Interaction receiver

Limit order protocol has the ability to call anintermediate contract when an order is filled.

Intermediate contract should implement an interface:InteractiveNotificationReceiver.sol

Use case

For example, we want to create a limit order1INCH -WETH , but so that when this order is filled, we will not receive WETH but ETH.

To do this, let's create asmart contract :

contract

WethUnwrapper

is InteractiveNotificationReceiver { // solhint-disable-next-line no-empty-blocks receive ( )

external

payable

{ }

function

notifyFillOrder (address

/ *taker* / , address

/ *makerAsset* / , address takerAsset , uint256

/ *makingAmount* / , uint256 takingAmount , bytes

calldata interactiveData)

external override { address

payable makerAddress ; // solhint-disable-next-line no-inline-assembly assembly

{ makerAddress :=

shr (96 ,

calldataload (interactiveData . offset)) } IWithdrawable (takerAsset) . withdraw (takingAmount) ; makerAddress . transfer (takingAmount) ; } } And create a limit order withpreInteraction and/orpostInteraction field:

const interactiveReceiverAddress =

'0x1282d0c06368c40c8d4a4d818d78f258d982437b' ; const walletAddress =

'0xfb3c7ebccccAA12B5A884d612393969Adddddddd' ;

const limitOrder = limitOrderBuilder . buildLimitOrder ({ makerAssetAddress :

'0xbb4cdb9cbd36b01bd1cbaebf2de08d9173bc095c' , takerAssetAddress :

'0x111111111117dc0aa78b770fa6a738034120c302' , makerAddress : walletAddress , reciever : interactiveReceiverAddress , makerAmount :

'100' , takerAmount :

'200' ,

// What to do before the transfer preInteraction :

'0x' , // What tot do after the trasnfer has been made postInteraction : interactiveReceiverAddress + walletAddress . slice (2) , }) ; As you can see, thepostInteraction field consists of two parts - thecontract address and theinteractiveData (which contains the wallet address). We also set theinteractiveReceiverAddress as thereciever . This is necessary in order for this contract to receive aWETH when filling out an limit order and returnETH to the creator of the limit order. Now the limit order

will work according to the following scenario:

- someone (taker
-) will create a transaction to execute a limit order
- taker
- funds (WETH
-) will be transferred to theinteractiveReceiver
- contract
- interactiveReceiver
- contract will send the same amount ofETH
- to themaker
- and finally, themaker asset
- (1INCH
-) will be transferred from themaker's
- address to thetaker's
- address

Example of transaction

However, this is one example of the use of interactivity. You can implement any other scenarioEdit this page Previous Validate a limit order Next Domain separator