Querying balances

Once you have established aconnection to a network node ↗ , you can use theLegderClient object to perform many useful operations, such asquerying balances . In particular theLedgerClient object allows you to:

1. Query all balances associated with aparticular address
2. .
3. Query all balances associated with aparticular denomination
4. .

Let's explore how to achieve both.

1. We start by importing CosmPy as well as the relevant modules.
2. Then, we connect to the chain we want to perform the query on.
3. Once connected, we specify the address we want to query. In our example we will connect to the Fetch.ai mainnet and we query all balances associated with a particular address:
4. import
5. cosmpy
6. from
7. cosmpy
8. .
9. aerial
10. .
11. client
12. import
13. LedgerClient
14. ,
15. NetworkConfig
16. ledger_client
17. =
18. LedgerClient
19. (NetworkConfig.
20. fetch_mainnet
21. ())
22. address
23. :
24. str
25. =
26. 'fetch12q5gw9l9d0yyq2th77x6pjsesczpsly8h5089x'
27. balances
28. =
29. ledger_client
30. .
31. query_bank_all_balances
32. (address)
33. print
34. (balances)

Wheneverquerying all balances associated with aparticular address returns aList ofCoin objects that containamount anddenom variables that correspond to all the funds held at the address and their denominations.

ℹ This list includes all natively defined coins along with any tokens transferred using the inter-blockchain communicationIBC ↗(opens in a new tab) protocol. Once completed, you will see the following output in your terminal:

[Coin (amount = '1616060698998992698400' , denom = 'afet'),

Coin (amount = '10' , denom = 'ibc/605C5B80A8253543F8038F96F56BA13BDD8D300E12F1B32A3FA2E1EB2A933FA1'),

Coin (amount = '5000000' , denom = 'ibc/B58E6786772640EC4B538AFC4393F742C326734B74CCAFAFBF7EFDC7D435B428')] In order to **query all balances associated with aparticular denomination* +, we can run the same script but substitute the final balance command we used in the snippet above, with:

balance

ledger_client . query_bank_balance (address, denom = 'afet') This will return the value of the (integer) funds held by the address with the specified denomination. If thedenom argument is omitted the function will return the fee denomination

specified in theNetworkConfig object used to initialize theLedgerClient .

Was this page helpful?

Connecting to a blockchain Wallets and private keys