Namada Indexer

In collaboration withZondax(opens in a new tab) , an indexer for the Namada blockchain has been born.

The Namada indexer (a.k.anamadexer) constantly queries the Namada blockchain, and together with theSDK , is able to map blocks, transactions, along with other valuable information into a relational database (postgres).

This is especially useful for performing analytics over the blockchain, including storing historical data in a way that could be easily queried.

Setting up

The namada indexer's source code can be foundhere(opens in a new tab) and is simple to set up.

Thenamadexer works best together withDocker(opens in a new tab)

git

clone

https://github.com/Zondax/namadexer.git cd

namadexer make

compose

Running the server and db

Once the DockerFile has run, it is straightforward to both set up the postgres database as well as the server that will query the database.

Make sure thatpostgres is installed(opens in a new tab) on the local machine.

Run postgres in docker

make

postgres

or run (and change arguments, e.g port):

docker run --name postgres -e POSTGRES_PASSWORD=wow -e POSTGRES_DB=blockchain -p 5432:5432 -d postgres

Once the postgres server is up and running, it is time to set up the server that will query the postgres db.

Execute the following command in order to set up the server

make run_server If successful, the server should be running as a daemon on the localhost at port30303 .

Run the indexer

First, ensure that theSettings.toml withinconfig/Settings.toml is configured correctly.

log_level

"info" network =

"public-testnet-14"

[database] host =

"0.0.0.0:5435" user =

"postgres" password =

"wow" dbname =

"blockchain"

Optional field to configure a timeout if database connection

fails.

connection_timeout

20

[server] serve_at =

"0.0.0.0" port =

30303

[indexer] tendermint_addr =

"0.0.0.0" port =

26657

[jaeger] enable =

false host =

"localhost" port =

6831

[prometheus] host =

"0.0.0.0" port =

9000 Expand ð Interpreting the toml

It is important to change the following parameters:

1. indexer.tendermint_addr
2.
    - This should be the address and corresponding port of a synced Namada full node
3. database.host
4.
    - This should be the tcp address (with port) where the postgres database is running. Once the setup is complete, it is possible to start the indexer

make

run_indexer

Querying the database

The pre-defined endpoints to query the database are described in the documentation here(opens in a new tab) .

Examples Networks