Risk Warning

Risk Warning and Usage Disclaimer:

Before proceeding with the installation and utilization of the trade automation script provided in this quick start guide, it is imperative to acknowledge the inherent risks associated with cryptocurrency trading. Cryptocurrency markets are highly volatile and unpredictable, subject to rapid price fluctuations and market manipulations. As such, engaging in cryptocurrency trading carries a significant risk of financial loss, and there is no guarantee of profit.

It is essential to understand that the trade automation script is a tool designed to assist in executing trades based on predefined parameters. However, it does not guarantee success or immunity from losses. The performance of the script is influenced by various factors, including market conditions, technological glitches, and user-defined settings.

Furthermore, it is crucial to note that the trade automation script provided in this guide is specifically tailored for use within the Crypto Andrew group environment. Access to both the free and Premium channels of Crypto Andrew is required for the script to function effectively.

By choosing to utilize the trade automation script, you acknowledge and accept the risks involved in cryptocurrency trading. You agree that any investment decisions made using the script are solely your responsibility, and you are aware of the potential financial consequences. It is highly recommended to exercise caution, conduct thorough research, and seek advice from financial professionals before engaging in cryptocurrency trading.

USE AT YOUR OWN RISK

Remember, past performance is not indicative of future results. No trading strategy can guarantee profits, and losses can exceed deposits.

Setup for Running Script

1. **Install Python 3.11**:

- Download and install Python 3.11 from the official Python website (https://www.python.org/downloads/).

- Follow the installation instructions for your operating system.

- Make sure to select the option to add Python to your system PATH during installation.

2. **Install Dependencies**:

- Once Python is installed, open a terminal or command prompt.

- Use pip, the Python package manager, to install the required dependencies:

```
pip install ccxt requests telethon
```

 - This command installs the ccxt, requests, and Telethon packages, which are used by the application.

3. **Configure API Keys**:

 - Edit the `config.json` file with the following structure:

~~~
   ```json
   {
     "exchange": {
       "name": "binanceusdm",
       "apiKey": "YOUR_API_KEY",
       "secret": "YOUR_API_SECRET",
       "enableRateLimit": true,
       "options": {
         "defaultType": "future"
       }
     },
     "telegram": {
       "appid": "YOUR_TELEGRAM_APP_ID",
       "apihash": "YOUR_TELEGRAM_API_HASH",
       "apiname": "AutoAndrew",
       "channel_ID": YOUR_CHANNEL_ID,
       "topic_ID": "2"
     },
     "tradeSettings": {
       "default_leverage": 20,
       "override_leverage": true,
       "leverage_override": 10,
       "leverage_on_spot_order": 5,
       "margin_mode": "cross",
       "risk_persentage": 1,
       "max_cost_limit": 10,
       "use_fixed_cost": true,
~~~

```
            "fixed_cost": 10,
            "keep_balance": 50
        }
    }
```

- Replace `YOUR_API_KEY`, `YOUR_API_SECRET`, `YOUR_TELEGRAM_APP_ID`, `YOUR_TELEGRAM_API_HASH’ with your actual API keys and Telegram credentials.

- Save the `config.json` file in the same directory as your Python script (`main.py`).

4. **Run the Script**:

- Open a terminal or command prompt.

- Navigate to the directory containing your Python script (`main.py`) and the `config.json` file.

- Run the Python script using the following command:

```
python main.py
```

- This command will execute your Python script, which will interact with the configured APIs based on the provided settings in the `config.json` file.

This guide provides a step-by-step process for setting up and running the Python script with the required dependencies and API keys configured.

Setup for Telegram API

1. **Create a Telegram Application**:

- Go to the Telegram website (https://telegram.org/) and log in with your phone number.

- Once logged in, go to the Telegram API development tools page (https://my.telegram.org/auth) to create a new application.

- Enter the required information, such as the name of your application, description, and website (optional).

- After submitting the form, you will receive an API ID and API hash. Keep these credentials secure, as they are required for accessing the Telegram API.

2. **Retrieve API ID and API Hash**:

- Log in to the Telegram website (https://telegram.org/) using your phone number.

- Go to the Telegram API development tools page (https://my.telegram.org/auth).

- If prompted, enter your phone number and verify it with the code sent to your device.

- Once logged in, you will see your list of applications. If you haven't created one yet, click on "Create Application" to create a new one.

- After creating the application, you will see your API ID and API hash listed under the "App configuration" section. These are the credentials you need to access the Telegram API.

3. **Use API ID and API Hash**:

- Copy the API ID and API hash provided on the Telegram website.

- Paste these credentials into your `config.json` file under the `"telegram"` section:

```json
"telegram": {
    "appid": "YOUR_TELEGRAM_APP_ID",
    "apihash": "YOUR_TELEGRAM_API_HASH",
    ...
}
```

- Replace `"YOUR_TELEGRAM_APP_ID"` with your API ID and `"YOUR_TELEGRAM_API_HASH"` with your API hash.

By following these steps, you can obtain your Telegram API ID and API hash, which are required for accessing the Telegram API in your Python script.

Setup for Binance API

1. **Create a Binance Account**:

- Go to the Binance website (https://accounts.binance.info/register?ref=28978810) and sign up for an account if you haven't already.

2. **Enable Two-Factor Authentication (2FA)**:

- After creating an account, enable Two-Factor Authentication (2FA) for added security. You can enable 2FA in your account settings.

3. **Generate API Key**:

- Once logged in, go to your account settings and navigate to the API Management section.

- Click on "Create New API Key" or a similar option to generate a new API key.

- You may be required to verify your identity and enable 2FA before generating an API key.

- Provide a label for your API key (e.g., "Trading Bot") to help identify its purpose.

- After generating the API key, you will receive an API key and a Secret key. These credentials are used to authenticate your requests to the Binance API.

4. **Set Permissions**:

- When generating the API key, you may need to set permissions for different types of operations (e.g., trading, withdrawal). Ensure that you only grant permissions necessary for your application.

5. **Copy API Key and Secret Key**:

- Copy the API key and Secret key provided by Binance. These keys are essential for accessing the Binance API.

6. **Use API Key and Secret Key**:

- Paste the API key and Secret key into your `config.json` file under the `"exchange"` section:

  ```json
  "exchange": {
    "name": "binanceusdm",
    "apiKey": "YOUR_API_KEY",
    "secret": "YOUR_API_SECRET",
    ...
  }
  ```

- Replace `"YOUR_API_KEY"` with your API key and `"YOUR_API_SECRET"` with your Secret key.

By following these steps, you can obtain API keys for Binance, which are required for accessing the Binance API in your Python script. Make sure to keep your API keys secure and never share them publicly.