Multisig

Celestia inherits support for multisig accounts from the Cosmos SDK. Multisig accounts behave similarly to regular accounts with the added requirement that a threshold of signatures is needed to authorize a transaction.

Multisig accounts can be created from thecommand line or using a graphical interface such asKeplr .

Command line

bash

!/bin/sh

Prerequisite: prior to running this script, start a single node devnet with ./scripts/single-node.sh

CHAIN_ID

"private" KEY_NAME = "validator" KEYRING_BACKEND = "test" BROADCAST_MODE = "block"

Create 3 test keys

celestia-appd

keys

add

test1 celestia-appd

keys

add

test2 celestia-appd

keys

add

test3

Create the multisig account

celestia-appd

keys

add

multisig

\ --multisig

test1,test2,test3

\ --multisig-threshold

2

Send some funds from the validator account to the

multisig account

celestia-appd

tx

bank

send VALIDATOR MULTISIG 100000 utia

\ --from VALIDATOR \ --fees

1000 utia

\ --chain-id CHAIN_ID \ --keyring-backend KEYRING_BACKEND \ --broadcast-mode BROADCAST_MODE \ --yes

Send some funds from the multisig account to the validator account.

Note this transaction will need to be signed by at least 2 of the 3 test accounts.

celestia-appd

tx

bank

send MULTISIG VALIDATOR 1 utia

\ --from MULTISIG \ --fees

1000 utia

\ --chain-id CHAIN_ID \ --keyring-backend KEYRING_BACKEND \ --generate-only

unsignedTx.json

Sign from test1 and test2

celestia-appd

tx

sign

unsignedTx.json

\ --multisig MULTISIG \ --from

test1

\ --output-document

test1sig.json

\ --chain-id CHAIN_ID celestia-appd

tx

sign

unsignedTx.json

\ --multisig MULTISIG \ --from

test2

\ --output-document

test2sig.json

\ --chain-id CHAIN_ID

Generate the final signed transaction

celestia-appd

tx

multisign

unsignedTx.json

multisig

\ test1sig.json

test2sig.json

\ --output-document

signedTx.json

\ --chain-id CHAIN_ID

!/bin/sh

Prerequisite: prior to running this script, start a single node devnet with ./scripts/single-node.sh

CHAIN_ID

"private" KEY_NAME = "validator" KEYRING_BACKEND = "test" BROADCAST_MODE = "block"

Create 3 test keys

celestia-appd

keys

add

test1 celestia-appd

keys

add

test2 celestia-appd

keys

add

test3

Create the multisig account

celestia-appd

keys

add

multisig

\ --multisig

test1,test2,test3

\ --multisig-threshold

2

Send some funds from the validator account to the multisig account

celestia-appd

tx

bank

send VALIDATOR MULTISIG 100000 utia

\ --from VALIDATOR \ --fees

1000 utia

\ --chain-id CHAIN_ID \ --keyring-backend KEYRING_BACKEND \ --broadcast-mode BROADCAST_MODE \ --yes

Send some funds from the multisig account to the validator account.

Note this transaction will need to be signed by at least 2 of the 3 test accounts.

celestia-appd

tx

bank

send MULTISIG VALIDATOR 1 utia

\ --from MULTISIG \ --fees

1000 utia

\ --chain-id CHAIN_ID \ --keyring-backend KEYRING_BACKEND \ --generate-only

unsignedTx.json

Sign from test1 and test2

celestia-appd

tx

sign

unsignedTx.json

\ --multisig MULTISIG \ --from

test1

\ --output-document

test1sig.json

\ --chain-id CHAIN_ID celestia-appd

tx

sign

unsignedTx.json

\ --multisig MULTISIG \ --from

test2

\ --output-document

test2sig.json

\ --chain-id CHAIN_ID

Generate the final signed transaction

celestia-appd

tx

multisign

unsignedTx.json

multisig

\ test1sig.json

test2sig.json

\ --output-document

signedTx.json

\ --chain-id CHAIN_ID

Resources

- https://docs.cosmos.network/main/user/run-node/multisig-guide#step-by-step-guide-to-multisig-transactions
- https://figment.io/insights/how-to-multi-sig-on-cosmos/
- https://github.com/aura-nw/Aura-Safe
- https://github.com/informalsystems/multisig [][Edit this page on GitHub] Last updated: Previous page Wallets in celestia-app Next page Create a vesting account []