

Frontend for UniswapV2 Rewrite

Links for this project

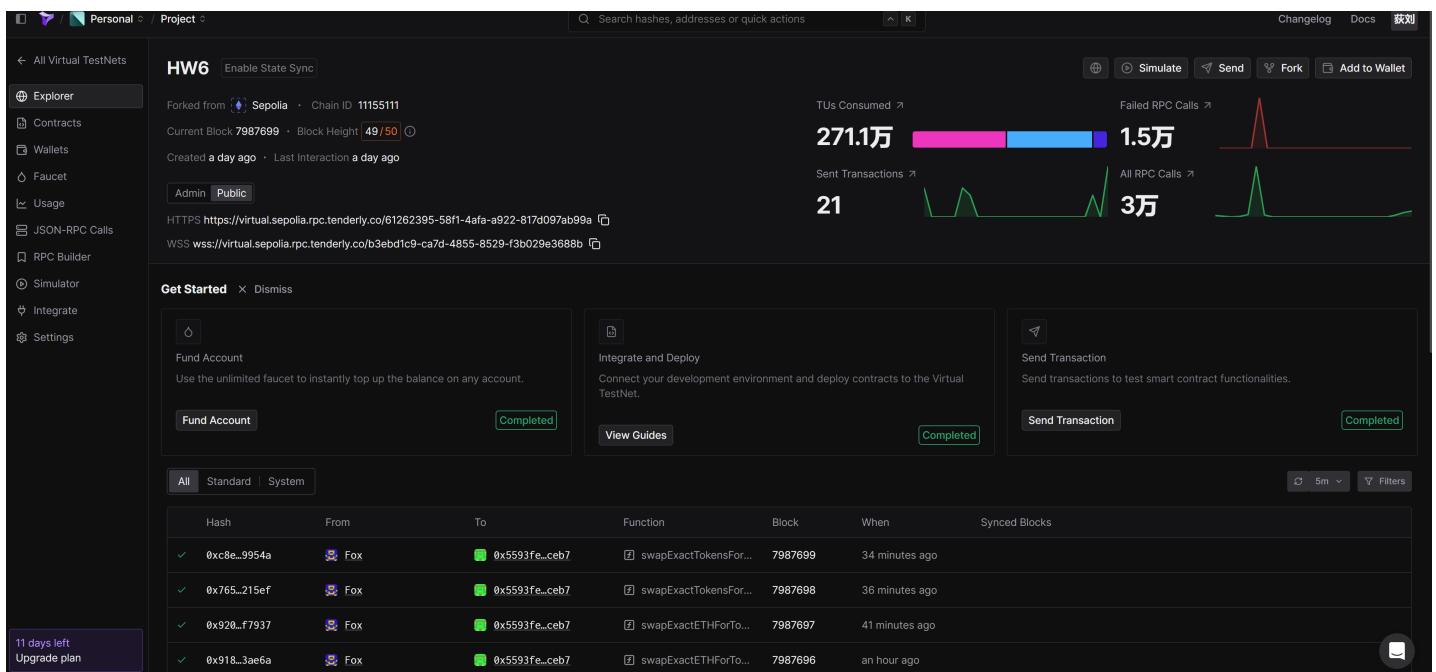
[HW6-Vercel](#)

[HW6-Github](#)

[Tenderly Public Link](#)

Tenderly virtual test net

Create a Tenderly virtual test net and deploy your contract on the test net



Tenderly test net

And deploy the contract use some scripts:

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.19;
3
4 import "forge-std/Script.sol";
5 import "forge-std/console.sol";
6 import "../contracts/UniswapV2Factory.sol";
7 import "../contracts/UniswapV2Router02.sol";
8 import "../contracts/test/WETH.sol";
9 import "../contracts/test/TestUSDC.sol";
10 import "../contracts/UniswapV2ERC20.sol";
11 import "../contracts/UniswapV2Pair.sol";
12 import "../contracts/UniswapV2Migrator.sol";
13
14 contract DeployScript is Script {
    function run() external {
```

```

15     uint256 deployerPrivateKey = vm.envUInt("PRIVATE_KEY");
16     address deployerAddress = vm.addr(deployerPrivateKey);
17     console.log("Deployer address: ", deployerAddress);
18
19     vm.startBroadcast(deployerPrivateKey);
20
21     // Deploy libraries first
22     // Note: In your case, you're already referencing deployed libraries
23     // But we'll add the deployment code here for completeness
24
25     console.log("Deploying Math library...");  

26     address mathLib = deployCode("contracts/libraries/Math.sol:Math");
27     console.log("Math library deployed at: ", mathLib);
28
29     console.log("Deploying TransferHelper library...");  

30     address transferHelperLib = deployCode("contracts/libraries/TransferHelper.s  
ol:TransferHelper");
31     console.log("TransferHelper library deployed at: ", transferHelperLib);
32
33     console.log("Deploying UniswapV2Library library...");  

34     address uniswapV2Lib = deployCode("contracts/libraries/UniswapV2Library.sol:  
UniswapV2Library");
35     console.log("UniswapV2Library library deployed at: ", uniswapV2Lib);
36
37     // Deploy test tokens
38     console.log("Deploying WETH...");  

39     WETH weth = new WETH();
40     console.log("WETH deployed at: ", address(weth));
41
42     console.log("Deploying TestUSDC...");  

43     TestUSDC testUSDC = new TestUSDC();
44     console.log("TestUSDC deployed at: ", address(testUSDC));
45
46     // Deploy main contracts
47     console.log("Deploying UniswapV2Factory...");  

48     UniswapV2Factory factory = new UniswapV2Factory(deployerAddress);
49     console.log("UniswapV2Factory deployed at: ", address(factory));
50
51     console.log("Deploying UniswapV2Router02...");  

52     UniswapV2Router02 router = new UniswapV2Router02(address(factory), address(w  
eth));
53     console.log("UniswapV2Router02 deployed at: ", address(router));
54
55     // Deploy additional contracts if needed
56     console.log("Deploying UniswapV2Migrator...");  

57     UniswapV2Migrator migrator = new UniswapV2Migrator(address(factory), address  
(router));
58     console.log("UniswapV2Migrator deployed at: ", address(migrator));
59
60     // Initialize a liquidity pool between WETH and TestUSDC
61     console.log("Creating WETH/TestUSDC pair...");  

62     address pair = factory.createPair(address(weth), address(testUSDC));
63     console.log("WETH/TestUSDC pair created at: ", pair);

```

```

64     // Mint some test tokens to the deployer
65     console.log("Minting test tokens...");
66     testUSDC.mint(deployerAddress, 1000000 * 10**6); // 1M USDC
67
68     // Approve router to spend tokens
69     testUSDC.approve(address(router), type(uint256).max);
70
71     // Add initial liquidity
72     console.log("Adding initial liquidity...");
73     uint256 wethAmount = 10 ether;
74     uint256 usdcAmount = 20000 * 10**6; // 20k USDC
75
76     // Wrap ETH to WETH
77     weth.deposit{value: wethAmount}();
78     weth.approve(address(router), type(uint256).max);
79
80     // Add liquidity
81     router.addLiquidity(
82         address(weth),
83         address(testUSDC),
84         wethAmount,
85         usdcAmount,
86         0, // min WETH
87         0, // min USDC
88         deployerAddress,
89         block.timestamp + 3600 // deadline: 1 hour from now
90     );
91
92     console.log("Initial liquidity added successfully");
93
94     vm.stopBroadcast();
95 }
96 }
```

Then run codes to deploy the contracts.

```

1 source .env
2 forge script script/DeployAll.s.sol:DeployScript --rpc-url="https://virtual.sepolia.
  rpc.tenderly.co/a6122906-66f1-4c1a-b4e7-92fecdc0e25" --broadcast --verify --etherscan-api-key=
  --verifier-url="https://virtual.sepolia.rpc.tenderly.co/a6122906-66f1-4c1a-b4e7-92fecdc0e25/verify/etherscan"
```

How to connect your wallet to HW6

1. Click on the network dropdown menu at the top of the MetaMask interface (it typically shows "Ethereum Mainnet" by default).
2. Click on "Security and Privacy" at the bottom of the dropdown menu.
3. Select "Network Providers"
4. Fill in the network details:

- Network Name: Enter a name for the network (e.g., "My Custom Network")
- New RPC URL: Enter the RPC endpoint URL before [Tenderly Public Link](#) (**Maybe I can't join the network without authorization because of the permissions settings.**)
- Chain ID: Enter the network's chain ID (a unique number identifying the blockchain)
- Currency Symbol: Enter the symbol for the native currency (e.g., "ETH", "MATIC")
- Block Explorer URL (optional): Enter the URL for the network's block explorer

5. Click "Save" to add the network.

网络提供商

选择您的网络

我们使用 Infura 作为我们的远程过程调用 (RPC) 提供商，以提供最可靠和最私密的以太坊数据访问。您可以选择自己的 RPC，但请谨记，任何 RPC 都可以接收您的 IP 地址和以太坊钱包以进行交易。请阅读我们的 [隐私政策](#)，进一步了解 Infura 如何处理数据。

添加自定义网络

动

Choose Network Providers

HW6 interface

Uniswap-Rewrite-ETH Ethereum Home Debug Contracts

980.8021 ETH Tenderly Virtual Sepolia Oxa402...D086

Uniswap V2

Decentralized Exchange - Trade, Provide Liquidity, Earn Yields

⚡ Select Pool

Create New Pair

Use ETH as pair

Token1

Token2

Existing Pairs

Pair #1

④ 1996.75 *

⚡ Please Select a Pool

More features will be shown after selecting a pool

Home page

Uniswap V2

Decentralized Exchange - Trade, Provide Liquidity, Earn Yields

Home page

Uniswap V2

Decentralized Exchange - Trade, Provide Liquidity, Earn Yields

⚡ Select Pool

Create New Pair

Use ETH as pair

Token1

Token2

Existing Pairs

Pair #1

④ 1996.75 *

Pool Status

Pool has liquidity, ready for trading

Pool Address: 0xA833c3627FC8487a273bBfa40f0C93971371392A

☒ Reserve Curve

Reserve Curve

Reserve Curve

Reserve Curve Current Price: 1 WETH = 744.083734 USDC



USDC Reserves

Legend: Reserve Curve (Blue Line), Current Position (Red Dot)

*

Add pool

+ Add Liquidity

Add Liquidity

Add liquidity to receive LP tokens and start trading

Token0 Amount (ETH)

0.0

ETH

Max

ETH Balance: 980.8021 ETH

Token1 Amount (USDC)

0.0

Balance: 997578.318857 USDC

Slippage (%)

Recommended: 1.0%

1.0

Add Liquidity



After adding liquidity, you will receive LP tokens representing your share in the pool.

This operation may require two transactions: one for approval and one for adding liquidity.

Troubleshooting & Help



Add liquidity

Reserve Curve

Reserve Curve

Refresh Data

Reserve Curve

Reserve Curve

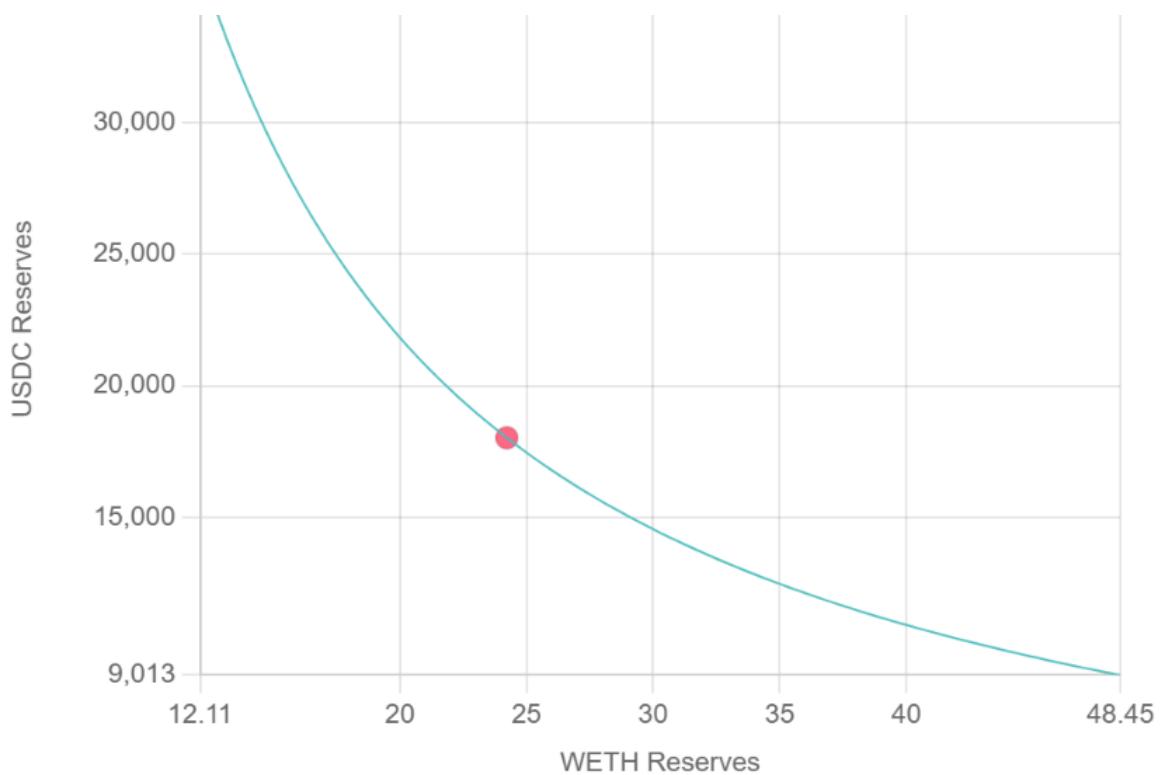
Current Price: 1 WETH = 744.083734
USDC

Refresh Chart

36,052.01

Reserve Curve Current Position

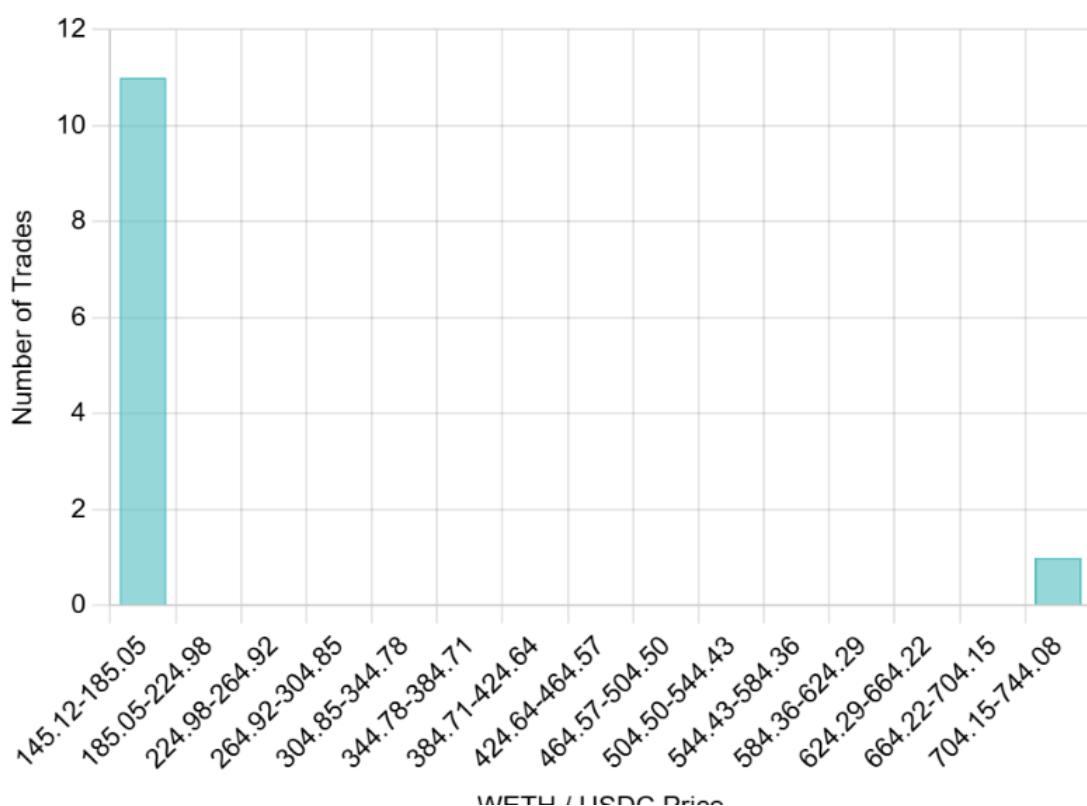




Price Distribution

Avg Price: 1 WETH = 197.279340
USDC

[Refresh Chart](#)



Chart

Swap Tokens

Token Swap
Swap tokens through liquidity pool

From (ETH): 0.0 **Max** **ETH**

To (USDC): 0.0 **Max** Balance: 997578.318857 USDC

Slippage (%): 0.5 Recommended: 0.5%

Swap

Swap operation exchanges one token for another through the liquidity pool.
This operation may require two transactions: one for approval and one for swapping.

Troubleshooting & Help

① 1996.75

User Guide

1. Select Pool
Choose or create a pool from the left panel
2. Add Liquidity
Add sufficient liquidity to enable trading
3. Start Trading
Once there's enough liquidity, you can swap tokens

Swap component

name "Test USDC" **Read**

symbol "USDC"

totalSupply 1000000000000000000

balanceOf address **Read**

Write

approve

spender address

address spender

amount uint256

uint256 amount *

Send

mint

to address

address to

amount uint256

uint256 amount *

Send

Deposit mint USDC