ClipperRouter

Clipper router that allows to use `ClipperExchangeInterface` for swaps

Quick links

Derives

Functions

Derives

- EthReceiver

Functions

constructor

```
function constructor(contract IWETH weth)
```

Parameters:

Name	Type	Description
`weth`	contract IWETH	

clipperSwapToWithPermit

```
function clipperSwapToWithPermit(
    contract IClipperExchangeInterface clipperExchange,
    address payable recipient,
    contract IERC20 srcToken,
    contract IERC20 dstToken,
    uint256 inputAmount,
    uint256 outputAmount,
    uint256 expiryWithFlags,
    bytes32 r,
    bytes32 vs,
    bytes calldata permit
) external returns(uint256 returnAmount)
```

Same as `clipperSwapTo` but calls permit first, allowing to approve token spending and make a swap in one transaction.

Parameters:

Name	Type	Description
`clipperExchange`	contract IClipperExchangeInterface	Clipper pool address
`recipient`	address payable	Address that will receive swap funds
`srcToken`	contract IERC20	Source token
`dstToken`	contract IERC20	Destination token
`inputAmount`	uint256	Amount of source tokens to swap
`outputAmount`	uint256	Amount of destination tokens to receive
`expiryWithFlags`	uint256	Timestamp until the swap will be valid with permit2 flag
`r`	bytes32	Clipper order signature (r part)
`vs`	bytes32	Clipper order signature (vs part)
`permit`	bytes	Should contain valid permit that can be used in `IERC20Permit.permit` calls. See tests for examples

Return values

Name	Type	Description
`returnAmount`	uint256	Amount of destination tokens received

clipperSwap

```
    function clipperSwap(
        contract IClipperExchangeInterface clipperExchange,
        contract IERC20 srcToken,
        contract IERC20 dstToken,
        uint256 inputAmount,
        uint256 outputAmount,
        uint256 expiryWithFlags,
        bytes32 r,
        bytes32 vs
    ) external payable returns(uint256 returnAmount)
```

Same as `clipperSwapTo` but uses `msg.sender` as recipient

Parameters:

Name	Type	Description
`clipperExchange`	contract IClipperExchangeInterface	Clipper pool address
`srcToken`	contract IERC20	Source token
`dstToken`	contract IERC20	Destination token
`inputAmount`	uint256	Amount of source tokens to swap
`outputAmount`	uint256	Amount of destination tokens to receive
`expiryWithFlags`	uint256	Timestamp until the swap will be valid with permit2 flag
`r`	bytes32	Clipper order signature (r part)
`vs`	bytes32	Clipper order signature (vs part)

Return values

Name	Type	Description
`returnAmount`	uint256	Amount of destination tokens received

clipperSwapTo

```
function clipperSwapTo(
    contract IClipperExchangeInterface clipperExchange,
    address payable recipient,
    contract IERC20 srcToken,
    contract IERC20 dstToken,
    uint256 inputAmount,
    uint256 outputAmount,
    uint256 expiryWithFlags,
    bytes32 r,
    bytes32 vs
) public payable returns(uint256 returnAmount)
```

Performs swap using Clipper exchange. Wraps and unwraps ETH if required. Sending non-zero `msg.value` for anything but ETH swaps is prohibited

Parameters:

Name	Type	Description
`clipperExchange`	contract IClipperExchangeInterface	Clipper pool address

Name	Type	Description
`recipient`	address payable	Address that will receive swap funds
`srcToken`	contract IERC20	Source token
`dstToken`	contract IERC20	Destination token
`inputAmount`	uint256	Amount of source tokens to swap
`outputAmount`	uint256	Amount of destination tokens to receive
`expiryWithFlags`	uint256	Timestamp until the swap will be valid with permit2 flag
`r`	bytes32	Clipper order signature (r part)
`vs`	bytes32	Clipper order signature (vs part)

Return values

Name	Type	Description
`returnAmount`	uint256	Amount of destination tokens received

Previous
‹ AggregationRouterV6

Next
GenericRouter ›