Staking

Introduction

Staking is a crucial mechanism for Proof-of-Stake (PoS) based blockchains, empowering users to actively participate in securing the network while earning rewards. Staked (delegated) funds are bonded to validator nodes. These validator nodes validate transactions and maintaining the network's integrity. By bonding funds to validators, individuals earn governance rights and actively contribute to the network's security and decentralization.

ℹ Bonding is the process of telling the network of your choice that you want to stake tokens on it. The bonding period is the lapse of time the blockchain delegator waits before his asset are bonded. Conversely, unbonding is the action of telling the network you want to unlock those assets. The unbonding period is the designated amount of time that a blockchain delegator waits before having access to move or sell their tokens.

Delegating

Considering CosmPy, theLedgerclient object provides utilities for interacting with the staking component of the network. As you continue to explore this guide, you will notice that theLedgerClient object plays a part in all stake related processes. To begin with, we will start with one example to showcase how to use CosmPy to delegate 20 FET to a validator, using awallet object:

validator_address

'fetchvaloper1e4ykjwcwhwtasqxq50d4m7xz9hh7a86e9y8h87'

tx

ledger_client . delegate_tokens (validator_address, 20 , wallet)

block until the transaction has been successful or failed

tx . wait_to_complete ()

Re-delegating

Similarly to the delegate option, re-delegating is the process of unbonding staked funds from one validator node and bonding them to another. This is highly useful in cases where a better validator option has arisen or in crisis management situations where a rapid re-delegation is needed. In the example below, you will learn how to re-delegate 10 tokens from an existingvalidator_address to anotheralternative_validator_address :

validator_address

'fetchvaloper1e4ykjwcwhwtasqxq50d4m7xz9hh7a86e9y8h87' alternate_validator_address =

'fetchvaloper1e4ykjwcwhwtasqxq50d4m7xz9hh7a86e9y8h87'

tx

ledger_client . redelegate_tokens (validator_address, alternate_validator_address, 10 , wallet)

block until the transaction has been successful or failed

tx . wait_to_complete ()

Undelegating

Undelegated funds go through anunbonding period (i.e., typically 14 to 21 days for Cosmos sdk based projects) and once their unbonding period is complete funds become transferable. The code snippet we have shared, showcases how to use theLedgerClient object to invoke the undelegation (in this case 5 tokens) and start the cool down process.

Importantly, the cool down is tracked for each undelegation invokation. For instance, if you trigger 3 undelegate actions on 3 consecutive days the first batch of tokens will become available 3 days before the final batch. In the snipet below, we provide the command needed to undelegate your tokens:

tx

ledger_client . undelegate_tokens (validator_address, 5 , wallet)

block until the transaction has been successful or failed

tx . wait_to_complete ()

Claiming Rewards

Delegated funds accrue rewards . Rewards can be collected at any time and unlike delegations become immediately available.

You can claim rewards from a specific validator using the following commands:

tx

ledger_client . claim_rewards (validator_address, wallet)

block until the transaction has been successful or failed

tx . wait_to_complete ()

Stake queries

Additionally, CosmPY allows you to perform usefulstake queries andsummaries . You can query the stake information on any particular address at any point using theLedgerClient as shown in the example below:

address

'fetch1h2l3cnu7e23whmd5yrfeunacez9tv0plv5rxqy'

s

ledger_client . query_staking_summary (address) print (f "Summary: Staked: { s.total_staked } Unbonding: { s.total_unbonding } Rewards: { s.total_rewards } ")

Was this page helpful?

Sending funds Smart contracts