**Crypto Trading Simulator API Documentation**

*Plamen Pavlov*

1. **Overview**

This backend API supports core functionalities for a cryptocurrency trading simulator. The main functionalities include:  
- User account creation and balance management  
- Live crypto price fetching via Kraken WebSocket  
- Buying/selling cryptocurrency  
- Tracking transactions and holdings

1. **API Endpoints**

**ApplicationController**

The purpose of this controller is to simply check if the backend server is operational.

*GET /api/test*

Response: Returns plain text message “Backend is working!”

**CryptoPriceController**

Purpose:

Provides access to live crypto price data for the top 20 cryptocurrencies, using Kraken WebSocket.

*GET /api/prices/top20*

Response: Returns a list of CryptoPrice objects representing real-time prices of the top 20 cryptocurrencies by market cap.

Design Notes:

- Uses @CrossOrigin(origins = "\*") to allow frontend access from any origin.

**TransactionController**

Manages user transaction history and calculates real-time holdings.

*GET /api/transactions/{userId}*

Returns all transactions for a specific/given user.

*GET /api/transactions/{userId}/holdings*

Calculates and returns the current holdings of the user, mapping symbols (e.g., BTC, ETH) to quantities.

Design Notes:

- Transactions include both buys and sells.

- Holdings are computed by summing buys and subtracting sells per crypto symbol.

**UserController**

This controller handles user creation, balance management, and buy/sell operations.

*POST /api/users/create?username=PlamenPavlov*

Creates a new user with the given username and initializes their balance - $10 000.

*GET /api/users/{id}*

Fetches user data (including balance) by user ID.

*POST /api/users/{id}/buy*

Buys cryptocurrency for a given user using current market price. Params: cryptoName, symbol, quantity

Validates: Live price is available, User has sufficient balance, Quantity is greater than 0.

*POST /api/users/{id}/sell*

Sells cryptocurrency for a given user using current market price. Params: cryptoName, symbol, quantity

Validates: User holds enough crypto, Quantity is valid, Live price is available.

*POST /api/users/{id}/reset*

Resets a user account: Sets balance to the default initial value - $10 000, clears all transactions and holdings for the specific user.

Design Notes:

- Uses live data via KrakenWebSocketClient

- Buy/sell actions update user balance, holdings and insert a transaction

- 'Reset' is useful for testing/simulation restarts

1. **Design decisions**

Live pricing

All trading operations use real-time prices from Kraken via WebSocket, ensuring up-to-date valuations.

Stateless API

No session management; users are identified by ID in requests.

Single responsibility per endpoint

Each controller and method is focused on one task (like buying, resetting, listing holdings), improving readability and testability.

CORS Enabled

@CrossOrigin allows any frontend (in this project a React app) to consume the price API without CORS errors.