Cryptrizx-Binance-Bot

REPO LINK - https://github.com/Kratugautam99/Cryptrizx-Binance-Bot-Project

Name of Author – Kratu Gautam

Date – 24th September 2025

Project Overview

Cryptrizx-Binance-Bot is a comprehensive Python-based trading bot for the Binance USDT-M Futures Testnet, offering both CLI (Command-Line Interface) and GUI (Streamlit Web App) modes. This robust tool enables traders to automate strategies ranging from simple market/limit orders to advanced algorithmic executions like OCO, Grid, Stop-Limit, and TWAP.

Live Demo (GUI Mode): https://kratugautam-cryptrizx-binance-bot-project.streamlit.app

---

Table of Contents

System Architecture

Modular Design

The source code is logically separated into a modular structure with advanced strategies organized into sub-packages. This separation ensures each file has a single responsibility, with client.py serving as the central engine for API communication and logging, while individual order scripts handle specific trading strategies.

Centralized API Client

The BinanceBotClient class provides secure authentication by loading API keys from environment files, configures the python-binance client for Futures Testnet, and offers unified order placement methods with consistent error handling.

Interactive User Interface

The bot features a fully interactive prompt-based system that provides guided input, dynamic hints based on market conditions, and robust validation before executing API calls.

Structured Logging

A comprehensive logging system outputs to both console for immediate feedback and a log file (bot.log) for persistent, timestamped records of all trading activities.

Features

Core Trading Strategies

- Market Orders - Instant execution at current market price
- Limit Orders - Set desired entry/exit prices
- Grid Orders - Automated buying low and selling high in ranges
- OCO Orders - One-Cancels-Other for risk management
- Stop-Limit Orders - Combine stop loss with limit orders
- TWAP Orders - Time-Weighted Average Price execution

Dual Interface

- CLI Mode - Scriptable terminal interface
- GUI Mode - User-friendly web dashboard (Streamlit)

Security and Management

- Secure API key management via environment variables
- Comprehensive activity logging
- Modular and extensible architecture

Installation and Setup

Prerequisites

- Python 3.10 or higher
- Binance account with API keys (Futures enabled)
- Git installed on your system

1. Clone Repository

```
git clone https://github.com/Kratugautam99/Cryptrizx-Binance-Bot-Project.git
cd Cryptrizx-Binance-Bot-Project
```

2. Environment Setup

Using Conda:

```
conda create -n cryptrizx python=3.10
conda activate cryptrizx
```

Using Venv:

```
python -m venv venv
source venv/bin/activate    # Linux/Mac
venv\Scripts\activate       # Windows
```

3. Install Dependencies

```
pip install -r requirements.txt
```

4. Configure API Keys

Create/update .env file in root directory:

```
BINANCE_API_KEY=your_actual_api_key_here
BINANCE_API_SECRET=your_actual_api_secret_here
```

Important: Ensure “Enable Futures” permission is activated for your Binance API keys.

CLI Mode Usage

Execute trading strategies directly from terminal:

Basic Orders

```
python CLI_Mode/market_orders.py
python CLI_Mode/limit_orders.py
```

Advanced Strategies

```
python CLI_Mode/advance/grid_orders.py
python CLI_Mode/advance/oco_orders.py
python CLI_Mode/advance/stop_limit_orders.py
python CLI_Mode/advance/twap_orders.py
```

GUI Mode Usage

Local Deployment

```
streamlit run GUI_Mode/app.py
```

Cloud Deployment

Access the live version: https://kratugautam-cryptrizx-binance-bot-project.streamlit.app/

Feature Implementation

Market Orders

Immediate execution at best available price:

```
$ python CLI_Mode/market_orders.py
Enter symbol (e.g., BTCUSDT): BTCUSDT
Enter side (BUY or SELL): BUY
Enter quantity: 0.001
```

Limit Orders

Execute at specific price or better:

```
$ python CLI_Mode/limit_orders.py
Enter symbol: ETHUSDT
Enter side: SELL
Enter quantity: 0.02
Enter limit price: 3800
```

Stop-Limit Orders

Conditional order triggered at specified price:

```
$ python CLI_Mode/advance/stop_limit_orders.py
Enter trigger/stop price: 67000
Enter limit price: 67100
```

OCO Orders

Take-profit and stop-loss pair (One-Cancels-Other):

```
$ python CLI_Mode/advance/oco_orders.py
Enter Take Profit price: 68000
Enter Stop Loss price: 62000
```

TWAP Orders

Time-weighted execution to minimize market impact:

```
$ python CLI_Mode/advance/twap_orders.py
Enter total quantity: 0.1
Enter duration (minutes): 30
Enter number of orders: 10
```

Grid Orders

Automated range trading strategy:

```
$ python CLI_Mode/advance/grid_orders.py
Enter price range: 60000-70000
Enter grid lines: 5
Enter quantity per order: 0.005
```

Project Structure

```
Cryptrizx-Binance-Bot/
│
├── CLI_Mode/
│   ├── client.py
│   ├── market_orders.py
│   ├── limit_orders.py
│   └── advance/
│       ├── grid_orders.py
│       ├── oco_orders.py
│       ├── stop_limit_orders.py
│       └── twap_orders.py
│
├── GUI_Mode/
│   └── app.py
│
├── images/
│   ├── bg.png
│   └── icon.png
│
├── .env
├── bot.log
└── requirements.txt
```

Logs and Monitoring

All trading activities are logged for monitoring and debugging:

```
tail -f bot.log
```

Example log output:

```
2024-01-15 10:30:45,123 - INFO - Market order placed: BUY 0.001 BTCUSDT
2024-01-15 10:31:22,456 - INFO - Limit order executed: SELL 0.02 ETHUSDT @
3800
```

Challenges and Solutions

API Authentication (Error 401)

Initial authentication failures due to incorrect API setup were resolved by ensuring Testnet keys with “Enable Futures” permission are used.

Trading Logic Errors (Error -2021)

“Order would immediately trigger” errors were addressed through interactive prompts with contextual market price hints.

Import Errors

ModuleNotFoundError issues with sub-packages were resolved by using the python -m flag for proper module execution.

API Parameter Issues (Error -1102, -400)

Quantity precision and string format requirements were handled through data validation and type conversion before API calls.

Tech Stack

- Python 3.10+ - Core programming language
- python-binance - Binance API integration
- Streamlit - Web application framework
- python-dotenv - Secure configuration management

Conclusion

The Cryptrizx-Binance-Bot project successfully delivers a robust, secure, and user-friendly trading automation tool that serves as a solid foundation for algorithmic trading strategy development. The implementation of six distinct order types, combined with dual interface support (CLI and GUI), provides flexibility for various user preferences and trading scenarios.

The interactive design approach not only ensures functional correctness but also educates users on proper trading logic, making the tool suitable for both beginners and experienced traders. The modular architecture allows for easy extension with additional trading strategies and features.

Future enhancements could include graphical user interface improvements, long-term state management for grid trading strategies, integration of technical indicators for automated decision-making, backtesting capabilities, and multi-exchange support. The project demonstrates effective problem-solving through its handling of API integration challenges and user experience considerations.

License

Cryptrizx-Binance-Bot
Automate with confidence, trade with precision
Built by Kratu Gautam