Unit Tests

Testing contract functionality can be done through thecargo test framework. These tests will run with a mocked blockchain and will allow testing function calls directly without having to set up/deploy to a network and sign serialized transactions on this network.

A common framework for tests, along with setting up a basic testing environment looks like:

[cfg(all(test, not(target_arch =

"wasm32"))] mod

tests

{ use

super :: * ; use

near_sdk :: test_utils :: VMContextBuilder ; use

near_sdk :: { testing_env ,

VMContext } ;

fn

get_context (is_view :

bool)

->

VMContext

{ VMContextBuilder :: new ( ) . signer_account_id ("bob_near" . parse ( ) . unwrap ( )) . is_view (is_view) . build ( ) }

[test]

fn

my_test ( )

{ let context =

get_context (false) ; testing_env! (context) ; // ... Write test here } } WhereVMContextBuilder allows for modifying the context of the mocked blockchain to simulate the environment that a transaction would be run. The documentation for what can be modified with this context can be foundhere .

Thetesting_env! macro will initialize the blockchain interface with theVMContext which is either initialized throughVMContextBuilder or manually through itself.

Note Thistesting_env! andVMContext is only used for testing outside ofwasm environments. When running the built contract on a network in awasm environment, the context from the blockchain will be used through host functions on the runtime. To test read-only function calls, setis_view totrue on theVMContext . This will test to verify that function calls which just read state do not try to modify state through unit tests. In the above example,true should be passed into theget_context call, which initializes the context as read-only.

You will want to usetesting_env! each time you need to update this context, such as mocking thepredecessor_accound_id to simulate the functions being called by or only allowing view operations as mentioned above. Each time this is done, a new mocked blockchain will be initialized while keeping the existing state. Edit this page Last updatedonMar 7, 2024 byDamián Parrino Was this page helpful? Yes No

Previous Integration Tests Next Best Practices