



added documentation about the frontend

Mino Petrizzo authored 4 days ago

 **README.md** 5.65 KiB

Blockchain Project

Balancing

The balancing process works by keeping the tokens in the pool at a desired ratio (0.5 in our example). This is done by periodically checking the amount of both tokens held in the pool and initiating a swap if necessary to maintain the desired ratio. For the swap, a dedicated wallet with enough of both tokens is used. The checking and swapping process is repeated at a specified interval.

Algorithm

The balancing process works in a repeating manner as follows:

1. Determine the amount of Token A and Token B in the swap pool
2. Calculate the ratio of the two tokens
3. If the threshold is exceeded, rebalancing is required
 1. Rebalancing required: initiate a swap to balance the token
 2. Otherwise: proceed to next step
4. Restart from top after timeout

Wallet

A dedicated wallet is used for the balancing process. This wallet must always have enough of both tokens for the balancing process to be successful. This is a critical aspect of the solution.

Balancing wallet address: `0xb1Ca66b93Fb23e56dEC272154457659776864bF0` Fun fact: The wallet was created with GO.

Start balancing proccess

To start the rebalancing process, build and run the corresponding Docker container.

```
docker compose build backend_balancing
```

```
docker compose run backend_balancing
```

Frontend Documentation for Svelte Page

Overview

This Svelte page allows users to interact with a decentralized finance application to trade Zeus Tokens (ZT). It connects to MetaMask, displays real-time data, and provides charts for balance and price tracking. The page also supports light and dark modes.

Features

- **Connect MetaMask:** Users can connect their MetaMask wallet.
- **Real-Time Data:** Displays token balances and the current price of Zeus Tokens (ZT).
- **Buy/Sell Tokens:** Users can buy or sell ZT using USDC.
- **Transaction History:** Shows recent transactions with transaction hashes linked to Etherscan.
- **Dark/Light Mode:** Option to toggle themes.

Components

Variables

- `walletAddress` : Stores the connected wallet address.
- `isConnected` : Indicates if the wallet is connected.
- `isDarkMode` : Tracks the theme (dark or light).
- `amount` : The amount entered by the user for trading.
- `transactions` : List of recent transactions.
- `currentPrice` , `usdcBalance` , `ztBalance` : Real-time data for token price and balances.
- `priceHistory` , `balanceChart` , `priceChart` : For displaying balance and price charts.

- `onMount` : Initializes charts and starts periodic updates for price and balances.
- `updateData` : Fetches the latest pool data (USDC and ZT balances) and updates the price.
- `tradeToken` : Handles the logic for buying or selling tokens.
- `approveToken` : Approves token spending before a trade.
- `updateBalanceChart` / `initializeBalanceChart` : Update and initialize the balance chart.
- `initializeChart` / `updateChart` : Initializes and updates the price chart with new data.
- `getPoolPrice` : Fetches and calculates the current pool price.
- `connectWallet` : Connects the user's wallet via MetaMask.
- `toggleTheme` : Switches between dark and light modes.

UI Overview

- **Hero Section:**
 - Button to connect MetaMask and toggle for dark/light mode.
 - Introductory text and a call to action link.
- **Dashboard** (Visible when wallet is connected):
 - **Monitoring:** Displays current price and charts for balances and price over time.
 - **Trading:** Form for buying/selling tokens with computed amounts.
 - **Transaction History:** Table showing the latest transactions with links to Etherscan.

Charts

- **Balance Chart:** Displays USDC and ZT balances in a bar chart.
- **Price Chart:** Shows the price of ZT over time in a line chart.

Dependencies

- **Svelte:** For building reactive UI.
- **Ethers.js:** For interacting with the Ethereum blockchain.
- **Chart.js:** For rendering charts.
- **chartjs-adapter-date-fns:** For handling dates in the price chart.

amountOutMinimum

We calculate the desired output amount by multiplying the amountIn (the amount of input token) by the current pool price (currentPrice). We then apply a 1% slippage tolerance by multiplying the desired output amount by 0.99 (1 - 0.01). This ensures that the minimum output amount you're willing to accept is 99% of the expected output based on the current pool price.

sqrtPriceLimitX96

We convert the current pool price (currentPrice) to the X96 format used by Uniswap V3 by multiplying it by 2⁹⁶. We then take the square root of the current price in X96 format to get the sqrtPriceLimitX96. This sets the price limit to the current pool price, which will help maintain the 1:1 parity between ZT and USDC.

Conclusion

This Svelte page provides an easy-to-use interface for trading Zeus Tokens, displaying real-time data, and managing wallet connections. It supports both light and dark modes and allows users to monitor their transactions and token balances. The choice to use a single-page design for this application was made to provide a clean and seamless user experience. By consolidating all the necessary features — wallet connection, trading, charts, and transaction history — on one page, users can have everything in view without the need to navigate between multiple sections. This approach ensures that users can quickly access important information like token prices and balances, while also streamlining the trading process. Moreover, it reduces the complexity of navigation, making the interface more intuitive and user-friendly. Additionally, a single-page design keeps everything more responsive and fast, which is especially important for real-time financial data.