How to transfer tokens between accounts

Account types

Main Account

- Otherwise known as your wallet/address account.
- This account holds tokens that are sent to/from the chain, including tokens used for gas and collateral.
- Gas for transactions is used from the main account.
- Main accounts cannot trade.

Subaccount

- Subaccounts are used to trade.
- Each main account can have 128,001 subaccounts.
- Each subaccount is uniquely identified using as subaccount ID of(main account address, integer)
- Once you deposit funds to a valid subaccount ID, the subaccount will automatically be created.
- Only the main account can send transactions on behalf of a subaccount.
- Subaccounts do not require gas (no gas is used for trading).
- Subaccounts require collateral token (currently USDC) in order to trade.

Subaccount types

Cross-margin Subaccount

- Cross-margin subaccounts are able to trade positions for all cross markets.
- Cross-margin subaccounts share a single collateral pool for all positions.
- Cross-margin subaccounts are not able to trade isolated markets.
- Frontends (Web, Mobile) will use subaccount number0
- for all cross-margin trading.

Isolated Subaccount

- Isolated subaccounts are able to trade positions for single isolated market at a time.
- Isolated subaccounts are not able to trade cross markets.
- Frontends (Web, Mobile) will use subaccount numbers128 - 128_000
- for isolated market trading.

Transfer from main account to subaccounts

Thedeposit transaction must be used to perform this transfer.

Parameters (link(opens in a new tab))

Example: depositing 100 USDC into subaccount number0 .

TypeScript Python import { FaucetClient } from

"@dydxprotocol/v4-client-js" ;

const

NETWORK

= < insert_network_here

; const

USDC_AMOUNT

=

100 ; const

USDC_ASSET_ID

=

0 ; const

SUBACCOUNT_NUMBER

=

0 ;

const

wallet

=

await

LocalWallet .fromMnemonic (DYDX_MNEMONIC ,

BECH32_PREFIX); const

client

=

await

ValidatorClient .connect (NETWORK);

const

subaccount

=

new

SubaccountInfo (wallet ,

SUBACCOUNT_NUMBER);

const

quantums

=

new

Long (USDC_AMOUNT

*

1_000_000); const

tx

=

await

client . post .deposit (subaccount ,

USDC_ASSET_ID , quantums);

Transfer from subaccount to main account

Thewithdraw transaction must be used to perform this transfer.

Parameters (link(opens in a new tab))

Example: withdrawing 100 USDC from subaccount number0 .

TypeScript Python import { FaucetClient } from

"@dydxprotocol/v4-client-js" ;

```
const
NETWORK
= < insert_network_here
        ; const
USDC_AMOUNT
=
100 ; const
USDC_ASSET_ID
=
0 ; const
SUBACCOUNT_NUMBER
=
0 ;
const
wallet
=
await
LocalWallet .fromMnemonic ( DYDX_MNEMONIC ,
BECH32_PREFIX ); const
client
=
await
ValidatorClient .connect ( NETWORK );
const
subaccount
=
new
SubaccountInfo (wallet ,
SUBACCOUNT_NUMBER );
const
quantums
=
new
Long ( USDC_AMOUNT
*
1_000_000 ); const
tx
```

=

await

client . post .withdraw (subaccount ,

USDC_ASSET_ID , quantums);

Transfer from subaccount to subaccount

Thetransfer transaction must be used to perform this transfer.

Parameters (link(opens in a new tab))

Example: transferring 100 USDC from subaccount number0 to100 .

TypeScript Python import { FaucetClient } from

"@dydxprotocol/v4-client-js" ;

const

NETWORK

= < insert_network_here

; const

USDC_AMOUNT

=

100 ; const

USDC_ASSET_ID

=

0 ; const

SUBACCOUNT_NUMBER_FROM

=

0 ; const

SUBACCOUNT_NUMBER_TO

=

1 ;

const

wallet

=

await

LocalWallet .fromMnemonic (DYDX_MNEMONIC ,

BECH32_PREFIX); const

client

=

await

ValidatorClient .connect (NETWORK);

const

subaccount

=

new

SubaccountInfo (wallet ,

SUBACCOUNT_NUMBER_FROM);

const

quantums

=

new

Long (USDC_AMOUNT

*

1_000_000); const

tx

=

await

client . post .transfer (subaccount , wallet .address , SUBACCOUNT_NUMBER_TO , USDC_ASSET_ID , quantums);

Determining parameters

- Asset

Asset ID can be fetched using the/dydxprotocol/assets/asset endpoint (Example(opens in a new tab)). Collateral token (USDC) will have an asset ID0 .

- Quantums

For collateral token, multiply by 10^6. For example,100 USDC = 100_000_000 quantums

Pulling current balance

Main Account

- Token balances can be fetched via/cosmos/bank/v1beta1/balances/{address}
- endpoint (Example(opens in a new tab)
-).

Subaccounts

- Collateral token position/balance can be fetched via/dydxprotocol/subaccounts/subaccount/{address}/{subaccountNumber}
- endpoint.Example(opens in a new tab)

Last updated onJune 20, 2024 CLI Python Script How to integrate APIs with FE isolated positions