# Sending funds

CosmPy allows for simplification of transactions broadcasting. Aftercreating a wallet ↗ you can send transactions by following the below provided example. In it, we first connect to the desired network then provide the desired destination address. The transaction is then broadcast using theLedger_client.send_tokens() function:

```
from cosmpy . aerial . client import LedgerClient , NetworkConfig
```

## Establishing connection to the network

## ledger_client

```
LedgerClient (NetworkConfig. fetch_mainnet ())
```

## Providing destination address

## destination_address

```
'fetch1h2l3cnu7e23whmd5yrfeunacez9tv0plv5rxqy'
```

## Broadcasting the transaction

## tx

```
ledger_client . send_tokens (destination_address, 10 , "atestfet" , wallet)
```

## block until the transaction has been successful or failed

tx . wait_to_complete () In this example, we want to send 10 atestfet tokens to thedestination_address . Thesend_tokens() method is called on theledger_client object, which is an instance of theLedgerClient class used to interact with the Fetch.ai blockchain network. The parameter wallet is the local wallet used to sign the transaction and provide the necessary credentials. The code then waits for the transaction to complete (either successfully or failed) before proceeding further.

### Sending and verifying a transaction

For a more comprehensive example, we will use thetestnet and submit a transaction then verify its completion by exploring the output on the terminal. For transactions made on the mainnet it is advisable to use ourblock explorer ↗(opens in a new tab) or any other block explorer, such asMintscan ↗(opens in a new tab). The following code snippet shows you how to create two wallets, namely foralice andbob .

Once created, the next step is to establish a connection to Fetch.ai's testnet. We then queryalice 's balance and include a while function to make surealice has a positive balance and can perform the transaction. In this casealice will sendbob 10 atestfet. As a final step, we will print the transaction hash, as well asbob 's andalice 's related balances:

## Import the required libraries and modules

```
from cosmpy . aerial . client import LedgerClient , NetworkConfig
```

## Required to establish a connection to the network

```
from cosmpy . aerial . faucet import FaucetApi
```

## Required to fund Alice's wallet if she has a 0 balance

```
from cosmpy . aerial . wallet import LocalWallet
```

## Required to create Alice and Bob's local wallets

def

main (): """Run main."""

## Create Alice and Bob's wallets

## alice

LocalWallet . generate () bob = LocalWallet . generate ()

## Establish a connection to the testnet

## ledger

LedgerClient (NetworkConfig. fetchai_stable_testnet ()) faucet_api =

FaucetApi (NetworkConfig. fetchai_stable_testnet ())

## Query Alice's Balance

## alice_balance

ledger . query_bank_balance (bob. address ())

## Should alice have no funds in her wallet, add funds

while alice_balance < ( 10 ** 18 ) : print ( "Providing wealth to alice..." ) faucet_api . get_wealth (alice. address ()) alice_balance = ledger . query_bank_balance (alice. address ())

## Print Alice and Bob's addresses and balances

print ( f "Alice Address: { alice. address () } Balance: { ledger. query_bank_balance (alice. address ()) } " )

print ( f "Bob Address: { bob. address () } Balance: { ledger. query_bank_balance (bob. address ()) } " )

## Broadcast the transaction and print the outputs on the terminal

## tx

ledger . send_tokens (bob. address (), 10 , "atestfet" , alice) print ( f "TX { tx.tx_hash } waiting to complete..." )

tx . wait_to_complete ()

print ( f "TX { tx.tx_hash } waiting to complete...done" )

print ( f "Alice Address: { alice. address () } Balance: { ledger. query_bank_balance (alice. address ()) } " )

print ( f "Bob Address: { bob. address () } Balance: { ledger. query_bank_balance (bob. address ()) } " )

if

**name**

==

"**main**" : main () Once you have executed the script, you will notice the console output is consistent with the code. At first, we see a print of the starting balance of both addresses. Then, we see the transaction hash along with the new balances for bothalice andbob :

Alice Address: fetch1kfsarzv8s0kl7pcznjgyc6mnaz96akl79kzttv Balance: 10000000000000000000 Bob Address: fetch1ww8gsxk4usmq0tdjyele4nqyxcmc3zquu3fc52 Balance: 0 TX 25E8F5D9F101DB5AC57DC5FF2566BA4F290609B6AADBFF1D566B398BA5FFAF48 waiting to complete... TX 25E8F5D9F101DB5AC57DC5FF2566BA4F290609B6AADBFF1D566B398BA5FFAF48 waiting to complete...done Alice Address: fetch1kfsarzv8s0kl7pcznjgyc6mnaz96akl79kzttv Balance: 9999416775000000512 Bob Address: fetch1ww8gsxk4usmq0tdjyele4nqyxcmc3zquu3fc52 Balance: 10

## Was this page helpful?

[Wallets and private keys](#) [Staking](#)