Interacting with governance

Thenamada_impl object is assumed to have been constructed as described in thesetting up a client section.

Constructing a proposal

let source_address = namada_sdk :: Address :: from_str ("tnam1v4ehgw36xq6ngs3ng5crvdpngg6yvsecx4znjdfegyurgwzzx4pyywfexuuyys69gc6rzdfnryrntx") . unwrap (); let start_epoch =

420

as

u64 ; let end_epoch =

424

as

u64 ; let grace_epoch =

428

as

u64 ; let signing_key = namada_sdk :: SecretKey :: from_str (FAUCET_KEY) . unwrap ();

let proposal =

json! ({ "proposal" : { "content" : { "title" :

"TheTitle" , "authors" :

" " , "discussions-to" :

"forum.namada.net" , "created" :

"2024-03-10T08:54:37Z" , "license" :

"MIT" , "abstract" :

"Ut convallis eleifend orci vel venenatis. Duis vulputate metus in lacus sollicitudin vestibulum. Suspendisse vel velit ac est consectetur feugiat nec ac urna. Ut faucibus ex nec dictum fermentum. Morbi aliquet purus at sollicitudin ultrices. Quisque viverra varius cursus. Praesent sed mauris gravida, pharetra turpis non, gravida eros. Nullam sed ex justo. Ut at placerat ipsum, sit amet rhoncus libero. Sed blandit non purus non suscipit. Phasellus sed quam nec augue bibendum bibendum ut vitae urna. Sed odio diam, ornare nec sapien eget, congue viverra enim." , "motivation" :

"Ut convallis eleifend orci vel venenatis. Duis vulputate metus in lacus sollicitudin vestibulum. Suspendisse vel velit ac est consectetur feugiat nec ac urna. Ut faucibus ex nec dictum fermentum. Morbi aliquet purus at sollicitudin ultrices." , "details" :

"Ut convallis eleifend orci vel venenatis. Duis vulputate metus in lacus sollicitudin vestibulum. Suspendisse vel velit ac est consectetur feugiat nec ac urna. Ut faucibus ex nec dictum fermentum. Morbi aliquet purus at sollicitudin ultrices. Quisque viverra varius cursus. Praesent sed mauris gravida, pharetra turpis non, gravida eros." , "requires" :

"2" }, "author" : source_address . to_string (), "voting_start_epoch" : start_epoch, "voting_end_epoch" : end_epoch, "grace_epoch" : grace_epoch } }) . to_string ()

Submitting the proposal

Once the json is constructed, the proposal can be submitted to the network using the following code:

let proposal_data = proposal . as_bytes () . to_vec ();

let init_proposal_tx_builder = namada_impl . new_init_proposal (proposal_data) . signing_keys (vec! [signing_key]);

let (mut init_proposal_tx, signing_data) = init_proposal_tx_builder . build (& namada_impl) .await . expect ("unable to build init_proposal tx");

namada_impl . sign (&mut init_proposal_tx, & init_proposal_tx_builder . tx, signing_data, default_sign, (),) .await . expect (

"unable to sign redelegate tx"); let tx = namada_impl . submit (init_proposal_tx, & init_proposal_tx_builder . tx) .await ;

Voting on a proposal

In order to vote on a proposal, we need a valid proposal id. We can retrieve the proposal id from the latest proposal submitted to the network using the following code:

let storage_key = namada_governance :: storage :: keys :: get_counter_key (); // This returns the next proposal_id, so always subtract 1 let proposal_id = namada_sdk :: rpc :: query_storage_value :: <_, u64

(namada_impl . client (), & storage_key) .await . unwrap () -

1 ; Once we have the proposal id, we can vote on the proposal using the following code:

let proposal_id =

1

as

u64 // placeholder, replace with actual proposal id let vote =

String :: from ("Yay"); let signing_public_key = signing_key . to_public ();

let vote_proposal_tx_builder = namada_impl . new_vote_prposal (vote . clone (), voter_address . clone ()) . proposal_id (proposal_id) . signing_keys (vec! [signing_public_key]);

let (mut vote_proposal_tx, signing_data) = vote_proposal_tx_builder . build (& namada_impl) .await . expect ("unable to build vote_proposal tx");

namada_impl . sign (&mut vote_proposal_tx, & vote_proposal_tx_builder . tx, signing_data, default_sign, (),) .await . expect ("unable to sign redelegate tx"); let tx = namada_impl . submit (vote_proposal_tx, & vote_proposal_tx_builder . tx) .await ;

Proof of stake transactions Integrating with the interface