■ TWS Orders Placement via TradingView Webhooks

Short Description

Automated order placement system that connects **TradingView alerts** with **Interactive Brokers TWS/IB Gateway**, enabling fully automated trading through webhooks.

■ Overview

This project allows traders to automatically execute **Buy to Open**, **Sell to Close**, **Sell to Open**, and **Buy to Close** orders on Interactive Brokers using alerts generated from **TradingView**.

It provides a **secure webhook listener** that translates TradingView signals into executable TWS API commands.

■■ Features

- ■ **Webhook Integration** with TradingView (JSON-based alert system)

- ■ **Interactive Brokers API (TWS/IB Gateway)** connectivity

- ■ **Secure request validation** with secret key

- ■ **Order logging & response tracking**

- ■ **Supports multiple order types** (MKT, LMT, etc.)

- ■ **Error handling & auto-retry mechanism**

- ■ **Lightweight & fast Flask/FastAPI backend**

■ Advanced Capabilities

- **Modular design** – easily extend to add new exchanges or strategy logic

- **Position management** – automatic close/open logic based on alert signal

- **Paper/live mode toggle** for testing safely

- **Customizable JSON alert schema** (works with any TradingView indicator)

- **Optional email/Telegram notifications** for trade confirmations

---

■ Tech Stack

- **Python 3.10+**

- **Flask or FastAPI**

- **Interactive Brokers API (ib_insync / TWS)**

- **TradingView Webhook**

- **SQLite / JSON for logs**

■ Folder Structure

```
```

TWS-orders-placement-via-TradingView-webhooks/

■

■■■ main.py # Main webhook server file

■■■ config.json # Configuration (API keys, ports, etc.)

■■■ requirements.txt # Python dependencies

■■■ logs/ # Order & execution logs

■■■ README.md # Project documentation

■ How It Works

1. **TradingView Alert → Webhook JSON**

You configure alerts in TradingView using your strategy and send them to your webhook endpoint.

2. **Webhook Receiver → Parse Signal**

The system receives and validates the signal (checks secret key, timestamp, etc.).

3. **Order Placement → Interactive Brokers**

The bot places an order via the TWS API based on the signal type (BTO, STC, etc.).

4. **Logging & Confirmation**

The trade result and TWS response are logged locally and optionally sent as notifications.

■ Setup Instructions

1. Clone the repo:

```bash
git clone https://github.com/yourusername/TWS-orders-placement-via-TradingView-webhooks.git
```

2. Install dependencies:

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

3. Configure `config.json` with:

- Your webhook secret key

- IBKR connection details (host, port, client ID)

4. Run:

```bash
python main.py
```

5. In TradingView alert message (example):

```json
{
"secret": "YOUR-SECRET",
"signal": "BTO",
"symbol": "{{ticker}}",
"qty": 10,
"ordertype": "MKT",
"price": {{close}}
}
```

■ Example Trading Flow

Signal	Action	Description
BTO	Buy to Open	Opens a new long position
STC	Sell to Close	Closes existing long
STO	Sell to Open	Opens a short position
BTC	Buy to Close	Closes existing short

■■ Security

- Requests validated with a **shared secret key**
- IP allow-list available for TradingView webhook source
- No credentials stored in plaintext

■ Future Enhancements

- Multi-account management
- Web dashboard for monitoring
- Integrate with Binance/MetaTrader 5
- Cloud deployment template (AWS Lambda, Railway)

■■■ Author

Kamran Ali (SaifyTech Ltd.)

■ Trading Automation & API Developer

■ kamranexpert786@gmail.com