

Create a limit order

LimitOrderBuilder.buildLimitOrder()

Parameters:

Field Type Description **makerAssetAddress** string the address of the asset you want to sell (address of a token contract) **takerAssetAddress** string the address of the asset you want to buy (address of a token contract) **makerAddress** string an address of the maker (wallet address) **takerAddress** string? by default contains a zero address, which means that a limit order is available for everyone to fill. If you set a value, then the limit order will be available for execution only for the specified address (private limit order) **receiver** string? by default contains a zero address, which means that taker asset will be sent to the address of the creator of the limit order. If you set a value, then taker asset will be sent to the specified address **makingAmount** string the number of maker asset tokens that you want to sell (in token units). For example: 5 DAI = 5000000000000000000 units **takingAmount** string the number of taker asset tokens that you want to receive for selling the maker asset (in token units). For example: 5 DAI = 50000000000000000000 units **predicate** string? a predicate call data. Default:0x . See [Predicate docs](#) **permit** string? a permit (EIP-2612) call data. Could be built using [utility library](#) . **default:0x** interaction string? a call data for InteractiveNotificationReceiver. See more [Interaction receiver docs](#) . Default:0x

Example:

[illegible]

Limit order signature

To fill a limit order, you need a typed data structure signature. You can create a signature following the example above.

But the example uses `Web3ProviderConnector` which is designed to work with a wallet. If you need to get the signature on the server side, you can use the `PrivateKeyProviderConnector` and get the signature using the private key.

```
const walletAddress =
'0xd337163ef588f2ee7cdd30a3387660019be415c9' ;

const privateKey = 'd8d1f95deb28949ea0ecc4e9a0decf89e98422c2d76ab6e5f736792a388c56c7' ; const limitOrderTypedData : EIP712TypedData =
{ // ... } ;

const web3Provider =
new
Web3 ( '...' ) ; const privateKeyProviderConnector =
new
PrivateKeyProviderConnector ( privateKey , web3Provider ) ;
```

```
const signature =
```

```
await privateKeyProviderConnector . signTypedData ( walletAddress , limitOrderTypedData ) Edit this page Previous Quick start Next Limit order structure
```