

Seid CLI

Seid is a command-line tool for interacting with the Sei blockchain. This page will guide you through setting up Seid and introduce you to some basic commands to get started.

Prerequisites

- Go 1.19+: Installation instructions are available [here\(opens in a new tab\)](#)
- .

Installation

To install Seid, locate the version you wish to use [here\(opens in a new tab\)](#) , and then execute the following commands:

```
git
```

```
clone
```

```
https://github.com/sei-protocol/sei-chain cd
```

```
sei-chain git
```

```
checkout
```

```
v4.1.4-evm-devnet make
```

install You can verify that Seid was installed correctly by running:

```
seid
```

version If you encounter an error like `command not found: seid` , you may need to set your GOPATH environment variable.

1. Use `go env GOPATH`
2. to find your GOPATH
3. Add the following lines to your `~/.bashrc`
4. or `~/.zshrc`
5. `:export`
6. `GOPATH`
7. `=`
8. `[your`
9. `GOPATH`
10. `from`
11. `step`
12. `1`
13. `]`
14. `export`
15. `PATH`
16. `=`
17. `PATH`
18. `:`
19. `GOPATH`
20. `/bin`

For more information see [here\(opens in a new tab\)](#) .

Commands

Seid supports all the commands you need to interact with the chain. To see a list of available commands and explanations of what they do, run `seid` :

Start sei app

Usage: `seid [command]`

Available Commands: `add-genesis-account` Add a genesis account to `genesis.json` `add-wasm-genesis-message` Wasm genesis subcommands `collect-gentxs` Collect genesis txs and output a `genesis.json` file `compact` Compact the application DB fully (only if it is a levelDB) `config` Create or query an application CLI configuration file `debug` Tool for helping with debugging your application `export` Export state to JSON `gentx` Generate a genesis tx carrying a self delegation `help` Help

about any command init Initialize private validator, p2p, genesis, and application configuration files keys Manage your application's keys latest_version Prints the latest version of the app DB migrate Migrate genesis to a specified target version prune Prune app history states by keeping the recent heights and deleting old heights query Querying subcommands rollback rollback cosmos-sdk and tendermint state by one height start Run the full node status Query remote node for status tendermint Tendermint subcommands tools A set of useful tools for sei chain tx Transactions subcommands validate-genesis validates the genesis file at the default location or at the location passed as an arg version Print the application binary version information

Flags: -h, --help help for seid --home string directory for config and data (default "/Users/bryantran/.sei") --log_format string The logging format (json|plain) --log_level string The logging level (trace|debug|info|warn|error|fatal|panic) --trace print out full stack trace on errors

Use "seid [command] --help" for more information about a command.

Adding a wallet

You can create a new wallet using theseid keys command:

seid

keys

add NAME Please replaceNAME with the name you would like to use for this key. This will generate a seed phrase and store the account in the CLI for use. Write this mnemonic down and store it in a safe place.

Alternatively, if you would like to import an existing seed phrase, you can add the--recover flag:

seid

keys

add NAME --recover You will then be prompted to input your seed phrase.

⚠ If you are importing from an EVM wallet like MetaMask, you may need to specify the coin type. Ethereum (EVM) based wallets use coin type60 , while the default on Sei is118 . For example,

seid

keys

add NAME --coin-type=60 This will generate a different address than the default coin type of118 . To see your local wallets, you can run

seid

keys

list to see a list of all wallets added, or

seid

keys

show NAME to see a details about a specific wallet.

Last updated onMay 24, 2024 [Ecosystem Map Building a frontend](#)