**004 Trading System**



**Names**: Braulio Iván Marin Ortiz, Pablo Emilio Sánchez Galicia

**Subject:** Microestructura y sistemas de trading

**Professor:** Luis Felipe Gómez Estrada

**Delivery Date:** 10/04/2025

**Introduction**

The purpose of this project is to develop a cryptocurrency trading dashboard that allows users to monitor market prices, view account balances, and place test orders through a simple and intuitive interface. The platform connects to the Binance Testnet, enabling real time data interaction without the risk of using real funds. Designed as a practical application of both financial and technological knowledge, this project combines frontend technologies for user interaction with a backend that handles data processing and communication with external APIs. It serves as a learning tool to better understand how digital trading platforms operate, and how market data can be retrieved, displayed, and acted upon in a simulated environment.

**Frontend**

The frontend of this project is composed of two main files: index.html and script.js. Together, they create a responsive and interactive user interface that allows users to monitor cryptocurrency prices, view historical data, check account balances, and place market or test orders. The index.html file structures the layout using modern CSS, while script.js connects the interface to backend services through API calls, dynamically updating charts and account information in real time. This section demonstrates how financial data and trading functionalities can be integrated for building effective fintech tools.

* **Index.html** - The index.html file is the main structure of the frontend for our Trading Dashboard project. This dashboard provides users with a clear and interactive interface to monitor their cryptocurrency account, visualize real-time market data, and simulate or place trading orders. It provides a responsive and functional layout for a simulated cryptocurrency trading platform. Its main goal is to offer an intuitive interface for non-technical users while enabling the visualization of financial data and the execution of basic trading operations. This type of project showcases the practical intersection between finance and programming.
* **Script.js** - The script.js file is the JavaScript component that powers the interactivity and dynamic behavior of the trading dashboard. It connects the user interface (defined in index.html) with backend services via API calls and handles events such as fetching data, updating charts, submitting orders, and displaying real-time feedback.

**Backend**

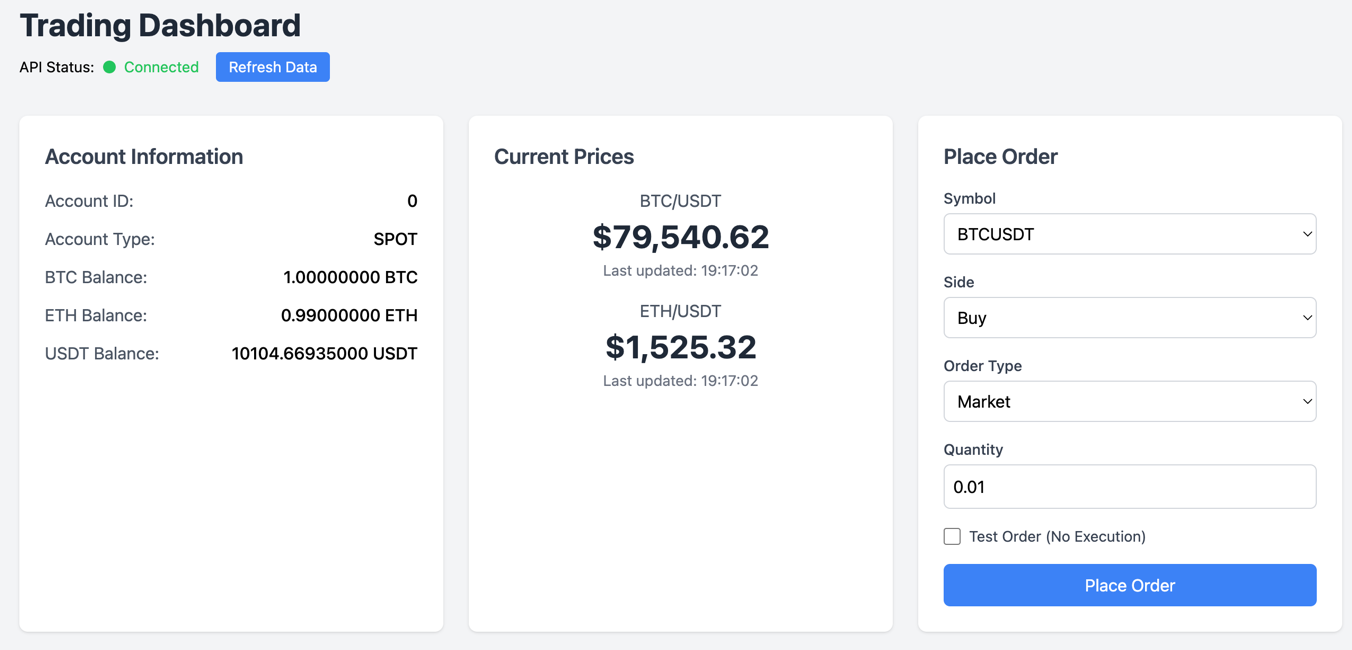
The backend of this project is built using FastAPI and is responsible for handling all the logic behind the trading dashboard. It connects to the Binance test environment to retrieve market data, manage account information, and simulate cryptocurrency orders. The backend is composed of several modules that organize the data, handle API requests, and ensure smooth communication between the dashboard and Binance. This allows the system to provide real-time updates and simulate trading actions in a safe and controlled environment.

* **Models.py** - The models.py file defines the data models used throughout the backend using Pydantic, a Python library that provides data validation and parsing. These models act as blueprints for the structure of the data exchanged between the backend and frontend or external services.

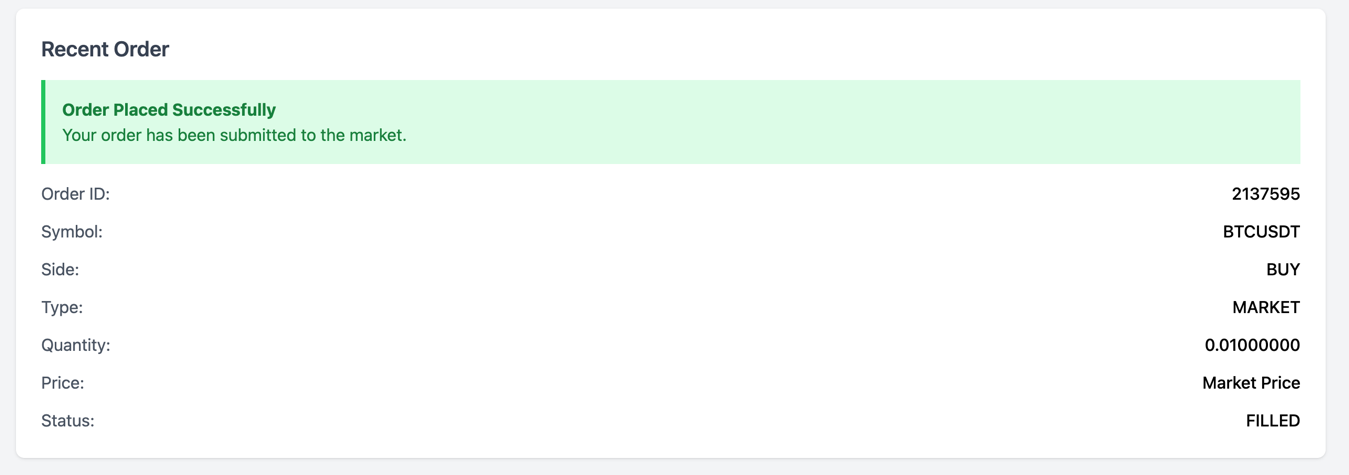
Each class in this file represents a specific type of financial data relevant to the system:

* + AssetBalance and AccountInfo model account information and cryptocurrency balances.
  + PriceInfo is used to store real-time price data of trading pairs like BTC/USDT.
  + CandleInfo models the structure of historical price candles, which are used for charting and technical analysis.
  + OrderRequest defines the expected input for placing a trade order.
  + Order and OrderFill capture the details and execution information of placed orders, closely matching the response format of APIs like Binance.
* **Client.py** - The client.py file is responsible for connecting the system to the Binance test environment. It allows the application to get important market data like the current price of cryptocurrencies and their historical behavior over time. This file makes it possible to send requests to Binance and receive updated information, which is later shown on the dashboard. By using this connection, the platform can simulate real trading scenarios without using real money.
* **Main.py** - The main.py file acts as the core of the backend system. It creates a set of services that allows the frontend (the dashboard) to communicate with the Binance test platform. Through this file, the system can check if everything is working properly, get account balances, see live and historical cryptocurrency prices, and place test orders. It uses FastAPI to create these services and makes sure they are accessible from the frontend. In simple terms, this file controls the main operations that power the dashboard’s functionality.

**Dashboard**

****

****

****

**Conclusion**

This project demonstrates the development of a functional and interactive cryptocurrency trading dashboard by integrating a friendly frontend with a responsive backend connected to the Binance Testnet. Through real-time data updates, price visualizations, and test order placement, the system provides a realistic simulation of a trading environment without financial risk. The use of FastAPI and JavaScript allows for efficient communication between the client and server, while the integration with Binance ensures access to real market behavior. Overall, this project reflects a practical application of financial technology concepts and showcases the ability to build and operate a basic trading platform, making it a valuable learning experience for anyone pursuing a career in finance or fintech.