Example Codes

I'll define a few pseudo-code functions and constants to make snippets more concise.

Some are also defined in thelib/Token.sol

```

Copy // Let's define a function for you to see more comfortably
functiontoTokenInfo(bytes1_tokenRefIndex,uint8_method,int128_amount)purereturns(bytes32) {
returnbytes32(bytes1(tokenRefIndex))|bytes32(bytes2(uint16(_method)))|bytes32(uint256(uint128(uint256(int256(_amount)))));; }
functiontoPoolId(uint8_optype,address_pool)purereturns(bytes32){ returnbytes32(bytes1(_optype))|bytes32(uint256(uint160(address(_pool)))); }
functiontoToken(IERC20tok)purereturns(Token) { returnToken.wrap(bytes32(uint256(uint160(address(tok))))); }
functiontoToken(TokenSpecTypespec,uint88id_,addressaddr_)purereturns(Token) { returnToken.wrap(
TokenSpecType.unwrap(spec_)|bytes32((bytes32(uint256(id_))<<ID_SHIFT)&ID_MASK) |bytes32(uint256(uint160(addr_))) ); } // use
NATIVE_TOKEN for ETH since we don't use weth internally
TokenconstantNATIVE_TOKEN=Token.wrap(0xEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE);//ref:
lib/Token.sol uint8constantSWAP=0; uint8constantGAUGE=1; uint8constantCONVERT=2; uint8constantVOTE=3; uint8constantUSERBALANCE=4;

uint8constantEXACTLY=0; uint8constantAT_MOST=1; uint8constantALL=2;

```

The helper functions above will help you populate the Token/poolId/TokenInfo struct.

These 3 components are all you need for execute!

Swap

Add liquidity

Stake

Voting

Multicall operations

cf) wrapper functions

Last updated4 months ago On this page