Router

wETH

// Returns address of the wETH contract. function

wETH ()

external

view

returns

(address);

factory

// Returns address of the factory contract. function

factory ()

external

view

returns

(address);

addLiquidity

// Adds (or creates) liquidity to pool. function

addLiquidity (address tokenA , address tokenB , uint amountADesired , uint amountBDesired , uint amountAMin , uint amountBMin , address to , uint deadline , uint feeType , bool stable)

external

returns

(uint amountA ,

uint amountB ,

uint liquidity);

addLiquidityETH

// Adds (or creates) liquidity to pool. function

addLiquidityETH (address token , uint amountTokenDesired , uint amountTokenMin , uint amountETHMin , address to , uint deadline , uint feeType , bool stable)

external

payable

returns

(uint amountToken ,

uint amountETH ,

uint liquidity);

removeLiquidity

```
// Removes liquidity from existing pool. function

removeLiquidity ( address tokenA , address tokenB , uint liquidity , uint amountAMin , uint amountBMin , address to , uint deadline , bool stable )

external

returns

( uint amountA ,

uint amountB );
```

removeLiquidityETH

```
// Removes liquidity from existing pool. function

removeLiquidityETH ( address token , uint liquidity , uint amountTokenMin , uint amountETHMin , address to , uint deadline , bool stable )

external

returns

( uint amountToken ,

uint amountETH );
```

removeLiquidityETHSupportingFeeOnTransferTokens

```
// Removes liquidity from existing pool for tokens with fee transfer. function

removeLiquidityETHSupportingFeeOnTransferTokens ( address token , uint liquidity , uint amountTokenMin , uint amountETHMin , address to , uint deadline , bool stable )

external

returns

( uint amountToken ,

uint amountETH );
```

swapExactTokensForTokens

```
// function

swapExactTokensForTokens ( uint amountIn , uint amountOutMin , address []

calldata path , address to , uint deadline , bool []

calldata stable )

external

returns

( uint []

memory amounts );
```

swapExactETHForTokens

```
// function

swapExactETHForTokens ( uint amountOutMin , address []

calldata path , address to , uint deadline , bool []

calldata stable )
```

external

payable

returns

(uint []

memory amounts);

swapExactTokensForETH

// function

swapExactTokensForETH (uint amountIn , uint amountOutMin , address []

calldata path , address to , uint deadline , bool []

calldata stable)

external

returns

(uint []

memory amounts);

swapExactETHForTokensSupportingFeeOnTransferTokens

// function

swapExactETHForTokensSupportingFeeOnTransferTokens (uint amountOutMin , address []

calldata path , address to , uint deadline , bool []

calldata stable)

external

payable ;

swapExactTokensForETHSupportingFeeOnTransferTokens

// function

swapExactTokensForETHSupportingFeeOnTransferTokens (uint amountIn , uint amountOutMin , address []

calldata path , address to , uint deadline , bool []

calldata stable)

external ;

swapExactTokensForTokensSupportingFeeOnTransferTokens

// function

swapExactTokensForTokensSupportingFeeOnTransferTokens (uint amountIn , uint amountOutMin , address []

calldata path , address to , uint deadline , bool []

calldata stable)

external ;

quote

// Gets reserve quotes function

quote (uint amountA ,

uint reserveA ,

uint reserveB) external

pure

returns

(uint amountB);

getAmountOut

// Calculates the best price through both stable and normal pools function

getAmountOut (uint amountIn , uint tokenIn , uint tokenOut)

external

view

returns

(uint amountOut ,

bool stable ,

uint fee);

getAmountsOutExpanded

// Calculates the best prices for path. Returns amounts, pool types, and pool fees for route function

getAmountsOutExpanded (uint amountIn , address []

calldata path)

external

view

returns

(uint []

memory amounts ,

bool []

memory stable ,

uint []

memory fees);

getAmountsOut

// Calculates price best on path and pool type (stable, normal) function

getAmountsOut (uint amountIn , address []

calldata path , bool []

calldata stable)

external

view

returns

```
( uint []

memory amounts );
```

getPairInfo

```
// Returns pair info function

getPairInfo ( address []

calldata path , bool stable )

external

view

returns ( address tokenA ,

address tokenB ,

address pair ,

uint reserveA ,

uint reserveB ,

uint fee )
```

pairFor

```
// Returns pair address function

pairFor ( address tokenA ,

address tokenB ,

bool stable ) external

view

returns ( address pair );
```

Koi Contracts - Previous ABIs Next - Koi Contracts Audits

Last modified11mo ago Was this page helpful?