NEAR CLI

The NEARCommand Line Interface (CLI) is a tool that enables to interact with the NEAR network directly from the shell. Among other things, the NEAR CLI enables you to:

- Login with a NEAR account
- Deploy a contract
- Interact and query information from a deployed contract

tip Under the hood, NEAR CLI utilizes theNEAR JavaScript API

info The NEAR CLI also comes with an implementation in Rust callednear-cli-rs . If you want to usenear-cli while you havenear-cli-rs installed, prefix the following commands withnpx .

Overview

Click on a command for more information and examples.

Command Description ACCESS KEYS near add-credentials Stores credentials for an account locallynear add-key adds a new access key to an accountnear delete-key deletes an access key from an account near generate-key generates a key pair andoptionally stores it locally as credentials for an accountIdnear list-keys displays all access keys and their details for a given accountnear login stores a full access key locally usingNEAR Wallet ACCOUNTS near create-account creates a new account, either using a faucet to fund it, or an account saved locallynear delete-account deletes an account and transfers remaining balance to a beneficiary account near list-keys displays all access keys for a given accountnear send-near sends tokens from one account to anothernear state shows general details of an account CONTRACTS near call makes a contract call which can invokechange or view methodsnear deploy deploys a smart contract to the NEAR blockchainnear storage Shows the storage state of a given contract, i.e. the data stored in a contract near view makes a contract call which canonly invoke aview method TRANSACTIONSnear tx-status queries a transaction's status bytxHash

Setup

Installation

Make sure you have a current version ofnpm andNodeJS installed.

Mac and Linux

1. Installnpm
2. andnode
3. using a package manager likenvm
4. as sometimes there are issues using Ledger due to how OS X handles node packages related to USB devices[click here]
5. Ensure you have installed Node version 12 or above.
6. Installnear-cli
7. globally by running:

npm install -g near-cli For example, on Ubuntu 20.04near-cli can be installed by running:

Install nvm (https://github.com/nvm-sh/nvm?tab=readme-ov-file#installing-and-updating)

curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash source ~/.bashrc

Install node

nvm install node

Install near-cli

npm install -g near-cli

near-cli works!

near --help

Windows

For Windows users, we recommend using Windows Subsystem for Linux (WSL). 1. InstallWSL 2[click here] 3. Installnpm 4. [click here] 5. InstallNode.js 6. [click here] 7. Changenpm 8. default directory[click here] 9. * This is to avoid any permission issues withWSL 10. OpenWSL 11. and installnear-cli 12. globally by running:

npm install -g near-cli

Network selection

The default network fornear-cli istestnet . * You can change the network by prepending an environment variable to your command.

NEAR_NETWORK=testnet near send ... * Alternatively, you can set up a global environment variable by running:

export NEAR_NETWORK=mainnet * All commands that interact with the network also allow to pass the--networkId * option.

near send-near ... --networkId mainnet

Custom RPC server selection

You can set custom RPC server URL by setting these env variables:

NEAR_MAINNET_RPC NEAR_TESTNET_RPC Clear them in case you want to get back to the default RPC server.

Example:

export NEAR_TESTNET_RPC=

Access Keys

All keys are stored locally at the root of yourHOME directory:

- ~/.near-credentials
- (MAC / Linux)
- C:\Users\YOUR_ACCOUNT.near-credentials
- (Windows)

Inside.near-credentials , access keys are organized in network subdirectories:testnet , andmainnet .

These network subdirectories contain.JSON objects with an:

- account_id
- private_key
- public_key

near add-credentials

Stores credentials (full-access-key) locally for an already existing account. * arguments:accountId * options:--seedPhrase * or--secretKey

Examples:

near add-credentials example-acct.testnet --seedPhrase "antique attitude say evolve ring arrive hollow auto wide bronze usual unfold"

near add-key

Adds either afull access orfunction access key to a given account. Optionally allows to sign with a Ledger:--signWithLedger --ledgerPath Note: You will use anexisting full access key for the account you would like to add anew key to. (near login)

1) add afull access

key

- arguments:accountId
- publicKey

Example:

near add-key example-acct.testnet Cxg2wgFYrdLTEkMu6j5D6aEZqTb3kXbmJygS48ZKbo1S Example Response Adding full access key = Cxg2wgFYrdLTEkMu6j5D6aEZqTb3kXbmJygS48ZKbo1S to example-acct.testnet. Transaction Id EwU1ooEvkR42HvGoJHu5ou3xLYT3JcgQwFV3fAwevGJg To see the transaction in the transaction explorer, please open this url in your browser https://testnet.nearblocks.io/txns/EwU1ooEvkR42HvGoJHu5ou3xLYT3JcgQwFV3fAwevGJg

2) add afunction call

key

- arguments:accountId
- publicKey
- --contract-id
- options:--method-names
- --allowance

accountId is the account you are adding the key to

--contract-id is the contract you are allowing methods to be called on

--method-names are optional and if omitted, all methods of the--contract-id can be called.

--allowance is the amount of Ⓝ the key is allowed to spend on gas feesonly (default: 0). Note: Each transaction made with this key will have gas fees deducted from the initial allowance and once it runs out a new key must be issued.

Example:

near add-key example-acct.testnet GkMNfc92fwM1AmwH1MTjF4b7UZuceamsq96XPkHsQ9vi --contract-id example-contract.testnet --method-names example_method --allowance 30000000000 Example Response Adding function call access key = GkMNfc92fwM1AmwH1MTjF4b7UZuceamsq96XPkHsQ9vi to example-acct.testnet. Transaction Id H2BQL9fXVmdTbwkXcMFfZ7qhZqC8fFhsA8KDHFdT9q2r To see the transaction in the transaction explorer, please open this url in your browser https://testnet.nearblocks.io/txns/H2BQL9fXVmdTbwkXcMFfZ7qhZqC8fFhsA8KDHFdT9q2r

near delete-key

Deletes an existing key for a given account. Optionally allows to sign with a Ledger:--signWithLedger --ledgerPath * arguments:accountId * publicKey * options:--networkId * ,force

Note: You will need separate full access key for the account you would like to delete a key from.near login)

Example:

near delete-key example-acct.testnet Cxg2wgFYrdLTEkMu6j5D6aEZqTb3kXbmJygS48ZKbo1S Example Response Transaction Id 4PwW7vjzTCno7W433nu4ieA6FvsAjp7zNFwicNLKjQFT To see the transaction in the transaction explorer, please open this url in your browser https://testnet.nearblocks.io/txns/4PwW7vjzTCno7W433nu4ieA6FvsAjp7zNFwicNLKjQFT

near generate-key

Displays a key-pair and seed-phrase and optionally stores it locally in.near-credentials . * arguments:accountId * ornone * options:--fromSeedPhrase * ,--saveImplicit * ,--queryLedgerPK

Note: There are several ways to usegenerate-key that return very different results. Please reference the examples below for further details.

1a)near generate-key

Creates and displays a key pair near generate-key Example Response Seed phrase: antique attitude say evolve ring arrive hollow auto wide bronze usual unfold Key pair: {"publicKey":"ed25519:BW5Q957u1rTATGpanKUktjVmixEmT56Df4Dt9hoGWEXz","secretKey":"ed25519:5StmPDg9xVNzpyudwxT8Y72iyRq7Fa86hcpsRk6Cq5eWGWqwsPbPT9woXbJs9Qe69crZJHh4 Implicit account: 9c07afc7673ea0f9a20c8a279e8bbe1dd1e283254263bb3b07403e4b6fd7a411

1b)near generate-key --saveImplicit

Creates and displays a key pair, saving it locally in.near-credentials as an implicit account. near generate-key --saveImplicit Example Response Seed phrase: antique attitude say evolve ring arrive hollow auto wide bronze usual unfold Key pair: {"publicKey":"ed25519:BW5Q957u1rTATGpanKUktjVmixEmT56Df4Dt9hoGWEXz","secretKey":"ed25519:5StmPDg9xVNzpyudwxT8Y72iyRq7Fa86hcpsRk6Cq5eWGWqwsPbPT9woXbJs9Qe69crZJHh4 Implicit account: 9c07afc7673ea0f9a20c8a279e8bbe1dd1e283254263bb3b07403e4b6fd7a411

Storing credentials for account: 9d6e4506ac06ab66a25f6720e400ae26bad40ecbe07d49935e83c7bdba5034fa (network: testnet) Saving key to '~/.near-credentials/testnet/9d6e4506ac06ab66a25f6720e400ae26bad40ecbe07d49935e83c7bdba5034fa.json'

2)near generate-key accountId

Creates a key pair locally in.near-credentials with anaccountId that you specify. Note: This does NOT create an account with this name.

near generate-key example.testnet Example Response Seed phrase: antique attitude say evolve ring arrive hollow auto wide bronze usual unfold Key pair: {"publicKey":"ed25519:BW5Q957u1rTATGpanKUktjVmixEmT56Df4Dt9hoGWEXz","secretKey":"ed25519:5StmPDg9xVNzpyudwxT8Y72iyRq7Fa86hcpsRk6Cq5eWGWqwsPbPT9woXbJs9Qe69crZJHh4 Implicit account: 9c07afc7673ea0f9a20c8a279e8bbe1dd1e283254263bb3b07403e4b6fd7a411

Storing credentials for account: example.testnet (network: testnet) Saving key to '~/.near-credentials/testnet/example.testnet.json'

3a)near generate-key --fromSeedPhrase="your seed phrase"

Uses a seed phrase to display a public key andimplicit account near generate-key --seedPhrase="antique attitude say evolve ring arrive hollow auto wide bronze usual unfold" Example Response Seed

phrase: antique attitude say evolve ring arrive hollow auto wide bronze usual unfold Key pair: {"publicKey":"ed25519:BW5Q957u1rTATGpanKUktjVmixEmT56Df4Dt9hoGWEXz","secretKey":"ed25519:5StmPDg9xVNzpyudwxT8Y72iyRq7Fa86hcpsRk6Cq5eWGWqwsPbPT9woXbJs9Qe69crZJHh4 Implicit account: 9c07afc7673ea0f9a20c8a279e8bbe1dd1e283254263bb3b07403e4b6fd7a411

3b)near generate-key accountId --seedPhrase="your seed phrase"

Will store the key pair corresponding to the seedPhrase in.near-credentials with anaccountId that you specify.

Example Response Seed phrase: antique attitude say evolve ring arrive hollow auto wide bronze usual unfold Key pair: {"publicKey":"ed25519:BW5Q957u1rTATGpanKUktjVmixEmT56Df4Dt9hoGWEXz","secretKey":"ed25519:5StmPDg9xVNzpyudwxT8Y72iyRq7Fa86hcpsRk6Cq5eWGWqwsPbPT9woXbJs9Qe69crZJHh4 Implicit account: 9c07afc7673ea0f9a20c8a279e8bbe1dd1e283254263bb3b07403e4b6fd7a411

4a)near generate-key --queryLedgerPK

Uses a connected Ledger device to display a public key andmplicit account using the default HD path ("44'/397'/0'/0'/1'") near generate-key --queryLedgerPK You should then see the following prompt to confirm this request on your Ledger device:

Make sure to connect your Ledger and open NEAR app Getting Public Key from Ledger... After confirming the request on your Ledger device, a public key and implicit accountId will be displayed.

Example Response Using public key: ed25519:B22RP10g695wyeRvKlWv61NjmQZEkWTMzAYgdfx6oSeB2 Implicit account: 42c320xc20739fd9a6bqf2f89z61rd14efe5d3de234199bc771235a4bb8b0e1

3b)near generate-key --queryLedgerPK --ledgerPath="HD path you specify"

Uses a connected Ledger device to display a public key andmplicit account using a custom HD path. near generate-key --queryLedgerPK --ledgerPath="44'/397'/0'/0'/2'" You should then see the following prompt to confirm this request on your Ledger device:

Make sure to connect your Ledger and open NEAR app Waiting for confirmation on Ledger... After confirming the request on your Ledger device, a public key and implicit accountId will be displayed.

Example Response Using public key: ed25519:B22RP10g695wye3dfa32rDjmQZEkWTMzAYgCX6oSeB2 Implicit account: 42c320xc20739ASD9a6bqf2Dsaf289z61rd14efe5d3de23213789009afDsd5bb8b0e1

near list-keys

Displays all access keys for a given account. * arguments:accountId

Example:

near list-keys client.chainlink.testnet Example Response Keys for account client.chainlink.testnet [{ public_key: 'ed25519:4wrVrZbHrurMYgkcyusfvSJGLburmaw7m3gmCApxgvY4', access_key: { nonce: 97, permission: 'FullAccess' } }, { public_key: 'ed25519:H9k5eiU4xXS3M4z8HzKJSLaZdqGdGwBG49o7orNC4eZW', access_key: { nonce: 88, permission: { FunctionCall: { allowance: '184832479873450655000000000', receiver_id: 'client.chainlink.testnet', method_names: ['get_token_price', [length]: 1] } } } },

]

near login

locally stores a full access key of an account you created with MyNEARWallet . * arguments:none * options:--networkId

Example:

near login Custom wallet url:

The default wallet URL ishttps://testnet.mynearwallet.com/ . However, if you want to change to a different wallet URL, you can set the environmental variableNEAR_MAINNET_WALLET orNEAR_TESTNET_WALLET .

export NEAR_TESTNET_WALLET=https://wallet.testnet.near.org/ near login

Accounts

near create-account

Creates an account using an existing account or a faucet service to pay for the account's creation and initial balance. * arguments:accountId * options:--initialBalance * ,--useFaucet * ,--useAccount * ,--seedPhrase * ,--publicKey * ,--signWithLedger * ,--ledgerPath * ,--useLedgerPK * ,--PkLedgerPath

Examples: :

Creating account using example-acct.testnet to fund it

near create-account new-acc.testnet --useAccount example-acct.testnet

Creating account using the faucet to fund it

near create-account new-acc.testnet --useFaucet

Creating a pre-funded account that can be controlled by the Ledger's public key

near create-account new-acc.testnet --useFaucet --useLedgerPK

Creating an account using a Ledger account

near create-account new-acc.testnet --useAccount ledger-acct.testnet --signWithLedger Subaccount example:

Using an account to create a sub-account

near create-account sub-acct.example-acct.testnet --useAccount example-acct.testnet

Creating a sub-account using the Ledger that can also be controlled by the ledger

near create-account sub.acc.testnet --useAccount sub.acc.testnet --signWithLedger --useLedgerPK Example using--initialBalance :

near create-account sub-acct2.example-acct.testnet --useAccount example-acct.testnet --initialBalance 10 Example Response Saving key to '/HOME_DIR/.near-credentials/default/sub-acct2.example-acct.testnet.json' Account sub-acct2.example-acct.testnet for network "default" was created.

near delete-account

Deletes an account and transfers remaining balance to a beneficiary account. * arguments:accountId * beneficiaryId * options:force * ,--signWithLedger * ,--ledgerPath

Example:

near delete-account sub-acct2.example-acct.testnet example-acct.testnet Example Response Deleting account. Account id: sub-acct2.example-acct.testnet, node: https://rpc.testnet.near.org, helper: https://helper.testnet.near.org, beneficiary: example-acct.testnet Transaction Id 4x8xohER1E3yxeYdXPfG8GvXin1ShiaroqE5GdCd5YxX To see the transaction in the transaction explorer, please open this url in your browser https://testnet.nearblocks.io/txns/4x8xohER1E3yxeYdXPfG8GvXin1ShiaroqE5GdCd5YxX Account sub-acct2.example-acct.testnet for network "default" was deleted.

near send-near

Sends NEAR tokens (Ⓝ) from one account to another. * arguments:senderId * receiverId * amount * options:--signWithLedger * ,--ledgerPath

Note: You will need a full access key for the sending account. (near login)

Example:

near send-near sender.testnet receiver.testnet 10 Example Response Sending 10 NEAR to receiver.testnet from sender.testnet Transaction Id BYTr6WNyaEy2ykAiQB9P5VvTyrJcFk6Yw95HPhXC6KfN To see the transaction in the transaction explorer, please open this url in your browser https://testnet.nearblocks.io/txns/BYTr6WNyaEy2ykAiQB9P5VvTyrJcFk6Yw95HPhXC6KfN

near state

Shows details of an account's state. * arguments:accountId

Example:

near state example.testnet Example Response { "amount": "99999999303364037168535000", "locked": "0", "code_hash": "G1PCjeQbvbUsJ8piXNb7Yg6dn3mfivDQN7QkvsVuMt4e", "storage_usage": 53528, "storage_paid_at": 0, "block_height": 21577354, "block_hash": "AWu1mrT3eMJLjqyhNHvMKrrbahN6DqcNxXanB5UH1RjB", "formattedAmount": "99.999999303364037168535" }

Contracts

near call

makes a contract call which can modifyor view state. Note: Contract calls require a transaction fee (gas) so you will need an access key for the--accountId that will be charged.(near login)

- arguments:contractName
- method_name
- { args }
- --accountId
- options:--gas
- --deposit
- --signWithLedger
- --ledgerPath

Example:

near call guest-book.testnet addMessage '{"text": "Aloha"}' --account-id example-acct.testnet Example Response Scheduling a call: guest-book.testnet.addMessage({"text": "Aloha"}) Transaction Id FY8hBam2iyQfdHkdR1dp6w5XEPJzJSosX1wUeVPyUvVK To see the transaction in the transaction explorer, please open this url in your browser https://testnet.nearblocks.io/txns/FY8hBam2iyQfdHkdR1dp6w5XEPJzJSosX1wUeVPyUvVK "

near deploy

Deploys a smart contract to a given accountId. * arguments:accountId * .wasmFile * options:initFunction * initArgs * initGas * initDeposit

Note: You will need a full access key for the account you are deploying the contract to. (near login)

Example:

near deploy example-contract.testnet out/example.wasm Initialize Example:

near deploy example-contract.testnet out/example.wasm --initFunction new --initArgs '{"owner_id": "example-contract.testnet", "total_supply": "10000000"}' Example Response Starting deployment. Account id: example-contract.testnet, node: https://rpc.testnet.near.org, helper: https://helper.testnet.near.org, file: main.wasm Transaction Id G8GhhPuujMHTRnwursPXE1Lv5iUZ8WUecwiST1PcKWMt To see the transaction in the transaction explorer, please open this url in your browser https://testnet.nearblocks.io/txns/G8GhhPuujMHTRnwursPXE1Lv5iUZ8WUecwiST1PcKWMt Done deploying to example-contract.testnet

near storage

Shows the storage state of a given contract, i.e. the data stored in a contract. * arguments:contractName * options:--finality * ,--utf8 * ,--blockId * ,--prefix

Example:

near storage hello.near-examples.testnet --finality optimistic --utf8 Example Response [{ key: 'STATE', value: '\x10\x00\x00\x00Passei por aqui!' }]

near view

Makes a contract call which canonly view state.(Call is free of charge) * arguments:contractName * method_name * { args } * options:default

Example:

near view guest-book.testnet getMessages '{}' Example Response View call: guest-book.testnet.getMessages({}) [{ premium: false, sender: 'waverlymaven.testnet', text: 'TGIF' }, { premium: true, sender: 'waverlymaven.testnet', text: 'Hello from New York ' }, { premium: false, sender: 'fhr.testnet', text: 'Hi' }, { premium: true, sender: 'eugenethedream', text: 'test' }, { premium: false 'dongri.testnet', text: 'test' }, { premium: false, sender: 'dongri.testnet', text: 'hello' }, { premium: true, sender: 'dongri.testnet', text: 'hey' }, { premium: false, sender: 'hirokihori.testnet', text: 'hello' }, { premium: true, sender: 'eugenethedream', text: 'hello' }, { premium: false, sender: 'example-acct.testnet', text: 'Aloha' },

]

Transactions

near tx-status

Queries transaction status by hash and accountId. * arguments:txHash * --accountId * options:default

Example:

near tx-status FY8hBam2iyQfdHkdR1dp6w5XEPJzJSosX1wUeVPyUvVK --accountId guest-book.testnet Example Response Transaction guest-book.testnet:FY8hBam2iyQfdHkdR1dp6w5XEPJzJSosX1wUeVPyUvVK { status: { SuccessValue: '' }, transaction: { signer_id: 'example-acct.testnet', public_key: 'ed25519:AXZZKnp6ZcWXyRNdy8FztYrniKf1qt6YZw6mCCReXrDB', nonce: 20, receiver_id: 'guest-book.testnet', actions: [{ FunctionCall: { method_name: 'addMessage', args: 'eyJ0ZXh0IjoiQWxvaGEifQ==', gas: 300000000000000, deposit: '0' } },

], signature: 'ed25519:5S6nZXPU72nzgAsTQLmAFfdVSykdKHWhtPMb5U7duacfPdUjrj8ipJxuRiWkZ4yDodvDNt92wcHLJxGLsyNEsZNB', hash: 'FY8hBam2iyQfdHkdR1dp6w5XEPJzJSosX1wUeVPyUvVK' }, transaction_outcome: { proof: [[length]: 0], block_hash: '6nsjvzt6C52SSuJ8UvfaXTsdrUwcx8JtHfnUj8XjdKy1', id: 'FY8hBam2iyQfdHkdR1dp6w5XEPJzJSosX1wUeVPyUvVK', outcome: { logs: [[length]: 0], receipt_ids: ['7n6wjMgpoBTp22ScLHxeMLzcCvN8Vf5FUuC9PMmCX6yU', [length]: 1], gas_burnt: 2427979134284, tokens_burnt: '242797913428400000000', executor_id: 'example-acct.testnet', status: { SuccessReceiptId: '7n6wjMgpoBTp22ScLHxeMLzcCvN8Vf5FUuC9PMmCX6yU' } } }, receipts_outcome: [{ proof: [[length]: 0], block_hash: 'At6QMrBuFQYgEPAh6fuRBmrTAe9hXTY1NzAB5VxTH1J2', id: '7n6wjMgpoBTp22ScLHxeMLzcCvN8Vf5FUuC9PMmCX6yU', outcome: { logs: [[length]: 0], receipt_ids: ['FUttfoM2odAhKNQrJ8F4tiBpQJPYu66NzFbxRKii294e', [length]: 1], gas_burnt: 3559403233496, tokens_burnt: '355940323349600000000', executor_id: 'guest-book.testnet', status: { SuccessValue: '' } } }, { proof: [[length]: 0], block_hash: 'J7KjpMPzAqE7iX82FAQT3qERDs6UR1EAqBLPJXBzoLCk', id: 'FUttfoM2odAhKNQrJ8F4tiBpQJPYu66NzFbxRKii294e', outcome: { logs: [[length]: 0], receipt_ids: [[length]: 0], gas_burnt: 0, tokens_burnt: '0', executor_id: 'example-acct.testnet', status: { SuccessValue: '' } } },

] }

Global Options

Option Description --help Show help [boolean] --version Show version number [boolean] -v, --verbose Prints out verbose output [boolean] [default: false] Edit this page Last updatedonMar 7, 2024 byDamián Parrino Was this page helpful? Yes No

Previous Unit Tests Next NEAR CLI RS