Filling a RFQ order

A RFQ order can be filled in whole or in part.

Important! You can fill in a RFQ order only once!

Parameters:

Field Description order structure of the RFQ order (see RFQ order structure) signature signature of the typed date of the RFQ order (signTypedData_v4) makerAmount the number of maker asset tokens that you want to fill (in token units). For example: 5 DAI = 5000000000000000000 units takerAmount the number of taker asset tokens that you want to fill (in token units). For example: 5 DAI = 5000000000000000000 units important! Only one of the assets amounts can be not zero. For example, if you specified the maker amount, then the taker amount must be zero and vice versa.

Filling with a typescript/javascript:

```
import Web3 from

'web3' ; import

{ LimitOrderBuilder , LimitOrderProtocolFacade , RFQOrder , Web3ProviderConnector }

from

'@1inch/limit-order-protocol-utils' ;

const contractAddress =

'0x7643b8c2457c1f36dc6e3b8f8e112fdf6da7698a' ; const walletAddress =

'0xd337163ef588f2ee7cdd30a3387660019be415c9' ;

const web3 =

new

Web3 ( '...' ) ; // You can create and use a custom provider connector (for example: ethers) const connector =

new

Web3ProviderConnector ( web3 ) ;

const limitOrderBuilder =

new

LimitOrderBuilder ( contractAddress , chainId , connector ) ;

const limitOrderProtocolFacade =

new

LimitOrderProtocolFacade ( contractAddress , chainId , connector , ) ;

const RFQorder : RFQOrder =

{ ... } ;

const typedData = limitOrderBuilder . buildRFQOrderTypedData ( RFQorder ) ; const signature =

await limitOrderBuilder . buildOrderSignature ( walletAddress , typedData ) ; const makerAmount =

'1000000000000000000' ; const takerAmount =

'0' ;

const callData = limitOrderProtocolFacade . fillRFQOrder ( order , signature , makerAmount , takerAmount ) ;

// Send transaction for the order RFQ filling // Must be implemented sendTransaction ( { from : walletAddress , gas :

150_000 ,
```

// Set your gas limit gasPrice :

600000000 ,

// Set your gas price to : contractAddress , data : callData , }) ;

Filling via CLI (with arguments):

gasPrice - in units of GWEI

npx limit-order-rfq-utils -- \ --operation = fill \ --chainId = 56

\ --privateKey = { xxx }

\ --gasPrice = 6

\ --order = "{ \ \" info \" : \" 29941961886664662336741887180811 \" , \ \" makerAsset \" : \" 0x111111111117dc0aa78b770fa6a738034120c302 \" , \ \" takerAsset \" : \" 0x1af3f329e8be154074d8769d1ffa4ee058b1dbc3 \" , \ \" makerAssetData \" : \" 0x23b872dd00...000 \" , \ \" takerAssetData \" : \" 0x23b872dd00...000 \" \ }"

\ --makerAmount = 1000000000000000000

\ --takerAmount = 0

Filling via CLI (through prompt):

npx limit-order-rfq-utils As result, you will receive a link to the transaction hash.Edit this page Previous Creating an RFQ order Next Canceling a limit order