Using Ledger wallet with ethers on Sei

Prerequisites:

1. Make sure you have a Ethereum app installed on your Ledger device.
2. Enable blind signing in the Ethereum app settings (needed for precompile examples and working with contracts in general).
3. Linux users may need to add a udev rule to allow access to the device. You can find the script to add a rule in the Ledger repohere(opens in a new tab)
4. .

Sending funds example

import {LedgerSigner} from

"@ethers-ext/signer-ledger" ; import Transport from

"@ledgerhq/hw-transport-node-hid" ; import {ethers} from

"ethers" ;

const

send

=

async (signer :

LedgerSigner , amount :

string , toAddress :

string) => { // Create a transaction const

transaction

= { to : toAddress , value :

ethers .parseEther (amount) , gasPrice :

ethers .parseUnits ('200' ,

'gwei') , }

// Send the transaction signer .sendTransaction (transaction) .then ((tx) => { console .log (tx); }); }

const

testApp

=

async () => { const

rpcUrl

=

"https://evm-rpc-testnet.sei-apis.com" ;

const

provider

=

new

ethers .JsonRpcProvider (rpcUrl); const

signer

=

new

LedgerSigner (Transport , provider);

await

send (signer ,

"5" ,

"<0xAddress>"); // Send 5 Sei

};

Send funds to native Sei address via precompile

import {LedgerSigner} from

"@ethers-ext/signer-ledger" ; import Transport from

"@ledgerhq/hw-transport-node-hid" ; import {ethers} from

"ethers" ;

import { BANK_PRECOMPILE_ABI , BANK_PRECOMPILE_ADDRESS , } from

"@sei-js/evm" ;

const

sendNative

=

async (signer :

LedgerSigner , amount :

string , nativeAddress :

string) => { const

contract

=

new

ethers .Contract (BANK_PRECOMPILE_ADDRESS ,

BANK_PRECOMPILE_ABI , signer); const

overrides

= { value :

ethers .parseEther (amount) }

try { const

result

=

await

contract .sendNative (nativeAddress , overrides); console .log ("SendNativeResult: " , result); } catch (e) { console .log (e) }
}

const

testApp

=

async () => { const

rpcUrl

=

"https://evm-rpc-testnet.sei-apis.com" ;

const

nativeAddress

=

"seiAddress" const

provider

=

new

ethers .JsonRpcProvider (rpcUrl); const

signer

=

new

LedgerSigner (Transport , provider);

await

sendNative (signer ,

"5" , nativeAddress); };

Staking via precompile example

import {LedgerSigner} from

"@ethers-ext/signer-ledger" ; import Transport from

"@ledgerhq/hw-transport-node-hid" ; import {ethers , toUtf8String} from

"ethers" ;

import { STAKING_PRECOMPILE_ABI , STAKING_PRECOMPILE_ADDRESS } from

"@sei-js/evm" ;

const

stake

=

async (signer :

LedgerSigner , amount : string , fromAddress :

string , validatorAddress :

string) => { const

contract

=

new

ethers .Contract (STAKING_PRECOMPILE_ADDRESS ,

STAKING_PRECOMPILE_ABI , signer); const

overrides

= { from : fromAddress , value :

ethers .parseEther (amount) ,

//staking 3.5 sei gasLimit :

ethers .parseEther ('0.00000000001') }

console .log ("Staking " , overrides) contract .delegate (validatorAddress , overrides) .then ((result) => { console .log ("DelegateResult: " ,

toUtf8String (result)); }); }

const

testApp

=

async () => { const

validatorAddress

=

"seivaloperAddress" ; const

rpcUrl

=

"https://evm-rpc-testnet.sei-apis.com" ;

const

provider

=

new

ethers .JsonRpcProvider (rpcUrl); const

signer

=

new

LedgerSigner (Transport , provider);

const

defaultAddress

=

await

signer .getAddress ();

await

stake (signer ,

"5" , defaultAddress , validatorAddress); // stake 5 Sei }; Last updated onAugust 21, 2024IBC Protocol Wallets