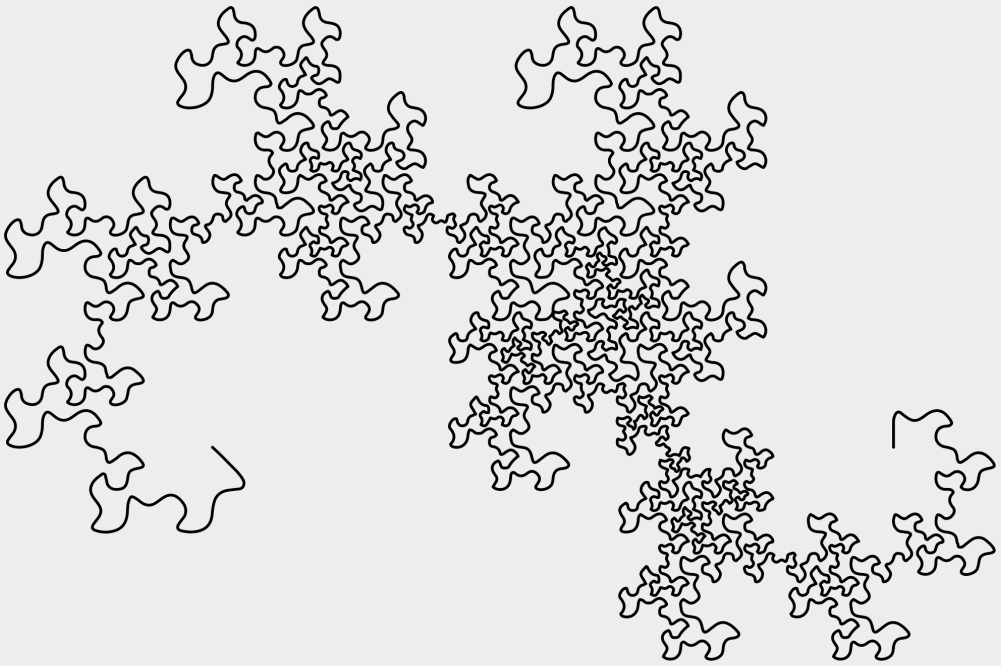
karpatkey

Manager role permissions - ENS Endowment (version 5)



Introduction

This document presents the Zodiac Roles Modifier permissions defining the Manager Role preset for the ENS Endowment Safe (Avatar) managed by karpatkey.

Permissions are organized by protocol, strategy and action.

- **Avatar Safe address:** 0x4F2083f5fBede34C2714aFfb3105539775f7FE64
- **Roles Modifier address:** 0x703806E61847984346d2D7DDd853049627e50A40
- **Manager Safe address:** 0xb423e0f6E7430fa29500c5cC9bd83D28c8BD8978

The Manager Role preset can be found at:

https://github.com/karpatkey/client-configs/blob/main/clients/ens/eth/roles/manage/permissions_actions.ts

The transactions created to test the initial configuration of the permissions have been created with DeFi-Kit:

https://github.com/karpatkey/defi-kit/tree/main/sdk/src/protocols

Note that Compound V3 was tested in the client-configs project, as it is still not a compatible protocol within DeFi-Kit.

Notation

- *argument[n]*: data stands for the permission only allowing data to be inputted as the n-th positional argument when calling the function.
- *valueETH*: True stands for the permission allowing the sending of ETH when calling the function.

Versions

Date	Version	Action	Details
May 5th, 2024	v5	Updated	Introduces new permissions and adjustments laid out in EP 5.12.
Oct 23rd, 2023	v4	Updated	Introduces new permissions and adjustments laid out in E.P 4.5.
Sept 1st, 2023	v3	Updated	Introduces minor adjustments laid out in E.P. 4.2.
June 15th, 2023	v2	Updated	Introduces additional permissions laid out in E.P. 4.1.
Jan 20th, 2023	-	Created	Roles Modifier deployed, initial permissions applied and document created

Changes from the previous version are coloured in green. Revoked permissions are coloured in ~~strikethrough red~~.

Permissions

Lido

Stake ETH

1. Stake ETH

Sender stakes ETH in exchange for stETH:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xae7ab96520DE3A18E5e111B5EaAb095312D7fE84",

signature: "**submit(address)**"

valueETH: True

}

2. Wrap stETH

Wraps stETH, sender receives wstETH:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x7f39C581F595B53c5cb19bD0b3f8dA6c935E2Ca0",

signature: "**wrap(uint256)**"

}

3. Unwrap wstETH

Unwraps wstETH, sender receives stETH:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x7f39C581F595B53c5cb19bD0b3f8dA6c935E2Ca0",

signature: "**unwrap(uint256)**"

}

4. Request stETH Withdrawal

Locks your stETH in the queue. In exchange you receive an NFT, that represents your position in the queue:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x889edC2eDab5f40e902b864aD4d7AdE8E412F9B1",

signature: "**requestWithdrawals(uint256[],address)**",

argument[1]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

}

OR; request withdrawal with permit:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x889edC2eDab5f40e902b864aD4d7AdE8E412F9B1",

signature: "**requestWithdrawalsWithPermit**(uint256[] _amounts, address _owner, tuple _permit)",

argument[1]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

}

5. Request wstETH Withdrawal

Transfers the wstETH to the unstETH to be burned in exchange for stETH. Then it locks your stETH in the queue. In exchange you receive an NFT, that represents your position in the queue:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x889edC2eDab5f40e902b864aD4d7AdE8E412F9B1",

signature: "**requestWithdrawalsWstETH(uint256[],address)**",

argument[1]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

}

OR; request withdrawal with permit:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x889edC2eDab5f40e902b864aD4d7AdE8E412F9B1",

signature: "**requestWithdrawalsWstETHWithPermit**(uint256[] _amounts, address _owner, tuple _permit)",

argument[1]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

}

6. Claim stETH

Once the request is finalized by the oracle report and becomes claimable claim your ether, burning the NFT:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x889edC2eDab5f40e902b864aD4d7AdE8E412F9B1",

signature: "**claimWithdrawals(uint256[],uint256[])**",

}

OR; claim a single withdrawal:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x889edC2eDab5f40e902b864aD4d7AdE8E412F9B1",

signature: "**claimWithdrawal**(uint256 _requestId)",

}

Compound v2

Lend USDC

1. Deposit USDC

Sender specifies the amount of the underlying asset to be supplied to mint the corresponding amount of cToken:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x39AA39c021dfbaE8faC545936693aC917d5E7563",

signature: "**mint(uint256)**"

}

2. Withdraw USDC (specific amount of cToken)

Sender redeems a specified amount of cTokens, it is called by the UI when MAX amount is set to be withdrawn:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x39AA39c021dfbaE8faC545936693aC917d5E7563",

signature: "**redeem(uint256)**"

}

3. Withdraw USDC (specific amount of underlying asset)

Sender redeems cTokens in exchange for a specified amount of the underlying asset, it is called by the UI when withdrawing an amount other than MAX amount:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x39AA39c021dfbaE8faC545936693aC917d5E7563",

signature: "**redeemUnderlying(uint256)**"

}

Lend - DAI

1. Deposit DAI

Sender specifies the amount of the underlying asset to be supplied to mint the corresponding amount of cToken:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x5d3a536E4D6DbD6114cc1Ead35777bAB948E3643",

signature: "**mint(uint256)**"

}

2. Withdraw DAI (specific amount of cToken)

Sender redeems a specified amount of cTokens, it is called by the UI when MAX amount is set to be withdrawn:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x5d3a536E4D6DbD6114cc1Ead35777bAB948E3643",

signature: "**redeem(uint256)**"

}

3. Withdraw DAI (specific amount of underlying asset)

Sender redeems cTokens in exchange for a specified amount of the underlying asset, it is called by the UI when withdrawing an amount other than MAX amount:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x5d3a536E4D6DbD6114cc1Ead35777bAB948E3643",

signature: "**redeemUnderlying(uint256)**"

}

~~Claim COMP rewards~~

~~1. Claim COMP~~

~~Claims accrued rewards by lending positions, allowing for all possible combinations: cUSDC + cDAI / cUSDC / cDAI:~~

~~{~~

~~from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",~~

~~to: "0x3d9819210a31b4961b30ef54be2aed79b9c9cd3b",~~

~~signature: "**claimComp(address,address[])**",~~

~~argument[0]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",~~

~~argument[1]:[["0x39AA39c021dfbaE8faC545936693aC917d5E7563",~~

~~"0x5d3a536E4D6DbD6114cc1Ead35777bAB948E3643"]~~

~~OR~~

~~["0x39AA39c021dfbaE8faC545936693aC917d5E7563"]~~

~~OR~~

~~["0x5d3a536E4D6DbD6114cc1Ead35777bAB948E3643"]~~

~~}~~

Compound v3

Lend USDC

1. Deposit USDC

Sender specifies the amount of the underlying asset to be supplied to mint the corresponding amount of cToken:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xc3d688B66703497DAA19211EEdff47f25384cdc3",

signature: "**supply(address,uint256)**",

argument[0]: "0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48"

}

2. Withdraw USDC

Sender redeems a specified amount of cTokens:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x39AA39c021dfbaE8faC545936693aC917d5E7563",

signature: "**withdraw(address,uint256)**”,

argument[0]: "0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48”

}

Lend ETH

1. Allow/Disallow Mainnet Bulker

Allows or disallows Compound v3 Mainnet Bulker to withdraw, or transfer from the sender:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xA17581A9E3356d9A858b789D68B4d866e593aE94",

signature: **"allow(address,bool)"**,

argument[0]: "0xa397a8C2086C554B531c02E29f3291c9704B00c7"

}

2. Deposit/Withdraw ETH

Sender specifies the amount of the underlying asset to be supplied to mint the corresponding amount of cToken / sender redeems a specified amount of cTokens:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xa397a8C2086C554B531c02E29f3291c9704B00c7",

signature: **"invoke(bytes32[],bytes[])”**,

argument[0]:
"0x414354494f4e5f535550504c595f4e4154956455f544f4b454e000000000000"

(ACTION_SUPPLY_NATIVE_TOKEN)

OR

"0x414354494f4e5f57495448445241575f4e4154956455f544f4b454e00000000"

(ACTION_WITHDRAW_NATIVE_TOKEN)

argument[1]: Encoded ABI for the action to be executed ->

Encoded ABI [address comet, address to, uint amount]:

["0xA17581A9E3356d9A858b789D68B4d866e593aE94",

"0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

undefined],

valueETH: True

}

Claim COMP rewards

1. Claim COMP

Claims accrued rewards by lending positions:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x1b0e765f6224c21223aea2af16c1c46e38885a40",

signature: "**claim(address,address,bool)**",

argument[0]: "0xc3d688B66703497DAA19211EEdff47f25384cdc3"

OR

"0xA17581A9E3356d9A858b789D68B4d866e593aE94",

argument[1]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

}

AAVE v3

Lend DAI

1. Deposit DAI

Sender specifies the amount of the underlying asset to be supplied to mint the corresponding amount of aToken:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x87870Bca3F3fD6335C3F4ce8392D69350B4fA4E2",

signature: "**supply(address,uint256,address,uint16)**",

argument[0]: "0x6B175474E89094C44Da98b954EedeAC495271d0F",

argument[2]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

OR; deposit as collateral:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x87870Bca3F3fD6335C3F4ce8392D69350B4fA4E2",

signature: "**setUserUseReserveAsCollateral**(address asset, bool useAsCollateral)",

argument[0]: "0x6B175474E89094C44Da98b954EedeAC495271d0F"

}

2. Withdraw DAI

Sender redeems a specified amount of cTokens:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x87870Bca3F3fD6335C3F4ce8392D69350B4fA4E2",

signature: "**withdraw(address,uint256,address)**”,

argument[0]: "0x6B175474E89094C44Da98b954EedeAC495271d0F",

argument[2]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

Lend USDC

1. Deposit USDC

Sender specifies the amount of the underlying asset to be supplied to mint the corresponding amount of aToken:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x87870Bca3F3fD6335C3F4ce8392D69350B4fA4E2",

signature: "**supply(address,uint256,address,uint16)**",

argument[0]: "0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48",

argument[2]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

OR; deposit as collateral:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x87870Bca3F3fD6335C3F4ce8392D69350B4fA4E2",

signature: "**setUserUseReserveAsCollateral**(address asset, bool useAsCollateral)",

argument[0]: "0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48"

}

2. Withdraw USDC

Sender redeems a specified amount of cTokens:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x87870Bca3F3fD6335C3F4ce8392D69350B4fA4E2",

signature: "**withdraw(address,uint256,address)**”,

argument[0]: "0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48",

argument[2]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

Lend ETH

1. Deposit ETH

Sender specifies the amount of the underlying asset to be supplied to mint the corresponding amount of aToken:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xD322A49006FC828F9B5B37Ab215F99B4E5caB19C",

signature: **"depositETH(address,address,uint16)"**,

argument[0]: "0x87870Bca3F3fD6335C3F4ce8392D69350B4fA4E2",

argument[1]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

valueETH: True

}

2. Withdraw ETH

ETH (non-ERC20 token) requires an approval before withdrawal:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x4d5F47FA6A74757f35C14fD3a6Ef8E3C9BC514E8",

signature: **"approve(address spender, uint256)”**,

argument[0]: "0x893411580e590D62dDBca8a703d61Cc4A8c7b2b9"

}

AND

Sender redeems a specified amount of aTokens:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xD322A49006FC828F9B5B37Ab215F99B4E5caB19C",

signature: **"withdrawETH(address,uint256,address)”**,

argument[0]: "0x87870Bca3F3fD6335C3F4ce8392D69350B4fA4E2",

argument[2]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

Lend WETH

1. Deposit WETH

Sender specifies the amount of the underlying asset to be supplied to mint the corresponding amount of aToken:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x87870Bca3F3fD6335C3F4ce8392D69350B4fA4E2",

signature: **"supply(address,uint256,address,uint16)"**,

argument[0]: "0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",

argument[2]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

OR; deposit as collateral:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x87870Bca3F3fD6335C3F4ce8392D69350B4fA4E2",

signature: "**setUserUseReserveAsCollateral**(address asset, bool useAsCollateral)",

argument[0]: "0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2"

}

2. Withdraw WETH

Sender redeems a specified amount of aTokens:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x87870Bca3F3fD6335C3F4ce8392D69350B4fA4E2",

signature: **"withdraw(address,uint256,address)”**,

argument[0]: "0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",

argument[2]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

Stakewise

WETH + sETH2

1. Stake ETH

Sender stakes ETH in exchange for sETH2:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xC874b064f465bdD6411D45734b56fac750Cda29A",

signature: "**stake()**",

valueETH: True

}

2. Claim rewards

Claims rETH2 rewards accrued by the staking of ETH:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xA3F21010e8b9a3930996C8849Df38f9Ca3647c20",

signature: "**claim(uint256,address,address[],uint256[],bytes32[])**",

argument[1]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

argument[2]: ["0x20BC832ca081b91433ff6c17f85701B6e92486c5",

"0x48C3399719B582dD63eB5AADf12A40B4C3f52FA2"]

}

Uniswap V3 WETH + sETH2, 0.3%

1. Add liquidity

Creates a new position in the pool, minting the NFT:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xC36442b4a4522E871399CD717aBDD847Ab11FE88",

Signature: "**mint((address,address,uint24,int24,int24,uint256,uint256,uint256,uint256,address,uint256))**",

argument[0]: "0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",

argument[1]: "0xFe2e637202056d30016725477c5da089Ab0A043A",

argument[2]: 3000,

argument[9]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

2. Increase liquidity

Increases the liquidity in the pool; NFT ID 424810 was created in transaction with hash

0x2995ba040fe1b07978428ca118d9701b5114ec7e2d3ac00f2b4df0f5747dc42e:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xC36442b4a4522E871399CD717aBDD847Ab11FE88",

Signature: "**increaseLiquidity((uint256,uint256,uint256,uint256,uint256,uint256))**",

argument[0]: 424810

}

3. Remove liquidity

Burns the token amounts in the pool, and increases token0Owed and token1Owed which represent the uncollected fees:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xC36442b4a4522E871399CD717aBDD847Ab11FE88",

signature: "**decreaseLiquidity((uint256,uint128,uint256,uint256,uint256))**"

}

4. Collect Fees

Collects token0Owed and token1Owed, which amount to the uncollected fees plus any liquidity previously removed:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xC36442b4a4522E871399CD717aBDD847Ab11FE88",

signature: "**collect((uint256,address,uint128,uint128))**",

argument[1]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

Curve

stETH + ETH

1. Add liquidity

Adds liquidity to the pool:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xDC24316b9AE028F1497c275EB9192a3Ea0f67022",

signature: "**add_liquidity(uint256[2],uint256)**",

valueETH: True

}

2. Remove liquidity (one coin)

Removes liquidity from the pool, withdrawing the total amount in one single token:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xDC24316b9AE028F1497c275EB9192a3Ea0f67022",

signature: "**remove_liquidity_one_coin(uint256,int128,uint256)**"

}

3. Remove liquidity (balanced amounts)

Removes liquidity from the pool, withdrawing balanced amounts:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xDC24316b9AE028F1497c275EB9192a3Ea0f67022",

signature: "**remove_liquidity(uint256,uint256[2])**"

}

4. Remove liquidity (imbalanced amounts)

Removes liquidity from the pool, withdrawing custom amounts:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xDC24316b9AE028F1497c275EB9192a3Ea0f67022",

signature: "**remove_liquidity_imbalance(uint256[2],uint256)**”

}

5. Stake LP token

Stakes LP token in gauge:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x182B723a58739a9c974cFDB385ceaDb237453c28",

signature: "**deposit(uint256)**"

}

6. Unstake LP token

Unstakes LP token from gauge:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x182B723a58739a9c974cFDB385ceaDb237453c28",

signature: "**withdraw(uint256)**"

}

7. Claim LDO rewards

Claims accrued LDO from staking:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x182B723a58739a9c974cFDB385ceaDb237453c28",

signature: "**claim_rewards()**"

}

8. Claim CRV rewards

Claims accrued CRV from staking:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xd061D61a4d941c39E5453435B6345Dc261C2fcE0",

signature: "**mint(address)**",

argument[0]: "0x182B723a58739a9c974cFDB385ceaDb237453c28"

}

9. Stake and Deposit ZAP approval

Allows the Stake and Deposit ZAP to deposit fund in the gauge:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x182B723a58739a9c974cFDB385ceaDb237453c28",

signature: "**set_approve_deposit(address,bool)**",

argument[0]: "0x56C526b0159a258887e0d79ec3a80dfb940d0cD7”

}

stETH-ng (factory)

1. Add liquidity

Adds liquidity to the pool:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x21E27a5E5513D6e65C4f830167390997aA84843a",

signature: "**add_liquidity**(uint256[2],uint256)",

}

2. Remove liquidity (one coin)

Removes liquidity from the pool, withdrawing the total amount in one single token:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x21E27a5E5513D6e65C4f830167390997aA84843a",

signature: "**remove_liquidity_one_coin**(uint256,int128,uint256)"

}

3. Remove Liquidity (balanced amounts)

Removes liquidity from the pool, withdrawing balanced amounts:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x21E27a5E5513D6e65C4f830167390997aA84843a",

signature: "**remove_liquidity**(uint256,uint256[2])"

}

4. Remove Liquidity (imbalanced amounts)

Removes liquidity from the pool, withdrawing custom amounts:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x21E27a5E5513D6e65C4f830167390997aA84843a",

signature: "**remove_liquidity_imbalance**(uint256[2],uint256)”

}

5.

Stake BPT

Stakes BPT in gauge:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x79F21BC30632cd40d2aF8134B469a0EB4C9574AA",

signature: "**deposit(uint256)**"

}

6. Unstake BPT

Unstakes BPT from gauge:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x79F21BC30632cd40d2aF8134B469a0EB4C9574AA",

signature: "**withdraw(uint256)**"

}

7. Claim wstETH rewards

Claims rewards:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x79F21BC30632cd40d2aF8134B469a0EB4C9574AA",

signature: "**claim_rewards()**"

}

8. Claim CRV rewards

Claims accrued CRV from staking:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xd061D61a4d941c39E5453435B6345Dc261C2fcE0",

signature: "**mint(address)**",

argument[0]: "0x79F21BC30632cd40d2aF8134B469a0EB4C9574AA”

}

cDAI + cUSDC

1. Add liquidity

Adds liquidity to the pool:

```
{
    from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",
    to: "0xA2B47E3D5c44877cca798226B7B8118F9BFb7A56",
    signature: "add_liquidity(uint256[2],uint256)",
}
```

2. Add liquidity using a ZAP contract

Adds liquidity to the pool using a ZAP contract which allows the deposit of underlying tokens:

```
{
    from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",
    to: "0xeB21209ae4C2c9FF2a86ACA31E123764A3B6Bc06",
    signature: "add_liquidity(uint256[2],uint256)",
}
```

3. Remove liquidity (one coin)

Removes liquidity from the pool using a ZAP contract which allows the withdrawal of an specific amount in one single underlying token:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xeB21209ae4C2c9FF2a86ACA31E123764A3B6Bc06",

signature: "**remove_liquidity_one_coin(uint256,int128,uint256)**"

}

4. Remove liquidity (balanced amounts)

Removes liquidity from the pool, withdrawing balanced amounts:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xA2B47E3D5c44877cca798226B7B8118F9BFb7A56",

signature: "**remove_liquidity(uint256,uint256[2])**"

}

5. Remove liquidity using a ZAP contract (balanced amounts)

Removes liquidity from the pool using a ZAP contract which allows the withdrawal of balanced underlying amounts:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xeB21209ae4C2c9FF2a86ACA31E123764A3B6Bc06",

signature: "**remove_liquidity(uint256,uint256[2])**"

}

6. Remove liquidity (imbalanced amounts)

Removes liquidity from the pool, withdrawing custom amounts:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xA2B47E3D5c44877cca798226B7B8118F9BFb7A56",

signature: "**remove_liquidity_imbalance(uint256[2],uint256)**"

}

7. Remove liquidity using a ZAP contract (imbalanced amounts)

Removes liquidity from the pool using a ZAP contract which allows the withdrawal of custom underlying amounts:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xeB21209ae4C2c9FF2a86ACA31E123764A3B6Bc06",

signature: "**remove_liquidity_imbalance(uint256[2],uint256)**"

}

~~8. Stake LP token~~

~~Stakes LP token in gauge:~~

~~{~~

~~from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",~~

~~to: "0x7ca5b0a2910B33e9759DC7dDB0413949071D7575",~~

~~signature: "**deposit(uint256)**"~~

~~}~~

~~9. Unstake LP token~~

~~Unstakes LP token from gauge:~~

~~{~~

~~from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",~~

~~to: "0x7ca5b0a2910B33e9759DC7dDB0413949071D7575",~~

~~signature: "**withdraw(uint256)**"~~

~~}~~

10. Claim CRV rewards

Claims accrued CRV from staking:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xd061D61a4d941c39E5453435B6345Dc261C2fcE0",

signature: "**mint(address)**",

argument[0]: "0x7ca5b0a2910B33e9759DC7dDB0413949071D7575"

OR

argument[0]: "0x182B723a58739a9c974cFDB385ceaDb237453c28"

}

~~11. Stake and Deposit ZAP approval~~

~~Allows the Stake and Deposit ZAP to deposit fund in the gauge:~~

~~{~~

~~from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",~~

~~to: "0x7ca5b0a2910B33e9759DC7dDB0413949071D7575",~~

~~signature: "**set_approve_deposit(address,bool)**",~~

~~argument[0]: "0x271fbE8aB7f1fB262f81C77Ea5303F03DA9d3d6A"~~

~~}~~

Swapping of tokens

~~cDai + cUSDC pool (cDAI, cUSDC, DAI and USDC)~~

~~1. Swap cDAI for cUSDC or cUSDC for cDAI~~

~~{~~

~~from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",~~

~~to: "0xA2B47E3D5c44877cca798226B7B8118F9BFb7A56",~~

~~signature: "**exchange(int128,int128,uint256,uint256)**",~~

~~}~~

~~2. Swap DAI for USDC or USDC for DAI~~

~~{~~

~~from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",~~

~~to: "0xA2B47E3D5c44877cca798226B7B8118F9BFb7A56",~~

~~signature: "**exchange_underlying(int128,int128,uint256,uint256)**",~~

~~}~~

stETH + ETH pool (ETH, stETH)

1. Swap ETH for stETH or stETH for ETH

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xDC24316b9AE028F1497c275EB9192a3Ea0f67022",

signature: "**exchange(int128,int128,uint256,uint256)**",

valueETH: True

}

3pool (DAI, USDC, USDT)

- Swap DAI for USDC.
- Swap DAI for USDT.
- Swap USDC for DAI.
- Swap USDC for USDT.
- Swap USDT for DAI.
- Swap USDT for USDC.

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xbEbc44782C7dB0a1A60Cb6fe97d0b483032FF1C7",

signature: "**exchange(int128,int128,uint256,uint256)**"

}

CVX + WETH pool

1. Swap CVX for WETH

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xDC24316b9AE028F1497c275EB9192a3Ea0f67022",

signature: "**exchange(int128,int128,uint256,uint256)**",

argument[0]: 1,

argument[1]: 0

}

ankrETH + ETH pool

1. Swap ankrETH <> ETH

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xA96A65c051bF88B4095Ee1f2451C2A9d43F53Ae2",

signature: **"exchange(int128,int128,uint256,uint256)"**,

valueETH: True

```
}
```

Deposit and Stake in one transaction using a special ZAP contract

1. Add liquidity + Stake

Adds liquidity to the pool and stakes in the gauge in the same transaction:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x271fbE8aB7f1fB262f81C77Ea5303F03DA9d3d6A",

signature:

"**deposit_and_stake(address,address,address,uint256,address[5],uint256[5],uint256,bool,address)**",

argument[0]: ["0xDC24316b9AE028F1497c275EB9192a3Ea0f67022",

OR

"0xA2B47E3D5c44877cca798226B7B8118F9BFb7A56"

OR

"0xcB21209ac4C2c9FF2a86ACA31E123764A3B6Bc06"],

argument[1]: ["0x06325440D014e39736583c165C2963BA99fAf14E",

OR

"0x845838DF265Dcd2c412A1Dc9e959c7d08537f8a2"],

argument[2] : ["0x182B723a58739a9c974cFDB385ceaDb237453c28",

OR

"0x7ca5b0a2910B33e9759DC7dDB0413949071D7575"],

argument[3]: 2,

argument[4]: [["0xEeeeeEeeeEeEeeEeEeEeeEEEeeeeEeeeeeeeEEeE",

"0xae7ab96520DE3A18E5e111B5EaAb095312D7fE84",

"0x0000000000000000000000000000000000000000",

"0x0000000000000000000000000000000000000000",

"0x0000000000000000000000000000000000000000"]

OR

["0x6B175474E89094C44Da98b954EedeAC495271d0F",

"0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48",

"0x0000000000000000000000000000000000000000",

"0x0000000000000000000000000000000000000000",

"0x0000000000000000000000000000000000000000"]

OR

["0x39AA39c021dfbaE8faC545936693aC917d5E7563",

"0x5d3a536E4D6DbD6114cc1Ead35777bAB948E3643",

"0x0000000000000000000000000000000000000000",

"0x0000000000000000000000000000000000000000",

~~"0x0000000000000000000000000000000000000000"]],~~

~~argument[8]: 0x4F2083f5fBede34C2714aFfb3105539775f7FE64,~~

~~valueETH: True~~

~~}~~

2. Add liquidity + Stake

Adds liquidity to the pool and stakes in the gauge in the same transaction:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x56C526b0159a258887e0d79ec3a80dfb940d0cD7",

signature:

"**deposit_and_stake(address,address,address,uint256,address[],uint256[],uint256,bool,bool,address)**",

argument[0]: ["0xDC24316b9AE028F1497c275EB9192a3Ea0f67022",

OR

"0x21E27a5E5513D6e65C4f830167390997aA84843a”],

argument[1]: ["0x06325440D014e39736583c165C2963BA99fAf14",

OR

"0x21E27a5E5513D6e65C4f830167390997aA84843a"],

argument[2] : ["0x79F21BC30632cd40d2aF8134B469a0EB4C9574AA",

OR

"0x182B723a58739a9c974cFDB385ceaDb237453c28"],

argument[3]: 2,

argument[4]: [["0xEeeeeEeeeEeEeeEeEeEeeEEEeeeeEeeeeeeeEEeE",

"0xae7ab96520DE3A18E5e111B5EaAb095312D7fE84"]

OR

["0x6B175474E89094C44Da98b954EedeAC495271d0F",

"0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48"]

OR

["0x39AA39c021dfbaE8faC545936693aC917d5E7563",

"0x5d3a536E4D6DbD6114cc1Ead35777bAB948E3643"]],

argument[8]: 0x4F2083f5fBede34C2714aFfb3105539775f7FE64,

valueETH: True

}

Aura

wstETH + WETH

3. Deposit and stake BPT

Adds liquidity to Balancer’s pool using the BPT and stakes the BPT in Aura’s rewarder:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xA57b8d98dAE62B26Ec3bcC4a365338157060B234",

signature: "**deposit(uint256,uint256,bool)**",

argument[0]: 153

}

4. Deposit and stake underlying tokens

Adds liquidity to Balancer’s pool using the underlying tokens (wstETH or WETH) and stakes the BPT in Aura’s rewarder:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xB188b1CB84Fb0bA13cb9ee1292769F903A9feC59",

signature:

"**depositSingle(address,address,uint256,bytes32,(address[],uint256[],bytes,bool))**"

argument[0]: "0x2a14dB8D09dB0542f6A371c0cB308A768227D67D",

argument[1]: ["0xae7ab96520DE3A18E5e111B5EaAb095312D7fE84"

OR

"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2"],

argument[3]:

"0x93d199263632a4ef4bb438f1feb99e57b4b5f0bd0000000000000000000000005c2",

}

5. Withdraw and unwrap BPT

Unstakes BPT:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x2a14dB8D09dB0542f6A371c0cB308A768227D67D",

signature: "**withdrawAndUnwrap(uint256,bool)**"

}

6. Claim rewards

Claims accrued rewards from staking:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x2a14dB8D09dB0542f6A371c0cB308A768227D67D",

signature: "**getReward()**"

}

7. Claim extra rewards

Claims base and extra rewards:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x2a14dB8D09dB0542f6A371c0cB308A768227D67D",

signature: "**getReward(address,bool)**"

argument[0]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

}

rETH + WETH

1. Deposit and stake BPT

Adds liquidity to Balancer's pool using the BPT and stakes the BPT in Aura's rewarder:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xA57b8d98dAE62B26Ec3bcC4a365338157060B234",

signature: "deposit(uint256,uint256,bool)",

argument[0]: 109

}

2. Deposit and stake underlying tokens

Adds liquidity to Balancer's pool using the underlying tokens (rETH or WETH) and stakes the BPT in Aura's rewarder:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xB188b1CB84Fb0bA13cb9ee1292769F903A9feC59",

Signature:

"**depositSingle(address,address,uint256,bytes32,(address[],uint256[],bytes,bool))**"

argument[0]: "0xDd1fE5AD401D4777cE89959b7fa587e569Bf125D",

argument[1]: ["0xae78736Cd615f374D3085123A210448E74Fc6393"

OR

"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2"],

argument[3]:

"0x1e19cf2d73a72ef1332c882f20534b6519be0276000200000000000000000112",

argument[4]: ["0xae78736Cd615f374D3085123A210448E74Fc6393",

"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2"]

}

3. Withdraw and unwrap BPT

Unstakes BPT:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xDd1fE5AD401D4777cE89959b7fa587e569Bf125D",

signature: "**withdrawAndUnwrap(uint256,bool)**"

}

4. Claim rewards

Claims accrued rewards from staking:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xDd1fE5AD401D4777cE89959b7fa587e569Bf125D",

signature: "**getReward()**"

}

5. Claim extra rewards

Claims base and extra rewards:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xDd1fE5AD401D4777cE89959b7fa587e569Bf125D",

signature: "**getReward(address,bool)**"

argument[0]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

}

Balancer

wstETH + WETH

9. Join pool

Adds liquidity to the pool:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xBA12222222228d8Ba445958a75a0704d566BF2C8",

signature: "**joinPool(bytes32,address,address,(address[],uint256[],bytes,bool))**",

argument[0]: "0x93d199263632a4ef4bb438f1feb99e57b4b5f0bd0000000000000000000005c2",

argument[1]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

argument[2]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

10. Exit pool

Removes liquidity from the pool:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xBA12222222228d8Ba445958a75a0704d566BF2C8",

signature: "**exitPool(bytes32,address,address,(address[],uint256[],bytes,bool))**",

argument[0]: "0x93d199263632a4ef4bb438f1feb99e57b4b5f0bd0000000000000000000005c2",

argument[1]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

argument[2]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

11. Stake BPT

Stakes BPT in gauge:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x2a14dB8D09dB0542f6A371c0cB308A768227D67D",

signature: "**deposit(uint256)**"

}

12. Unstake BPT

Unstakes BPT from gauge:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x2a14dB8D09dB0542f6A371c0cB308A768227D67D",

signature: "**withdraw(uint256)**"

}

13. Claim rewards

Claims rewards:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x2a14dB8D09dB0542f6A371c0cB308A768227D67D",

signature: "**claim_rewards()**"

}

14. Claim BAL rewards

Claims BAL rewards:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x239e55f427d44c3cc793f49bfb507ebe76638a2b",

signature: "**mint(address)**",

argument[0]: "0x2a14dB8D09dB0542f6A371c0cB308A768227D67D"

}

rETH + WETH

1. Join pool

Adds liquidity to the pool:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xBA12222222228d8Ba445958a75a0704d566BF2C8",

signature: "**joinPool(bytes32,address,address,(address[],uint256[],bytes,bool))**",

argument[0]: "0x1e19cf2d73a72ef1332c882f20534b6519be0276000200000000000000000112",

argument[1]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

argument[2]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

2. Exit pool

Removes liquidity from the pool:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xBA12222222228d8Ba445958a75a0704d566BF2C8",

signature:
"**exitPool(bytes32,address,address,(address[],uint256[],bytes,bool))**",

argument[0]:
"0x1e19cf2d73a72ef1332c882f20534b6519be027600020000000000000000112",

argument[1]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

argument[2]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

3. Stake BPT

Stakes BPT in gauge:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x79eF6103A513951a3b25743DB509E267685726B7",

signature: "**deposit(uint256)**"

}

4. Unstake BPT token

Unstakes BPT from gauge:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x79eF6103A513951a3b25743DB509E267685726B7",

signature: "**withdraw(uint256)**"

}

5. Claim rewards

Claims rewards:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x79eF6103A513951a3b25743DB509E267685726B7",

signature: "**claim_rewards()**"

}

6. Claim BAL rewards

Claims BAL rewards:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x239e55f427d44c3cc793f49bfb507ebe76638a2b",

signature: "**mint(address)**",

argument[0]: "0x79eF6103A513951a3b25743DB509E267685726B7"

}

Swapping of tokens (ankrETH, AURA, BAL, COMP, ETHx, rETH, WETH, wstETH)

- **Pools List**: [ankrETH/wstETH, AURA/WETH, BAL/WETH, COMP/WETH, ETHx/WETH, rETH/WETH, WETH/DAI, WETH/USDC, wstETH/WETH]
- **Token In List**: [ankrETH, AURA, BAL, COMP, ETHx, rETH, WETH, wstETH]
- **Token Out List**: [ankrETH, DAI, ETHx, rETH, USDC, WETH, wstETH]

{

"**swap((bytes32,uint8,address,address,uint256,bytes),(address,bool,address,bool),uint256,uint256)**",

argument[0]:

["0xcfca23ca9ca720b6e98e3eb9b6aa0ffc4a5c08b9000200000000000000000274"

OR

"0x5c6ee304399dbdb9c8ef030ab642b10820db8f56000200000000000000000014"

OR

"0x0b09dea16768f0799065c475be02919503cb2a3500020000000000000000001a"

OR

"0x96646936b91d6b9d7d0c47c496afbf3d6ec7b6f8000200000000000000000019"

OR

"0xefaa1604e82e1b3af8430b90192c1b9e8197e37700020000000000000000021"

~~OR~~

~~"0x32296969ef14eb0c6d29669c550d4a044913023000020000000000000080"~~

"0xf01b0684c98cd7ada480bfdf6e43876422fa1fc10002000000000000000005de",

OR

“0x93d199263632a4ef4bb438f1feb99e57b4b5f0bd0000000000000000000005c2”,

OR

"0x1e19cf2d73a72ef1332c882f20534b6519be027600020000000000000000112",

OR

"0xdfe6e7e18f6cc65fa13c8d8966013d4fda74b6ba0000000000000000000000558",

OR

"0x37b18b10ce5635a84834b26095a0ae5639dcb752000000000000000000000005cb"],

argument[2]: ["0xE95A203B1a91a908F9B9CE46459d101078c2c3cb"

OR

"0xC0c293ce456fF0ED870ADd98a0828Dd4d2903DBF"

OR

"0xba100000625a3754423978a60c9317c58a424e3D"

OR

"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2"

OR

"0xc00e94Cb662C3520282E6f5717214004A7f26888"

OR

"0x7f39C581F595B53c5cb19bD0b3f8dA6c935E2Ca0"

OR

"0xA35b1B31Ce002FBF2058D22F30f95D405200A15b"

OR

"0xae78736Cd615f374D3085123A210448E74Fc6393"],

argument[3]: ["0xE95A203B1a91a908F9B9CE46459d101078c2c3cb"

OR

"0x6B175474E89094C44Da98b954EedeAC495271d0F"

OR

"0xA35b1B31Ce002FBF2058D22F30f95D405200A15b"

OR

"0xae78736Cd615f374D3085123A210448E74Fc6393"

OR

"0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48"

OR

"0x7f39C581F595B53c5cb19bD0b3f8dA6c935E2Ca0"

OR

"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2"],

argument[6]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

argument[8]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64”

}

IMPORTANT: the first chunk of the swap functions' data is constrained to ensure it always is “0x0000000000000000000000000000000000000000000000000000000000000e0”. This check is a MUST to avoid the possibility of getting a malicious swap.

Convex

ETH + stETH (Curve LP: steCRV)

1. Deposit LP token

Adds liquidity to Curve’s pool using the LP token:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xF403C135812408BFbE8713b5A23a04b3D48AAE31",

signature: "**deposit(uint256,uint256,bool)**",

argument[0]: 25

}

2. Deposit LP token (total amount)

Adds liquidity to Curve’s pool using the total LP token amount:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xF403C135812408BFbE8713b5A23a04b3D48AAE31",

signature: "**depositAll(uint256,bool)**",

argument[0]: 25

}

3. Withdraw LP token

Withdraws LP token:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xF403C135812408BFbE8713b5A23a04b3D48AAE31",

signature: "**withdraw(uint256,uint256)**",

argument[0]: 25

}

4. Stake LP token

Stakes LP token in rewarder:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x0A760466E1B4621579a82a39CB56Dda2F4E70f03",

signature: "**stake(uint256)**"

}

5. Unstake LP token

Unstakes LP token from rewarder:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x0A760466E1B4621579a82a39CB56Dda2F4E70f03",

signature: "**withdraw(uint256)**"

}

6. Unstake and withdraw LP token

Unstakes LP token from rewarder and withdraws BPT:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x0A760466E1B4621579a82a39CB56Dda2F4E70f03",

signature: "**withdrawAndUnwrap(uint256,bool)**"

}

7. Claim rewards

Claims rewards:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x0A760466E1B4621579a82a39CB56Dda2F4E70f03",

signature: "**getReward(address,bool)**",

argument[0]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

8. Claim rewards (through Base RewardPool)

Claims rewards:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x6B27D7BC63F1999D14fF9bA900069ee516669ee8",

signature: "**getReward(address,bool)**",

argument[0]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

9. Withdraw (through Base RewardPool)

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x6B27D7BC63F1999D14fF9bA900069ee516669ee8",

signature: "**withdraw(address,bool)**"

}

10. Stake(through Base RewardPool)

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x6B27D7BC63F1999D14fF9bA900069ee516669ee8",

signature: "**stake(uint256)**"

}

11. Withdraw and Unwrap (through Base RewardPool)

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x6B27D7BC63F1999D14fF9bA900069ee516669ee8",

signature: "**withdrawAndUnwrap(uint256, bool)**"

}

ETH - stETH (Curve LP: stETH-ng-f)

1. Deposit LP token

Adds liquidity to Curve's pool using the LP token:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xF403C135812408BFbE8713b5A23a04b3D48AAE31",

signature: "**deposit(uint256,uint256,bool)**",

argument[0]: 177

}

2. Deposit LP token (total amount)

Adds liquidity to Curve's pool using the total LP token amount:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xF403C135812408BFbE8713b5A23a04b3D48AAE31",

signature: "**depositAll(uint256,bool)**",

argument[0]: 177

}

3. Withdraw LP token

Withdraws LP token:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xF403C135812408BFbE8713b5A23a04b3D48AAE31",

signature: "**withdraw(uint256,uint256)**",

argument[0]: 177

}

~~cDAI + cUSDC~~

~~1. Deposit LP token~~

~~Adds liquidity to Curve's pool using the LP token:~~

~~{~~

~~from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",~~

~~to: "0xF403C135812408BFbE8713b5A23a04b3D48AAE31",~~

~~signature: "**deposit(uint256,uint256,bool)**",~~

~~argument[0]: 0~~

~~}~~

~~2. Deposit LP token (total amount)~~

~~Adds liquidity to Curve's pool using the total LP token amount:~~

~~{~~

~~from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",~~

~~to: "0xF403C135812408BFbE8713b5A23a04b3D48AAE31",~~

~~signature: "**depositAll(uint256,bool)**",~~

~~argument[0]: 0~~

~~}~~

~~3. Withdraw LP token~~

~~Withdraws LP token:~~

~~{~~

~~from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",~~

~~to: "0xF403C135812408BFbE8713b5A23a04b3D48AAE31",~~

~~signature: "**withdraw(uint256,uint256)**",~~

argument[0]: 0

}

4. Stake LP token

Stakes LP token in rewarder:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xf34DFF761145FF0B05e917811d488B441F33a968",

signature: "**stake(uint256)**"

}

5. Unstake LP token

Unstakes LP token from rewarder:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xf34DFF761145FF0B05e917811d488B441F33a968",

signature: "**withdraw(uint256)**"

}

6. Unstake and withdraw LP token

Unstakes LP token from rewarder and withdraws BPT:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xf34DFF761145FF0B05e917811d488B441F33a968",

signature: "**withdrawAndUnwrap(uint256,bool)**"

}

7. Claim rewards

Claims rewards:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xf34DFF761145FF0B05e917811d488B441F33a968",

signature: "**getReward(address,bool)**",

argument[0]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

Maker

DSR (DAI Savings Rate)

1. Deposit DAI

Deposits DAI:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x373238337Bfe1146fb49989fc222523f83081dDb",

signature: "**join(address,uint256)**",

argument[0]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

2. Withdraw DAI

Withdraws DAI:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x373238337Bfe1146fb49989fc222523f83081dDb",

signature: "**exit(address,uint256)**",

argument[0]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

3. Withdraw DAI (total amount)

Withdraws total amount of DAI:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x373238337Bfe1146fb49989fc222523f83081dDb",

signature: "**exitAll(address)**",

argument[0]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

Rocket Pool

Stake ETH

1. Deposit

Deposits ETH in exchange for rETH:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xDD3f50F8A6CafbE9b31a427582963f465E745AF8",

signature: "**deposit()**",

valueETH: True

}

2. Withdraw

Withdraws rETH in exchange for ETH:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xae78736Cd615f374D3085123A210448E74Fc6393",

signature: "**burn(uint256)**",

argument[0]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

3. Swap ETH for rETH through Swap Router

When there is not enough rETH on the Deposit Pool in exchange for the ETH being deposited, the Swap Router swaps the ETH for rETH in secondary markets (Balancer and Uniswap):

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x16d5a408e807db8ef7c578279beeee6b228f1c1c",

signature: "**swapTo(uint256,uint256,uint256,uint256)**",

valueETH: True

}

4. Swap rETH for ETH through Swap Router

When there is not enough ETH on the Deposit Pool in exchange for the rETH you are depositing, the Swap Router swaps the rETH for ETH in secondary markets (Balancer and Uniswap):

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x16d5a408e807db8ef7c578279beeee6b228f1c1c",

signature: "**swapFrom(uint256,uint256,uint256,uint256,uint256)**",

}

Ankr

Stake ETH

1. Stake ETH

Stakes a given amount of ETH in exchange for ankrETH

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x84db6ee82b7cf3b47e8f19270abde5718b936670",

signature: **"stakeAndClaimAethC()"**,

valueETH: True

}

2. Standard Unstake ETH

Burns the ankrETH and the distributeRewards() function is periodically called to transfer different amounts of ETH to the users on the Ethereum unstake queue, until it completes the total unstaked amount of each (within the 6 day window):

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x84db6ee82b7cf3b47e8f19270abde5718b936670",

signature: **"unstakeAETH()"**,

valueETH: True

}

3. Flash Unstake ETH

Uses a pool to swap your Liquid Staking tokens for your original assets, which means instant release of your funds. While it offers instant release of your funds, it poses a few limitations:

- You have to pay a technical service fee for a flash unstake — 0.5% of the unstaked amount.
- Your unstake is limited by the current capacity of the flash-unstake pool. If you exceed it, the interface switches to the standard unstake with its regular release time.

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xf047f23ACFdB1315cF63Ad8aB5146d5fDa4267Af",

signature: **"swapEth(uint256,address)"**,

argument[1]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

Stader

Stake ETH

1. Stake ETH

Stakes a given amount of ETH in exchange for ETHx:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xcf5EA1b38380f6aF39068375516Daf40Ed70D299",

signature: **"deposit(address)"**,

argument[0]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

valueETH: True

}

OR

Including a referral ID (used by Stader Labs internally to track organics stakes).

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xcf5EA1b38380f6aF39068375516Daf40Ed70D299",

signature: **"deposit(address,string)"**,

argument[0]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

valueETH: True

}

2. Unstake ETH

The unstaked ETH will not be immediately added to your wallet as there is an unstaking period of 7-10 days, subject to entry and exit queues on the Ethereum network:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x9F0491B32DBce587c50c4C43AB303b06478193A7",

signature: **"requestWithdraw(uint256,address)"**,

argument[1]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

3. Claim

All your active unstake requests will be listed in the Withdraw tab. Those available to be withdrawn will have an active withdraw button:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x9F0491B32DBce587c50c4C43AB303b06478193A7",

signature: **"claim(uint256)"**

}

Spark

Lend ETH

1. Deposit ETH

Deposit ETH - sender specifies the amount of the underlying asset to be supplied to mint the corresponding amount of spToken:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xBD7D6a9ad7865463DE44B05F04559f65e3B11704",

signature: **"depositETH(address,address,uint16)"**,

argument[0]: "0xC13e21B648A5Ee794902342038FF3aDAB66BE987",

argument[1]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

valueETH: True

}

2. Withdraw ETH

Sender redeems a specified amount of spTokens:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xBD7D6a9ad7865463DE44B05F04559f65e3B11704",

signature: **"withdrawETH(address,uint256,address)”**,

argument[0]: "0xC13e21B648A5Ee794902342038FF3aDAB66BE987",

argument[2]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

3. Claim rewards

Sender redeems a specified amount of reward tokens:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x4370d3b6c9588e02ce9d22e684387859c7ff5b34",

signature: **"claimRewardsaddresses[] assets, uint256 amount, address to, address reward)”**,

argument[2]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

Lend WETH

1. Deposit WETH

Sender specifies the amount of the underlying asset to be supplied to mint the corresponding amount of spToken:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xC13e21B648A5Ee794902342038FF3aDAB66BE987",

signature: **"supply(address,uint256,address,uint16)"**,

argument[0]: "0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",

argument[2]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

OR

Set reserve as collateral:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xC13e21B648A5Ee794902342038FF3aDAB66BE987",

signature: **"setUserUseReserveAsCollateral(address,uint256,address,uint16)"**,

argument[0]: "0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",

argument[2]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

2. Withdraw WETH

Sender redeems a specified amount of spTokens:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xC13e21B648A5Ee794902342038FF3aDAB66BE987",

signature: **"withdraw(address,uint256,address)”**,

argument[0]: "0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",

argument[2]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

Uniswap

~~Swapping of tokens (COMP, CRV, CVX, LDO, rETH, sETH2, rETH2, SWISE, WETH, USDC, DAI and USDT)~~

~~1. Swap COMP for USDC - trading route is COMP→WETH→USDC.~~

~~2. Swap COMP for DAI - trading route is COMP →WETH →DAI.~~

~~3. Swap COMP for WETH - trading route is COMP→WETH.~~

~~4. Swap CRV for USDC - trading route is CRV →WETH →USDC.~~

~~5. Swap CRV for DAI- trading route is CRV→WETH →DAI.~~

~~6. Swap CRV for WETH - trading route is CRV→WETH.~~

~~7. Swap LDO for USDC - trading route is LDO →WETH →USDC.~~

~~8. Swap LDO for DAI- trading route is LDO→WETH →DAI.~~

~~9. Swap LDO for WETH - trading route is LDO →WETH.~~

~~10.Swap WETH for USDC - trading route is WETH →USDC.~~

~~11.Swap WETH for DAI - trading route is WETH→DAI.~~

~~12.Swap WETH for USDT - trading route is WETH →USDT.~~

~~13.Swap USDC for WETH - trading route is USDC→WETH.~~

~~14.Swap USDC for DAI - trading route is USDC →WETH →DAI.~~

15.Swap USDC for DAI - trading route is USDC→DAI.

16.Swap USDC for USDT - trading route is USDC →WETH →USDT.

17.Swap USDC for USDT - trading route is USDC→USDT.

18.Swap DAI for WETH - trading route is DAI→WETH.

19.Swap DAI for USDC - trading route is DAI →WETH →USDC.

20.Swap DAI for USDC - trading route is DAI→USDC.

21.Swap DAI for USDT - trading route is DAI →WETH →USDT.

22.Swap DAI for USDT - trading route is DAI →USDT.

23.Swap USDT for WETH - trading route is USDT→WETH.

24.Swap USDT for USDC - trading route is USDT →WETH →USDC.

25.Swap USDT for USDC - trading route is USDT →USDC.

26.Swap USDT for DAI - trading route is USDT →WETH →DAI.

27.Swap USDT for DAI - trading route is USDT→DAI.

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x68b3465833fb72A70ecDF485E0e4C7bD8665Fc45",

signature: "**swapExactTokensForTokens(uint256,uint256,address[],address)**",

argument[2]: [

["0xc00e94Cb662C3520282E6f5717214004A7f26888",

~~"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",~~

~~"0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48"],~~

~~OR~~

~~["0xc00e94Cb662C3520282E6f5717214004A7f26888",~~

~~"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",~~

~~"0x6B175474E89094C44Da98b954EedeAC495271d0F"],~~

~~OR~~

~~["0xc00e94Cb662C3520282E6f5717214004A7f26888",~~

~~"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2"],~~

~~OR~~

~~["0xD533a949740bb3306d119CC777fa900bA034cd52",~~

~~"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",~~

~~"0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48"],~~

~~OR~~

~~["0xD533a949740bb3306d119CC777fa900bA034cd52",~~

~~"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",~~

~~"0x6B175474E89094C44Da98b954EedeAC495271d0F"],~~

~~OR~~

~~["0xD533a949740bb3306d119CC777fa900bA034cd52",~~

~~"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2"],~~

~~OR~~

~~["0x5A98FcBEA516Cf06857215779Fd812CA3beF1B32",~~

~~"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",~~

~~"0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48"],~~

~~OR~~

~~["0x5A98FcBEA516Cf06857215779Fd812CA3beF1B32",~~

~~"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",~~

~~"0x6B175474E89094C44Da98b954EedeAC495271d0F"],~~

~~OR~~

~~["0x5A98FcBEA516Cf06857215779Fd812CA3beF1B32",~~

~~"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2"],~~

~~["0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",~~

~~"0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48"],~~

~~OR~~

~~["0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",~~

~~"0x6B175474E89094C44Da98b954EedeAC495271d0F"],~~

~~OR~~

~~["0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",~~

~~"0xdAC17F958D2ee523a2206206994597C13D831ec7"],~~

~~OR~~

~~["0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48",~~

~~"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2"],~~

~~OR~~

~~["0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48",~~

~~"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",~~

~~"0x6B175474E89094C44Da98b954EedeAC495271d0F"],~~

~~OR~~

~~["0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48",~~

~~"0x6B175474E89094C44Da98b954EedeAC495271d0F"],~~

~~OR~~

~~["0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48",~~

~~"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",~~

~~"0xdAC17F958D2ee523a2206206994597C13D831ec7"],~~

~~OR~~

~~["0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48",~~

~~"0xdAC17F958D2ee523a2206206994597C13D831ec7"],~~

~~OR~~

~~["0x6B175474E89094C44Da98b954EedeAC495271d0F",~~

~~"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2"],~~

~~OR~~

~~["0x6B175474E89094C44Da98b954EedeAC495271d0F",~~

~~"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",~~

~~"0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48"],~~

~~OR~~

~~["0x6B175474E89094C44Da98b954EedeAC495271d0F",~~

~~"0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48"],~~

~~OR~~

~~["0x6B175474E89094C44Da98b954EedeAC495271d0F",~~

~~"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",~~

~~"0xdAC17F958D2ee523a2206206994597C13D831ec7"],~~

~~OR~~

~~["0x6B175474E89094C44Da98b954EedeAC495271d0F",~~

~~"0xdAC17F958D2ee523a2206206994597C13D831ec7"],~~

~~OR~~

~~["0xdAC17F958D2ee523a2206206994597C13D831ec7",~~

~~"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2"],~~

~~OR~~

~~["0xdAC17F958D2ee523a2206206994597C13D831ec7",~~

~~"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",~~

~~"0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48"],~~

~~OR~~

~~["0xdAC17F958D2ee523a2206206994597C13D831ec7",~~

~~"0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48"],~~

~~OR~~

~~["0xdAC17F958D2ee523a2206206994597C13D831ec7",~~

~~"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",~~

~~"0x6B175474E89094C44Da98b954EedeAC495271d0F"],~~

~~OR~~

~~["0xdAC17F958D2ee523a2206206994597C13D831ec7",~~

~~"0x6B175474E89094C44Da98b954EedeAC495271d0F"],~~

~~argument[3]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"~~

~~}~~

- **Token In List**: [CRV, DAI, USDT, BAL, AURA, CVX, ETHx, COMP, rETH, SWISE, wstETH, LDO, WETH, ankrETH, USDC, stETH, sETH2, rETH2]
- **Token Out List**: [DAI, USDT, rETH, wstETH, WETH, USDC, stETH, sETH2]

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x68b3465833fb72A70ecDF485E0e4C7bD8665Fc45",

signature: "**exactInputSingle((address,address,uint24,address,uint256,uint256,uint160))**",

argument[0]: ["0xD533a949740bb3306d119CC777fa900bA034cd52",

OR

"0x6B175474E89094C44Da98b954EedeAC495271d0F",

OR

"0xdAC17F958D2ee523a2206206994597C13D831ec7",

OR

"0xba100000625a3754423978a60c9317c58a424e3D",

OR

"0xC0c293ce456fF0ED870ADd98a0828Dd4d2903DBF",

OR

"0x4e3FBD56CD56c3e72c1403e103b45Db9da5B9D2B",

OR

"0xA35b1B31Ce002FBF2058D22F30f95D405200A15b",

OR

"0xc00e94Cb662C3520282E6f5717214004A7f26888",

OR

"0xae78736Cd615f374D3085123A210448E74Fc6393",

OR

"0x48C3399719B582dD63eB5AADf12A40B4C3f52FA2",

OR

"0x7f39C581F595B53c5cb19bD0b3f8dA6c935E2Ca0",

OR

"0x5A98FcBEA516Cf06857215779Fd812CA3beF1B32",

OR

"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",

OR

"0xE95A203B1a91a908F9B9CE46459d101078c2c3cb",

OR

"0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48",

OR

"0xae7ab96520DE3A18E5e111B5EaAb095312D7fE84",

~~OR~~

~~"0x20BC832ca081b91433ff6c17f85701B6e92486c5",~~

~~OR~~

~~"0xFe2e637202056d30016725477c5da089Ab0A043A"~~],

argument[1]: ["0x6B175474E89094C44Da98b954EedeAC495271d0F",

OR

“0xdAC17F958D2ee523a2206206994597C13D831ec7”,

OR

“0xae78736Cd615f374D3085123A210448E74Fc6393”,

OR

“0x7f39C581F595B53c5cb19bD0b3f8dA6c935E2Ca0”,

OR

“0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2”,

OR

“0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48”,

OR

“0xae7ab96520DE3A18E5e111B5EaAb095312D7fE84”,

~~OR~~

~~"0x20BC832ca081b91433ff6c17f85701B6e92486c5",~~

~~OR~~

~~"0xFe2e637202056d30016725477c5da089Ab0A043A"~~],

argument[3]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64”

}

PancakeSwap

Swapping of tokens (ETHx and WETH)

- **Token In List**: [ETHx, WETH]
- **Token Out List**: [ETHx, WETH]

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0x13f4EA83D0bd40E75C8222255bc855a974568Dd4",

Signature: **"exactInputSingle((address,address,uint24,address,uint256,uint256,uint160))"**,

argument[0]: ["0xA35b1B31Ce002FBF2058D22F30f95D405200A15b",

"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2"],

argument[1]: ["0xA35b1B31Ce002FBF2058D22F30f95D405200A15b",

"0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2"],

argument[3]: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64"

}

Wrapping and unwrapping of ETH

1. Wrap ETH

Wraps ETH, sender receives WETH:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",

signature: "**deposit()**"

}

2. Unwrap WETH

Unwraps WETH, sender receives ETH:

{

from: "0x4F2083f5fBede34C2714aFfb3105539775f7FE64",

to: "0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2",

signature: "**withdraw(uint256)**"

CowSwap

1. Swap tokens ()

Generates a signed order to the CowswapOrderSigner to swap from any “sellToken” to any buyToken”:

- Token In List (“sellToken”): [CRV, DAI, USDT, BAL, AURA, CVX, ETHx, COMP, rETH, SWISE, wstETH, LDO, WETH, ankrETH, USDC, stETH]
- Token Out List (“buyToken”): [DAI, USDT, rETH, wstETH, WETH, USDC, stETH]

{

from: "0x0EFcCBb9E2C09Ea29551879bd9Da32362b32fc89",

to: "0x23dA9AdE38E4477b23770DeD512fD37b12381FAB",

signature: **"signOrder(tuple order, uint32 validDuration, uint256 feeAmountBP)"**,

Order: sellToken: [“0xD533a949740bb3306d119CC777fa900bA034cd52”,

OR

“0x6B175474E89094C44Da98b954EedeAC495271d0F”,

OR

“0xdAC17F958D2ee523a2206206994597C13D831ec7”,

OR

“0xba100000625a3754423978a60c9317c58a424e3D”,

OR

“0xC0c293ce456fF0ED870ADd98a0828Dd4d2903DBF”,

OR

“0x4e3FBD56CD56c3e72c1403e103b45Db9da5B9D2B”,

OR

“0xA35b1B31Ce002FBF2058D22F30f95D405200A15b”,

OR

“0xc00e94Cb662C3520282E6f5717214004A7f26888”,

OR

“0xae78736Cd615f374D3085123A210448E74Fc6393”,

OR

“0x48C3399719B582dD63eB5AADf12A40B4C3f52FA2”,

OR

“0x7f39C581F595B53c5cb19bD0b3f8dA6c935E2Ca0”,

OR

“0x5A98FcBEA516Cf06857215779Fd812CA3beF1B32”,

OR

“0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2”,

OR

“0xE95A203B1a91a908F9B9CE46459d101078c2c3cb”,

OR

“0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48”,

OR

“0xae7ab96520DE3A18E5e111B5EaAb095312D7fE84”]

Order: buyToken:
[“0x6B175474E89094C44Da98b954EedeAC495271d0F”,

OR

“0xdAC17F958D2ee523a2206206994597C13D831ec7”,

OR

“0xae78736Cd615f374D3085123A210448E74Fc6393”,

OR

“0x7f39C581F595B53c5cb19bD0b3f8dA6c935E2Ca0”,

OR

“0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2”,

OR

“0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48”,

OR

“0xae7ab96520DE3A18E5e111B5EaAb095312D7fE84”]

}