

Examples

Single token-to-ETH price usage

```
const
Web3
=
require ( 'web3' ) ; const
{
  BigNumber
}
=
require ( 'ethers' ) ;
const yourInfuraKey =
'add your key here' ; const web3 =
new
Web3 ( https://mainnet.infura.io/v3/ { yourInfuraKey } ) ;
// eslint-disable-next-line max-len const
OffChainOracleAbi
=
[{"inputs":[{"internalType":"contract MultiWrapper","name":"_multiWrapper","type":"address"},{"internalType":"contract IOracle[]","name":"existingOracles","type":"address[]"},{"internalType":"enum OffchainOracle.OracleType[]","name":"oracleTypes","type":"uint8[]"},{"internalType":"contract IERC20[]","name":"existingConnectors","type":"address[]"},{"internalType":"contract IERC20","name":"wBase","type":"address"}],"stateMutability":"nonpayable","type":"constructor"},{"anonymous":false,"inputs":[{"indexed":false,"internalType":"contract IERC20","name":"connector","type":"address"}],"name":"ConnectorAdded","type":"event"},{"anonymous":false,"inputs":[{"indexed":false,"internalType":"contract IERC20","name":"connector","type":"address"}],"name":"ConnectorRemoved","type":"event"},{"anonymous":false,"inputs":[{"indexed":false,"internalType":"contract MultiWrapper","name":"multiWrapper","type":"address"}],"name":"MultiWrapperUpdated","type":"event"},{"anonymous":false,"inputs":[{"indexed":false,"internalType":"contract IOracle","name":"oracle","type":"address"}],"indexed":false,"internalType":"enum OffchainOracle.OracleType","name":"oracleType","type":"uint8"},"name":"OracleAdded","type":"event"},{"anonymous":false,"inputs":[{"indexed":false,"internalType":"contract IOracle","name":"oracle","type":"address"}],"indexed":false,"internalType":"enum OffchainOracle.OracleType","name":"oracleType","type":"uint8"},"name":"OracleRemoved","type":"event"},{"anonymous":false,"inputs":[{"indexed":true,"internalType":"address","name":"previousOwner","type":"address"}, {"indexed":true,"internalType":"address","name":"newOwner","type":"address"}],"name":"OwnershipTransferred","type":"event"}, {"inputs":[{"internalType":"contract IERC20","name":"connector","type":"address"}],"name":"addConnector","outputs": [],"stateMutability":"nonpayable","type":"function"}, {"inputs":[{"internalType":"contract IOracle","name":"oracle","type":"address"}],"internalType":"enum OffchainOracle.OracleType","name":"oracleKind","type":"uint8"},"name":"addOracle","outputs": [],"stateMutability":"nonpayable","type":"function"}, {"inputs": [],"name":"connectors","outputs":[{"internalType":"contract IERC20[]","name":"allConnectors","type":"address[]"}],"stateMutability":"view","type":"function"}, {"inputs": [{"internalType":"contract IERC20","name":"srcToken","type":"address"}],"internalType":"contract IERC20","name":"dstToken","type":"address"}, {"internalType":"bool","name":"useWrappers","type":"bool"},"name":"getRate","outputs": [{"internalType":"uint256","name":"weightedRate","type":"uint256"}],"stateMutability":"view","type":"function"}, {"inputs": [{"internalType":"contract IERC20","name":"srcToken","type":"address"}],"internalType":"bool","name":"useSrcWrappers","type":"bool"},"name":"getRateToEth","outputs": [{"internalType":"uint256","name":"weightedRate","type":"uint256"}],"stateMutability":"view","type":"function"}, {"inputs": [],"name":"multiWrapper","outputs":[{"internalType":"contract MultiWrapper","name":"","type":"address"}],"stateMutability":"view","type":"function"}, {"inputs": [],"name":"oracles","outputs":
```

```

[{"internalType":"contract IOracle[]","name":"allOracles","type":"address[]"}, {"internalType":"enum
OffchainOracle.OracleType[]","name":"oracleTypes","type":"uint8[]"},"stateMutability":"view","type":"function"}, {"inputs":
[], "name":"owner", "outputs":[{"internalType":"address", "name":""," "type":"address"}], "stateMutability":"view", "type":"function"},
{"inputs":[{"internalType":"contract IERC20", "name":"connector", "type":"address"}], "name":"removeConnector", "outputs":
[], "stateMutability":"nonpayable", "type":"function"}, {"inputs":[{"internalType":"contract
IOracle", "name":"oracle", "type":"address"}], "internalType":"enum
OffchainOracle.OracleType", "name":"oracleKind", "type":"uint8[]"}, {"name":"removeOracle", "outputs":
[], "stateMutability":"nonpayable", "type":"function"}, {"inputs":[], "name":"renounceOwnership", "outputs":
[], "stateMutability":"nonpayable", "type":"function"}, {"inputs":[{"internalType":"contract
MultiWrapper", "name":"_multiWrapper", "type":"address"}], "name":"setMultiWrapper", "outputs":
[], "stateMutability":"nonpayable", "type":"function"}, {"inputs":
[{"internalType":"address", "name":"newOwner", "type":"address"}], "name":"transferOwnership", "outputs":
[], "stateMutability":"nonpayable", "type":"function"}] ; const offChainOracleAddress =

'0x07D91f5fb9Bf7798734C3f606dB065549F6893bb' ; const offChainOracleContract =

new

web3 . eth . Contract ( JSON . parse ( OffChainOracleAbi ) , offChainOracleAddress ) ;

const token =

{ address :

'0xdac17f958d2ee523a2206206994597c13d831ec7' ,

// USDT decimals :

6 , } ;

offChainOracleContract . methods . getRateToEth ( token . address ,

// source token true ,

// use source wrappers ) . call ( ) . then ( ( rate )

=>

{ const numerator =

BigNumber . from ( 10 ) . pow ( token . decimals ) ; const denominator =

BigNumber . from ( 10 ) . pow ( 18 ) ;

// eth decimals const price =

BigNumber . from ( rate ) . mul ( numerator ) . div ( denominator ) ; console . log ( price . toString ( ) ) ;

// 472685293218315 } ) . catch ( console . log ) ;

```

Multiple token-to-ETH prices usage

```

const

Web3

=

require ( 'web3' ) ; const

{

BigNumber

}

=

require ( 'ethers' ) ;

const yourInfuraKey =

'add your key here' ; const web3 =

```

new

Web3 ([https://mainnet.infura.io/v3/ { yourInfuraKey }](https://mainnet.infura.io/v3/{ yourInfuraKey })) ;

// eslint-disable-next-line max-len const

MultiCallAbi

=

```
{["inputs":{"components":[{"internalType":"address","name":"to","type":"address"},
{"internalType":"bytes","name":"data","type":"bytes"}],"internalType":"struct
MultiCall.Call[]","name":"calls","type":"tuple[]"},"name":"multicall","outputs":
[{"internalType":"bytes[]","name":"results","type":"bytes[]"},
{"internalType":"bool[]","name":"success","type":"bool[]"},"stateMutability":"view","type":"function"}] ; // eslint-disable-next-line
max-len const
```

OffChainOracleAbi

=

```
{["inputs":{"internalType":"contract MultiWrapper","name":"_multiWrapper","type":"address"},{"internalType":"contract
IOracle[]","name":"existingOracles","type":"address[]"},{"internalType":"enum
OffchainOracle.OracleType[]","name":"oracleTypes","type":"uint8[]"},{"internalType":"contract
IERC20[]","name":"existingConnectors","type":"address[]"},{"internalType":"contract
IERC20","name":"wBase","type":"address"},"stateMutability":"nonpayable","type":"constructor"},{"anonymous":false,"inputs":
[{"indexed":false,"internalType":"contract
IERC20","name":"connector","type":"address"},"name":"ConnectorAdded","type":"event"},{"anonymous":false,"inputs":
[{"indexed":false,"internalType":"contract
IERC20","name":"connector","type":"address"},"name":"ConnectorRemoved","type":"event"},{"anonymous":false,"inputs":
[{"indexed":false,"internalType":"contract
MultiWrapper","name":"multiWrapper","type":"address"},"name":"MultiWrapperUpdated","type":"event"},
{"anonymous":false,"inputs":[{"indexed":false,"internalType":"contract IOracle","name":"oracle","type":"address"},
{"indexed":false,"internalType":"enum
OffchainOracle.OracleType","name":"oracleType","type":"uint8"}],"name":"OracleAdded","type":"event"},
{"anonymous":false,"inputs":[{"indexed":false,"internalType":"contract IOracle","name":"oracle","type":"address"},
{"indexed":false,"internalType":"enum
OffchainOracle.OracleType","name":"oracleType","type":"uint8"}],"name":"OracleRemoved","type":"event"},
{"anonymous":false,"inputs":[{"indexed":true,"internalType":"address","name":"previousOwner","type":"address"},
{"indexed":true,"internalType":"address","name":"newOwner","type":"address"},"name":"OwnershipTransferred","type":"event"},
{"inputs":[{"internalType":"contract IERC20","name":"connector","type":"address"},"name":"addConnector","outputs":
[],"stateMutability":"nonpayable","type":"function"},{"inputs":[{"internalType":"contract
IOracle","name":"oracle","type":"address"},{"internalType":"enum
OffchainOracle.OracleType","name":"oracleKind","type":"uint8"}],"name":"addOracle","outputs":
[],"stateMutability":"nonpayable","type":"function"},{"inputs":[],"name":"connectors","outputs":[{"internalType":"contract
IERC20[]","name":"allConnectors","type":"address[]"},"stateMutability":"view","type":"function"},{"inputs":
[{"internalType":"contract IERC20","name":"srcToken","type":"address"},{"internalType":"contract
IERC20","name":"dstToken","type":"address"},
{"internalType":"bool","name":"useWrappers","type":"bool"}],"name":"getRate","outputs":
[{"internalType":"uint256","name":"weightedRate","type":"uint256"},"stateMutability":"view","type":"function"},{"inputs":
[{"internalType":"contract IERC20","name":"srcToken","type":"address"},
{"internalType":"bool","name":"useSrcWrappers","type":"bool"}],"name":"getRateToEth","outputs":
[{"internalType":"uint256","name":"weightedRate","type":"uint256"},"stateMutability":"view","type":"function"},{"inputs":
[],"name":"multiWrapper","outputs":[{"internalType":"contract
MultiWrapper","name":"","type":"address"},"stateMutability":"view","type":"function"},{"inputs":[],"name":"oracles","outputs":
[{"internalType":"contract IOracle[]","name":"allOracles","type":"address[]"},{"internalType":"enum
OffchainOracle.OracleType[]","name":"oracleTypes","type":"uint8[]"},"stateMutability":"view","type":"function"},{"inputs":
[],"name":"owner","outputs":[{"internalType":"address","name":"","type":"address"},"stateMutability":"view","type":"function"},
{"inputs":[{"internalType":"contract IERC20","name":"connector","type":"address"},"name":"removeConnector","outputs":
[],"stateMutability":"nonpayable","type":"function"},{"inputs":[{"internalType":"contract
IOracle","name":"oracle","type":"address"},{"internalType":"enum
OffchainOracle.OracleType","name":"oracleKind","type":"uint8"}],"name":"removeOracle","outputs":
[],"stateMutability":"nonpayable","type":"function"},{"inputs":[],"name":"renounceOwnership","outputs":
[],"stateMutability":"nonpayable","type":"function"},{"inputs":[{"internalType":"contract
MultiWrapper","name":"_multiWrapper","type":"address"},"name":"setMultiWrapper","outputs":
[],"stateMutability":"nonpayable","type":"function"},{"inputs":
[{"internalType":"address","name":"newOwner","type":"address"},"name":"transferOwnership","outputs":
[],"stateMutability":"nonpayable","type":"function"}] ;
```

const offChainOracleAddress =

```

'0x07D91f5fb9Bf7798734C3f606dB065549F6893bb' ;

const multiCallContract =

new

web3 . eth . Contract ( JSON . parse ( MultiCallAbi ) ,

'0xda3c19c6fe954576707fa24695efb830d9cca1ca' ) ; const offChainOracleContract =

new

web3 . eth . Contract ( JSON . parse ( OffChainOracleAbi ) ) ;

const tokens =

[ { address :

'0x6b175474e89094c44da98b954eedeac495271d0f' ,

// DAI decimals :

18 , } , { address :

'0xa0b86991c6218b36c1d19d4a2e9eb0ce3606eb48' ,

// USDC decimals :

6 , } , { address :

'0xdac17f958d2ee523a2206206994597c13d831ec7' ,

// USDT decimals :

6 , } ,

{ address :

'0x11111111117dc0aa78b770fa6a738034120c302' ,

// 1INCH decimals :

18 , } , ] ;

const callData = tokens . map ( ( token )

=>

( { to : offChainOracleAddress , data : offChainOracleContract . methods . getRateToEth ( token . address , true ,

// use wrapper ) . encodeABI ( ) , } ) ) ;

multiCallContract . methods . multicall ( callData ) . call ( ) . then ( ( { results , success , } )

=>

{ const prices =

{} ; for

( let i =

0 ; i < results . length ; i ++ )

{ if

( ! success [ i ] )

{ continue ; }

const decodedRate = web3 . eth . abi . decodeParameter ( 'uint256' , results [ i ] ) . toString ( ) ; const numerator =

BigNumber . from ( 10 ) . pow ( tokens [ i ] . decimals ) ; const denominator =

BigNumber . from ( 10 ) . pow ( 18 ) ;

```

```
// eth decimals const price =
```

```
BigNumber . from ( decodedRate ) . mul ( numerator ) . div ( denominator ) ; prices [ tokens [ i ] . address ]
```

```
= price . toString ( ) ; } console . log ( prices ) ; { '0x6b175474e89094c44da98b954eedeac495271d0f': '527560209915550',  
'0xa0b86991c6218b36c1d19d4a2e9eb0ce3606eb48': '507746821617073',  
'0xdac17f958d2ee523a2206206994597c13d831ec7': '529527134930000',  
'0x111111111117dc0aa78b770fa6a738034120c302': '1048752594621361' } / } ) . catch ( console . log ) Edit this page  
Previous Introduction Next Oracle Deployment Guide
```