# #

Bank

# #

Summary

This module is mainly used to transfer coins between accounts, query account balances, and provide common offline transaction signing and broadcasting methods. In addition, the available units of tokens in the IRIShub system are defined usingcoin-type .

# #

Usage Scenario

1. Query the balances of an account
2. Query the total balance of an account or of a specific denomination.
3. iris query bank balances[
4. address]
5. --denom
6. =
7. [
8. denom]
9. Transfer between accounts
10. For example, transfer 10iris from account A to account B:
11. iris tx bank send[
12. A]
13. [
14. B]
15. [
16. 10iris]
17. --fees
18. =
19. 0
20. .3iris --chain-id=
21. irishub
22. IRIShub supports multiple tokens in circulation, and in the future IRIShub will be able to include multiple tokens in one transaction.
23. Sign transactions generated offline
24. To improve account security, IRIShub supports offline signing of transactions to protect the account's private key. In any transaction, you can build an unsigned transaction using the flag --generate-only. Take transfer transaction as an example:
25. iris tx bank send[
26. from_key_or_address]
27. [
28. to_address]
29. [
30. amount]
31. --fees
32. =
33. 0
34. .3iris --generate-only
35. Return the built transaction with empty signatures:
36. {
37. "type"
38. :
39. "auth/StdTx"
40. ,
41. "value"
42. :
43. {
44. "msg"
45. :
46. [
47. "txMsg"
48. ]

,
"fee"
:
"fee"
,
"signatures"
:
null
,
"memo"
:
""
}
}
Save the result to a file.
Sign the transaction:
iris tx sign<
file>
--chain-id=
irishub--from
=
<
key-name>
Return signed transactions:
{
"type"
:
"auth/StdTx"
,
"value"
:
{
"msg"
:
[
"txMsg"
]
,
"fee"
:
"fee"
,
"signatures"
:
[
{
"pub_key"
:
{
"type"
:
"tendermint/PubKeySecp256k1"
,
"value"
:
"A+qXW5isQDb7blT/KwEgQHepji8RfpzIstkHpKoZq0kr"
}
,
"signature"
:
"5hxk/R81SWmKAGi4kTW2OapezQZpp6zEnaJbVcyDiWRfgBm4Uejq8+CDk6uzk0aFSgAZzz06E014UkgGpeIU7w=="
,
"account_number"
:
"0"
,
"sequence"
:

117. "11"
118. }
119. ]
120. ,
121. "memo"
122. :
123. ""
124. }
125. }
126. Save the result to a file.
127. Broadcast transactions
128. Broadcast offline signed transactions. Here you can just use the transaction generated by the above signing command. Of course, you can generate your signed transaction by any methods, eg.irisnet-cryptoopen in new window
129. .
130. iris tx broadcast<
131. file>
132. The transaction will be broadcast and executed in IRIShub.