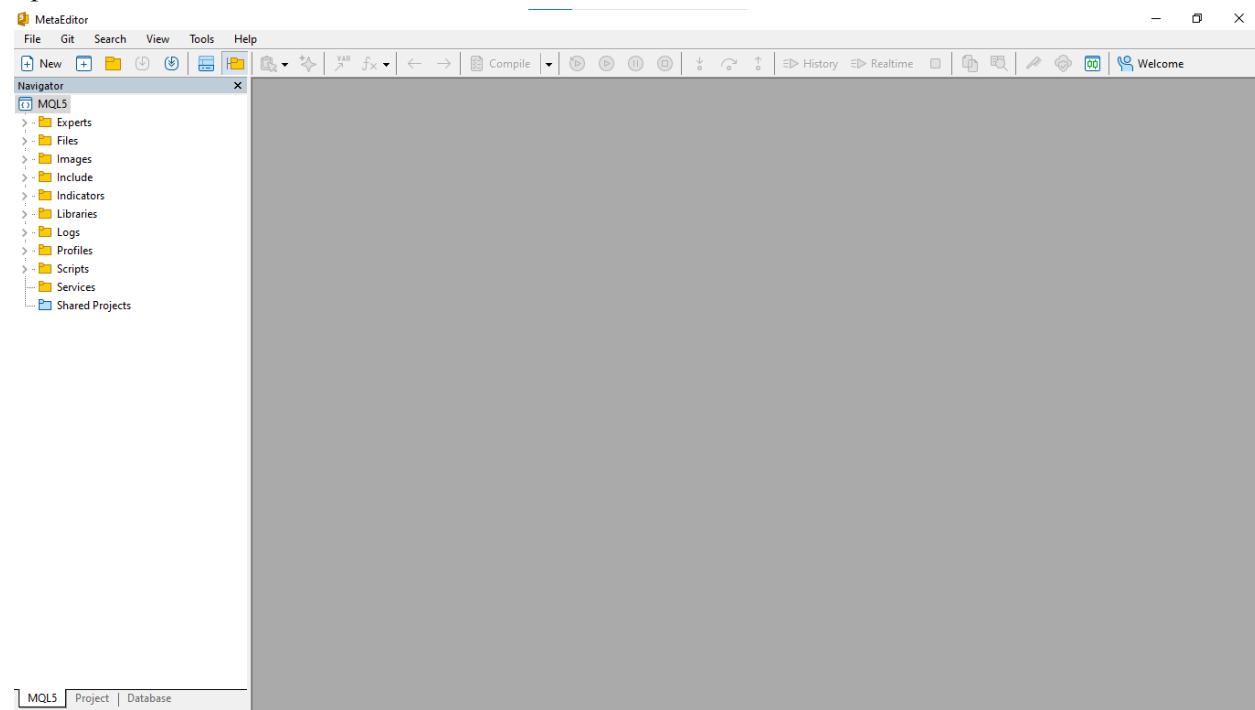
Relocate every single file from our repository to a similar path inside the MT5’s MQL5 legit path directory.

For;

ZmqPublisher.mq5 file: make sure its at \MQL5\Experts\ZmqPublisher.mq5 path

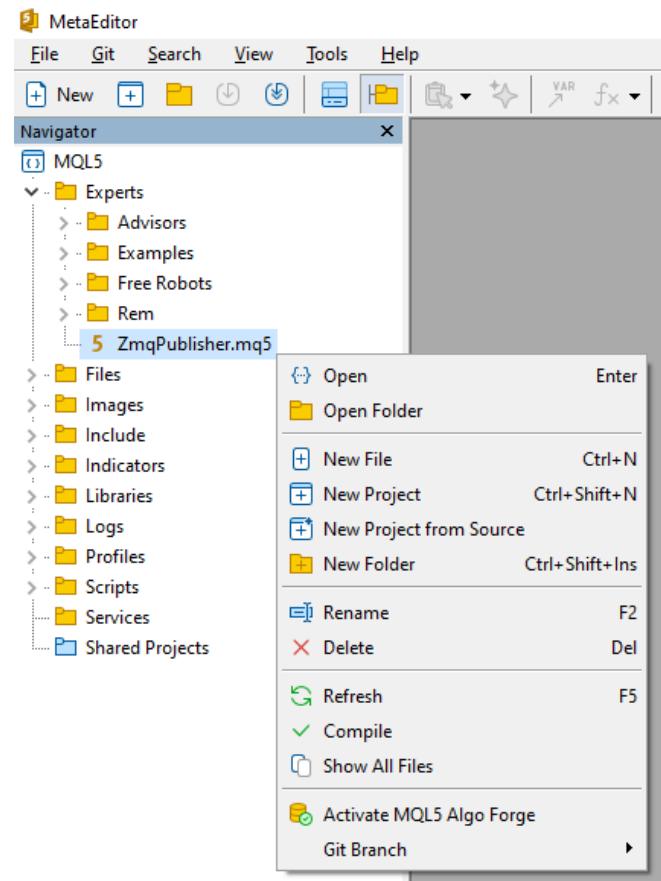
Zmq.mqh file: make sure its at \MQL5\Include\Zmq\Zmq.mqh path
libsodium.dll file: make sure its at \MQL5\Libraries\libsodium.dll path
libzmq.dll file: make sure its at \MQL5\Libraries\libzmq.dll path

Open MetaEditor5



Go to Experts → ZmqPublisher.mq5 → Open (if you wanna see the source code) → press compile on the top, (press F7). The source code is useful, especially if you are using a different Broker. I use an Exness Standard-cent live account, and my ticker/symbols are written with a ‘c’ suffix (eg, I use XAUUSDc for demonstration). Just change the code-line’s symbol if you want. Or

Go to Experts → ZmqPublisher.mq5 → Compile

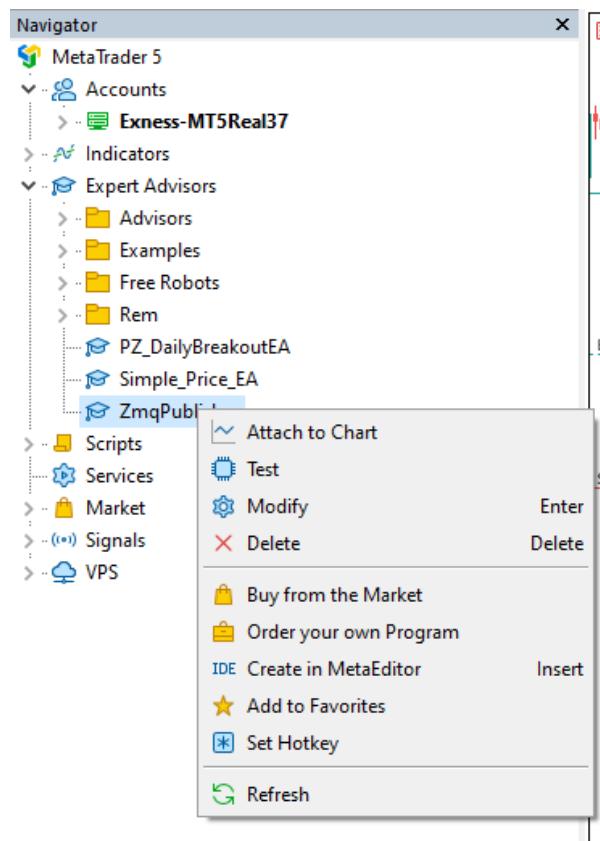


Go back to MetaTrader5 (MT5). Don't mind my test-open position

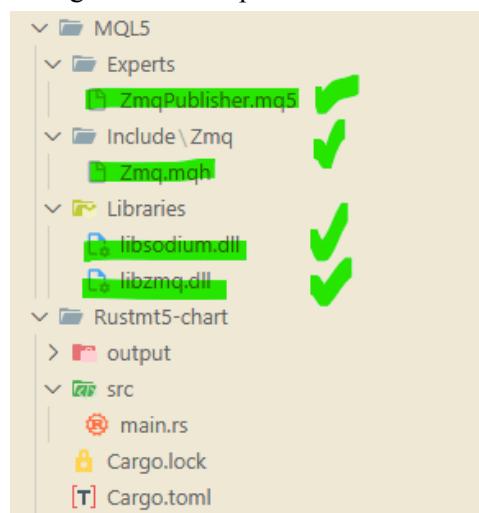


Go to ‘Navigator’ (ctrl+n) → Expert Advisors → right-click ZmqPublisher.mq5

Press ‘Attach to Chart’ to open or drag ZmqPublisher.mq5 to the main-content area. The EA will work whatever timeframe the chart has. In my case, I run it on XAUUSDc 3-minute timeframe (M3).



With that, we are finished with relocating the contents inside the MQL5 Folder from our repository into the legitimate MT5 path.



We don't have to do anything for the **Rustmt5-chart** Folder, no relocation. We simply need to run commands inside a terminal.

For this, I recommend using your raw-terminal. In my case, i cant run Rust commands on the Antigravity-terminal because it won't bypass my Windows PC security/antivirus system protection.

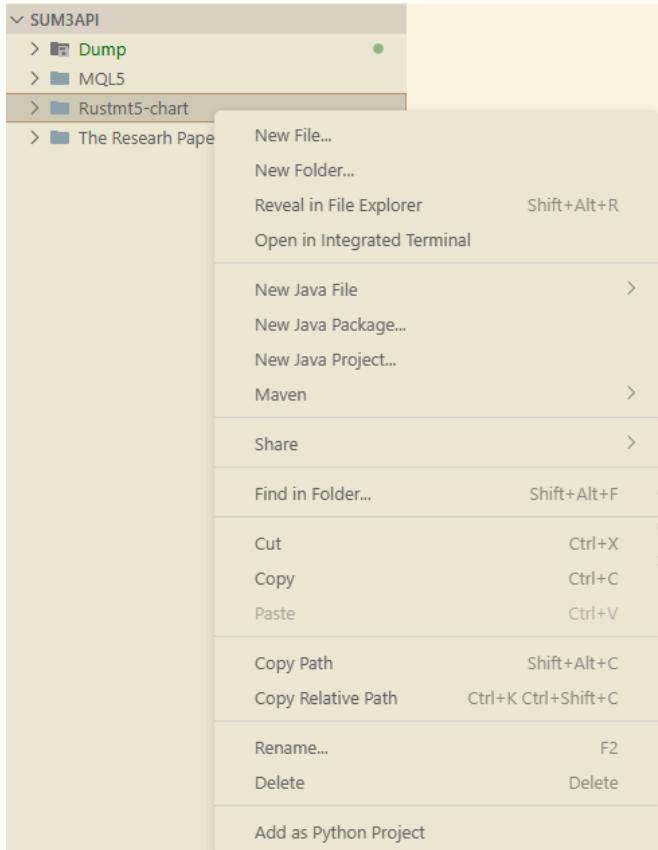
Open a raw-terminal (search/type '**cmd**' on Windows start/settings)



```
Command Prompt
Microsoft Windows [Version 10.0.19045.6456]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>
```

Find the path of our **Rustmt5-chart** Folder by simply → go to code-editor (Antigravity in my case) → right click → select/press '**Copy Path**' (shift+alt+c)

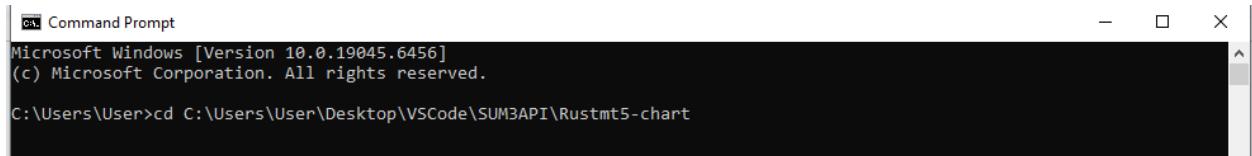


In my case, I will get:

C:\Users\User\Desktop\VSCode\SUM3API\Rustmt5-chart

Go back to the raw-terminal and bash:

cd C:\Users\User\Desktop\VSCode\SUM3API\Rustmt5-chart (simply type '**cd**' and paste)



```
Command Prompt
Microsoft Windows [Version 10.0.19045.6456]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>cd C:\Users\User\Desktop\VSCode\SUM3API\Rustmt5-chart
```

bash

cargo clean (cleaning first if there are rust files before mine)

In my case, I have to close my Antigravity if i will run *cargo build -release*

bash

cargo build -release

(that's a double dash '-' btw. Like `- -`). It will create a **target folder** for file storage after downloading/compiling what is written on out '**Cargo.toml**' file

- [1] cd C:\Users\User\Desktop\VSCode\SUM3API\Rustmt5-chart (path directory)
- [2] cargo clean (for cleaning previous rust files)
- [3] cargo build --release (download all Rust dependencies)

```
Command Prompt - cargo build --release
Microsoft Windows [Version 10.0.19045.6456]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>cd C:\Users\User\Desktop\VSCode\SUM3API\Rustmt5-chart
C:\Users\User\Desktop\VSCode\SUM3API\Rustmt5-chart>cargo clean
    Removed 35 files, 7.3MiB total

C:\Users\User\Desktop\VSCode\SUM3API\Rustmt5-chart>cargo build --release
Compiling proc-macro2 v1.0.105
Compiling quote v1.0.43
Compiling unicode-ident v1.0.22
Compiling cfg-if v1.0.4
Compiling windows-link v0.2.1
Compiling smallvec v1.15.1
Compiling stable_deref_trait v1.2.1
Compiling once_cell v1.21.3
Building [                                ] 6/217: smallvec, once_cell, stable_deref_trait, proc-macro2(build)
```

Wait for it to fully download; **it will take time**. And make sure your Code editor platform, like Antigravity, is closed.

Upon **Finished** You will see something like this: we are one bash away from opening the SUM3API software (trading terminal)

```
Command Prompt
Compiling color_quant v1.1.0
Compiling byteorder v1.5.0
Compiling web-time v0.2.4
Compiling serde v1.0.228
Compiling egui-winit v0.27.2
Compiling egui_glow v0.27.2
Compiling image v0.24.9
Compiling tokio-util v0.7.18
Compiling document-features v0.2.12
Compiling dashmap v5.5.3
Compiling futures v0.3.31
Compiling crossbeam-queue v0.3.12
Compiling uuid v1.19.0
Compiling regex v1.12.2
Compiling rand v0.8.5
Compiling asynchronous-codec v0.7.0
Compiling serde_derive v1.0.228
Compiling async-trait v0.1.89
Compiling itoa v1.0.17
Compiling zeromq v0.5.0-pre
Compiling eframe v0.27.2
Compiling egui_plot v0.27.2
Compiling chrono v0.4.43
Compiling mt5-chart v0.1.0 (C:\Users\User\Desktop\VSCode\SUM3API\Rustmt5-chart)
    Finished `release` profile [optimized] target(s) in 7m 30s

C:\Users\User\Desktop\VSCode\SUM3API\Rustmt5-chart>
```

Final bash

cargo run --release (to open)

Before running, make sure you've already done this

1. MT5 platform is running
2. The MQL5 EA is running, you've attached the **ZmqPublisher.mq5** on the chart. (in my case, I've attached in on XAUUSDC M3 chart). You may use any symbol chart; it doesn't matter, you don't need to change any code.
3. All of that before the bash: **cargo run --release**

And you'll see something like this. ENJOY!

