Project Title:

Automated Binance Futures Trading Bot

Abstract:

This project presents an automated trading bot for Binance USDT-Margined Futures using Python. It supports multiple order types including Market, Limit, Stop-Limit, One-Cancels-the-Other (OCO), and Time-Weighted Average Price (TWAP) orders. The system provides a robust Command Line Interface (CLI) for interactive trading, implements detailed logging for debugging and auditing, and includes strong input validation. This bot is designed for safe and flexible trade execution on the Binance Testnet, facilitating learning, testing, and automation of futures trading strategies.

Objective:

- Enable efficient and automated order placement in Binance Futures using Python.
- Provide support for all major order types: Market, Limit, Stop-Limit, OCO, and TWAP.
- Ensure clear user interaction with input validation and order confirmations.
- Maintain a structured log of all trading activities for transparency and debugging.

Architecture Overview:

- **Frontend**: Command Line Interface (CLI)
- **Backend**: Python script utilizing the official Binance API
- **Logging**: Custom logger using Python's logging module
- **Environment**: Binance Futures Testnet
- **Data Flow**:
  - User inputs symbol, order type, side, quantity, etc.
  - Real-time price is fetched from Binance.
  - Order parameters are validated and formatted.
  - Order is placed via the Binance API.
  - Logs are created and shown to the user.

Key Features:

1. **Interactive Order Placement**
   - Prompts for symbol, order type, buy/sell side, and quantity.
   - Supports all major order types with tailored prompts.

2. Order Type Implementations
   - **Market Order**: Instant execution at market price.
   - **Limit Order**: Executes at a specific target price.
   - **Stop-Limit Order**: Triggered when price crosses a stop value.
   - **OCO (One-Cancels-the-Other)**: Combines take-profit and stop-loss logic.
   - **TWAP (Time-Weighted Average Price)**: Splits orders into slices and places them periodically over time.

3. **Real-Time Price Fetching**

Uses Binance Futures API to fetch live prices for decision making and user prompts.

4. **Robust Logging**
   - Logs every action and error with timestamps.
   - Logs stored in a single file with section breaks for each session.

5. **Validation and Safety**
   - Input checks for symbols, prices, quantity.
   - Confirmation prompts before executing trades.
   - Price sanity checks to avoid erroneous large deviations.

6. **Modular Codebase**
   - Separated modules for each order type.
   - Easy to extend or maintain.

Usage Flow

1. User runs the bot and selects from menu options.
2. Enters trade details (symbol, order type, side, quantity, etc.).
3. Bot fetches live market price (if needed).
4. User confirms the summary.
5. Bot places the order and logs the result.
6. User can view balance, open orders, trade history, or cancel an order.

Sample Log Snippet

```
2025-07-10 11:08:31 | trading_bot | INFO    | Account balance: 15672.08717625 USDT | Available: 10509.20065811 USDT
2025-07-10 11:08:31 | trading_bot | INFO    | API connection validated
2025-07-10 11:08:31 | trading_bot | INFO    | Trading bot initialized successfully
2025-07-10 11:08:31 | trading_bot | INFO    | Trading interface initialized
```

Advanced Features

- **Real-Time Price Fetching**
  Fetches live price before executing advanced orders to assist decision-making.
- **Order Confirmation Summary**
  Shows a formatted trade summary before confirmation to avoid accidental execution.
- **Structured Logging**
  Logs all user actions and API responses with timestamps for traceability.
- **Error Handling & Recovery**
  Catch exceptions for network, API key, and logic failures with safe exit paths.
- **Account Operations**
  - View account balance
  - View open orders
  - View trade history
  - Cancel active orders

Benefits of the Bot

- Simulates real-world futures trading scenarios.
- Excellent tool for testing trading strategies on testnet.
- Teaches various order types and when to use them.
- Can be extended with risk management, portfolio tracking, or signal-based trading logic.

Possible Future Enhancements

- **Add Telegram/email notifications for trades.**
- **Implement advanced strategies using technical indicators.**
- **Add a web dashboard for monitoring.**
- **Integrate AI/ML to detect trade signals.**
- **Run on live Binance environment (with added safeguards).**

Conclusion

This trading bot offers a comprehensive and modular solution for understanding and simulating automated futures trading on Binance. With rich features, clear user interaction, and robust backend logic, it stands as a strong foundation for further development and experimentation in the domain of algorithmic trading.