CLI - Delegation

Delegations refers to the act of assigning the responsibility of validating transactions and creating new blocks to a specific validator node. Below you can find different operations executable through theCLI ↗ .

How to query the current staking holdings of validators

You can query the current staking holdings of validators using thefetchd CLI, and run the following command:

fetchd

query

staking

validators This command will provide information about all the existing validators in the network, including details such as their operator address, consensus public key, status, tokens staked, commission rates, and more.

The output will be a list of validators, each represented in a block of information, similar to the example you provided earlier. This information gives an overview of each validator's status and staked tokens.

Ondorado test network, this command will produce an output similar to the following:

- | operatoraddress: fetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w conspubkey: fetchvalconspub1zcjduepq3urw6c6u0zvqmde4vr4gmy56nnq57shdhg56jynpu8n3s74hrm0q0mzqrx jailed: false status: 2 tokens: "1000000000000000000000" delegatorshares: "1000000000000000000000.000000000000000000" description: moniker: validator5 identity: "" website: "" security_contact: "" details: "" unbondingheight: 0 unbondingcompletiontime: 1970-01-01T00:00:00Z commission: commission_rates: rate: "0.050000000000000000" max_rate: "0.100000000000000000" max_change_rate: "0.010000000000000000" update_time: 2021-02-12T12:41:25.579730119Z minselfdelegation: "1000000000000000000000" producingblocks: true
- | operatoraddress: fetchvaloper1ysc8n5uspv4698nyk8u75lx98uu92zt7m3udw8 conspubkey: fetchvalconspub1zcjduepqmxr8gmcs6pwuxpsma264ax59wxtxd3vchrcv2c06deq9986kwt3s0wsk6n jailed: false status: 2 tokens: "1000000000000000000000" delegatorshares: "1000000000000000000000.000000000000000000" description: moniker: validator2 identity: "" website: "" security_contact: "" details: "" unbondingheight: 0 unbondingcompletiontime: 1970-01-01T00:00:00Z commission: commission_rates: rate: "0.050000000000000000" max_rate: "0.100000000000000000" max_change_rate: "0.010000000000000000" update_time: 2021-02-03T13:00:00Z minselfdelegation: "1000000000000000000000" producingblocks: true ... Similarly, if you wish to retrieve the same information but now for asingle validator , use the following command, by providing theoperator_address of the specific validator:

fetchd

query

staking

validator

operator_address For instance:

fetchd

query

staking

validator

fetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w This command will provide detailed information about the specific validator, including their commission rates, minimum self-delegation, and other relevant details.

A delegator will be particularly interested in the following keys:

- commission/commission_rates/rate
- : the commission rate on revenue charged to any delegator by the validator.
- commission/commission_rates/max_change_rate
- : the maximum daily increase of the validator's commission. This parameter cannot be changed by the validator operator.
- commission/commission_rates/max_rate
- : the maximum commission rate this validator can charge. This parameter cannot be changed by the validator

operator.

- minselfdelegation
- : minimum amount ofatestfet
- the validator need to have bonded at all time. If the validator's self-bonded stake falls below this limit, their entire staking pool (i.e. all its delegators) will unbond. This parameter exists as a safeguard for delegators. Indeed, when a validator misbehaves, part of their total stake gets slashed. This includes the validator's self-delegateds stake as well as their delegators' stake. Thus, a validator with a high amount of self-delegatedatestfet
- has more skin-in-the-game than a validator with a low amount. The minimum self-bond amount parameter guarantees to delegators that a validator will never fall below a certain amount of self-bonded stake, thereby ensuring a minimum level of skin-in-the-game. This parameter can only be increased by the validator operator.

How to query the delegations made to a validator

You canquery the list of delegations made to a specific validator using thefetchd command-line interface, and run the following command:

fetchd

query

staking

delegations-to

fetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w Here, replaceVALIDATOR_OPERATOR_ADDRESS with the actual operator address of the validator you are interested in. For instance:

fetchd

query

staking

delegations-to

VALIDATOR_OPERATOR_ADDRESS This command will provide information about the delegations made to the specified validator, including details such as the delegator's address, the validator's address, and the amount of shares and tokens delegated.

Below, you can find an example of delegations tovalidator2 received ondorado testnet:

- delegation: delegator_address: fetch1z72rph6l5j6ex83n4urputykawcqg6t9zzruef validator_address: fetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w shares: "1000000000000000000000.000000000000000000" balance: denom: atestfet amount: "1000000000000000000000"
- delegation: delegator_address: fetch15fn3meky8ktfry3qm73xkpjckzw4dazxpfx34m validator_address: fetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w shares: "100000.000000000000000000" balance: denom: atestfet amount: "100000" This output shows two delegations made to the validator, along with the delegator's address, the validator's address, the number of shares, and the amount of tokens delegated.

How to query re-delegations

Re-delegation is the process of transferring already delegated tokens from one validator to another. This allows participants to change their delegation strategy without having to unbond and wait for the unbonding period to complete.

Delegators can choose to re-delegate the tokens they already delegated from one validator to another at any time. Re-delegation takes effect immediately, without any waiting period. However, the tokens can not be re-delegated until the initial re-delegation transaction has completed its 21 day completion time. The unlocking time is indicated by theredelegationentry/completion_time field in the outputs below.

You canquery the list of re-delegations made from a validator by using thefetchd command-line interface and run the following command:

fetchd

query

staking

redelegations-from

VALIDATOR_OPERATOR_ADDRESS ReplaceVALIDATOR_OPERATOR_ADDRESS with the actual operator address of the validator you are interested in. For instance:

fetchd

query

staking

redelegations-from

fetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w This command will provide information about the re-delegations made from the specified validator, including details such as the delegator's address, the source validator's address, the destination validator's address, and information about the re-delegation entries. The output will be a list of re-delegations, each represented in a block of information. Below you can find an example output:

fetchd query staking redelegations-from fetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w - redelegation: delegator_address: fetch15fn3meky8ktfry3qm73xkpjckzw4dazxpfx34m validator_src_address: fetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w validator_dst_address: fetchvaloper122veneudkzyalay6gusvrhhpp0560mparpanvu entries: [] entries: - redelegationentry: creation_height: 291037 completion_time: 2021-03-24T14:24:38.973444629Z initial_balance: "50000" shares_dst: "50000.000000000000000000" balance: "50000" - redelegationentry: creation_height: 291133 completion_time: 2021-03-24T14:33:43.425472866Z initial_balance: "10000" shares_dst: "10000.000000000000000000" balance: "10000" Here, delegatorfetch15fn3meky8ktfry3qm73xkpjckzw4dazxpfx34m issued 2 re-delegations fromfetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w tofetchvaloper122veneudkzyalay6gusvrhhpp0560mparpanvu :

Similarly, you can obtain the list of re-delegations issued by a delegator by running the following command:

fetchd

query

staking

redelegations

fetch15fn3meky8ktfry3qm73xkpjckzw4dazxpfx34m

How to query rewards

Once you have delegated tokens to a validator, you will be eligible to a share of the rewards the validator collects.

If you wish toretrieve all the outstanding rewards for a specific address , run the following command using thefetchd command-line interface, :

fetchd

query

distribution

rewards

DELEGATOR_ADDRESS You will need to replaceDELEGATOR_ADDRESS with the actual address of the delegator whose rewards you want to query. For instance:

fetchd

query

distribution

rewards

fetch15fn3meky8ktfry3qm73xkpjckzw4dazxpfx34m This command will provide information about the rewards earned by the specified delegator, including details such as the validator addresses and the amount of rewards in different denominations. The output will be a list of rewards, each represented in a block of information. Below you can find an example output:

rewards: - validator_address: fetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w reward: - denom: atestfet amount: "0.000000000000200000" - validator_address: fetchvaloper1ysc8n5uspv4698nyk8u75lx98uu92zt7m3udw8 reward: - denom: atestfet amount: "0.000000000001000000" total: - denom: atestfet amount: "0.000000000001200000" In this example, the delegator atDELEGATOR_ADDRESS has earned rewards from two different validators ondorado test network. The rewards are listed in different denominations, such asatestfet .

You can alsofilter rewards for a given validator . For instance, you can filter rewards forvalidator5 (fetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w) as shown below:

fetchd

query

distribution

rewards

fetch15fn3meky8ktfry3qm73xkpjckzw4dazxpfx34m

fetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w In the output, we get the rewards from this specific validator:

- denom: atestfet amount: "0.000000000000200000"

Delegator operations

How to delegate tokens

If you want to delegate1000000 atestfet tokens to thefetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w validator from the accountmyKey , then you will need to use the following command:

fetchd

tx

staking

delegate

fetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w

1000000 atestfet

--from

myKey This will require a confirmation before issuing a transaction. After the transaction gets processed, it should appear under the delegations of thefetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w validator.

ℹ Once delegated, tokens can only bere-delegated to another validator, orunbond in order to be returned to their original account. It is important to note that those two operations take21 days to complete , period in which the involved tokens will be unavailable.

Re-delegating tokens

Re-delegating tokens allows you to transfer already delegated tokens from one validator to another .

From the above example where you delegated1000000 atestfet tofetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w , you can now re-delegate parts or all of those tokens to another validator. For instance, you can re-delegate400000 atestfet fromfetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w tofetchvaloper122veneudkzyalay6gusvrhhpp0560mparpanvu by running the following command:

fetchd

tx

staking

redelegate

fetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w

fetchvaloper122veneudkzyalay6gusvrhhpp0560mparpanvu

400000 atestfet

--from

myKey This will prompt for confirmation and issue a new transaction once accepted.

From here, if you inspect the delegations from our account, you will be able to see that your delegated tokens are now:

- 600000atestfet
- to validatorfetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w
- (our initial 1000000 minus the 400000 re-delegated).
- 400000atestfet
- to validatorfetchvaloper122veneudkzyalay6gusvrhhpp0560mparpanvu
- .

Now, those400000 atestfet you re-delegated can not be re-delegated anymore for 21 days (the exact date can be found by querying the re-delegation transaction, under thecompletion_time key).

ℹ It is still possible to unbond those tokens if needed.

How to unbond tokens

Bonding refers to the act of locking up a certain amount of cryptocurrency tokens in a wallet or smart contract to participate in the network's consensus mechanism. These tokens are often referred to as thestake . Conversely,unbonding is the process of withdrawing or releasing the previously bonded tokens. When a user initiates an unbonding transaction, they are indicating that they want to take back their tokens from the staking mechanism.

You can transfer parts or all of our delegated tokens back to your account at any time by running the following command:

fetchd

tx

staking

unbond

fetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w

300000 atestfet

--from

myKey Once again, this will prompt for confirmation and issue a transaction, initiating the transfer of300000 atestfet from our stake onfetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w validator address back to your account. Those tokens will then be availableafter a 21 day period (the exact date can be found by querying the re-delegation transaction, under thecompletion_time key).

How to withdraw rewards

In orderto transfer rewards to the wallet , run the following command:

fetchd

tx

distribution

withdraw-rewards

validator_address

--from

myKey It requires the validator address from where the reward is withdrawn, and the name of the account private key having delegated tokens to the validator. For instance:

fetchd

tx

distribution

withdraw-rewards

fetchvaloper1z72rph6l5j6ex83n4urputykawcqg6t98xul2w

--from

myKey You canclaim all rewards when you have delegated tokens to multiple validators , by running the following command:

fetchd

tx

distribution

withdraw-all-rewards

--from

myKey The rewards will appear on the account as soon as the transaction is being processed.

Was this page helpful?

CLI - Multisig keys Governance proposals