

Validate a limit order

There is the possibility to check limit order validity. For example, you can check that a limit order is valid by predicates.

LimitOrderProtocolFacade.simulate

LimitOrderProtocolFacade . simulate (targetAddress :

string , calldata :

string) Under the hood:

On asimulate() call, the contract reverts witherror SimulationResults(bool success, bytes res)

Wheresuccess indicates that call doesn't revers, andres is a function call result.

LimitOrderProtocolFacade.checkPredicate

LimitOrderProtocolFacade . checkPredicate (order : LimitOrder) More lightweight version, onlypredicate field is required in order structure.

Example:

```
import Web3 from
```

```
'web3' ; import
```

```
{ LimitOrderProtocolFacade , LimitOrder , Web3ProviderConnector }
```

```
from
```

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

```
const contractAddress = limirOrderProtocolAddresses [ chainId ] ; const order : LimitOrder =
```

```
{ ... } ;
```

```
const connector =
```

```
new
```

```
Web3ProviderConnector ( new
```

```
Web3 ( '...' ) ) ; const limitOrderProtocolFacade =
```

```
new
```

```
LimitOrderProtocolFacade ( contractAddress , chainId , connector ) ;
```

```
const
```

```
{ success :
```

```
boolean , rawResult :
```

```
string , }
```

```
=
```

```
await limitOrderProtocolFacade . simulate ( contractAddress , order . predicate ) ; console . log ( 'Order validity: ' , success )  
;
```

```
const predicateValidity :
```

```
boolean
```

```
= limitOrderProtocolFacade . checkPredicate ( order ) ; console . log ( 'Predicate validity: ' , predicateValidity )
```

[Edit this page](#) [Previous](#) [Limit order remaining](#) [Next](#) [Interaction receiver](#)