Automated Liquidity Addition and Token Purchase Script

Overview

This script automates the process of adding liquidity to a PancakeSwap pool and purchasing a specific token (TokenA) with BNB (Wrapped Binance Coin) on the Binance Smart Chain (BSC) testnet. It utilizes the ethers.js library for Ethereum development and interacts with BSC testnet contracts.

Prerequisites

Before running the script, ensure you have the following prerequisites:

Node.js installed on your machine.

Access to the BSC testnet network.

A testnet wallet with BNB and TokenA for liquidity and token purchase.

Setup

Install required dependencies:

ethers: For Ethereum wallet and contract interactions.

axios: For making HTTP requests.

bash

Copy code

npm install ethers axios

Replace the following placeholders in the script:

bscTestnetRpcUrl: BSC testnet RPC endpoint.

privateKey: Private key of your testnet wallet.

wBNBtoken: Address of WBNB (Wrapped Binance Coin) contract.

tokenAddress: Address of TokenA contract on the BSC testnet.

pancakeSwapRouterAddress: Address of the PancakeSwap router contract on the BSC testnet.

Adjust the amount and numOfPurchase values as needed for your liquidity addition and token purchase.

Script Execution

To run the script, execute the following command:

node .\automatePancakeSwap.js

Script Workflow

The script initializes Ethereum providers, wallets, and contract instances for interacting with the BSC testnet.

It approves the PancakeSwap Router to spend WBNB and TokenA on behalf of your wallet.

Liquidity Addition: The script attempts to add liquidity to the PancakeSwap pool for the WBNB/TokenA pair. It specifies the amount of WBNB and TokenA to provide for liquidity.

Token Purchase: After adding liquidity, the script calculates the total amount of TokenA to purchase and converts it to Wei (the smallest unit of Ether). It then purchases TokenA using the PancakeSwap Router, swapping BNB for TokenA.

The script continues to execute the liquidity addition and token purchase process in a loop as specified by main(). You can customize the loop as needed.

Error Handling

The script includes error handling to capture and log any errors that occur during execution. It will log errors related to contract interactions or transactions.

Note

Ensure you have sufficient BNB and TokenA in your testnet wallet for liquidity addition and token purchase.

Adjust gas limits, deadlines, and other parameters in the script as needed for your specific use case.

This documentation presentation provides an overview of the automated liquidity addition and token purchase script, including setup instructions, execution details, and error handling. Please replace placeholders and customize the script to suit your specific requirements before running it on the BSC testnet.