

NEAR Lake Indexer

NEAR Lake is an indexer built on top of [NEAR Indexer Framework](#) to watch the network and store all the [events](#) as JSON files on AWS S3.

GitHub repo You can find the Lake Indexer source code in [this GitHub repository](#).

How it works

tip [Pagoda Inc.](#) runs NEAR Lake nodes to store the data in JSON format on AWS S3. There is no need to run your own NEAR Lake unless you have specific reasons to do that. There are AWS S3 buckets created:

- near-lake-data-testnet
- (eu-central-1
- region)
- near-lake-data-mainnet
- (eu-central-1
- region)

All the buckets are set up the way the requester pays for the access. Anyone can read from these buckets by connecting to them with their own AWS credentials to be charged by Amazon.

Data structure

The data structure used by Lake Indexer is the following:

/ block.json shard_0.json shard_1.json ... shard_N.json is a 12-character-long [u64](#) string with leading zeros (e.g "000042839521"). See [this issue for reasoning](#).

block.json contains JSON-serialized BlockView struct. NB! this struct might change in the future, we will announce it

shard_N.json where N is [u64](#) starting from 0. Represents the index number of the shard. In order to find out the expected number of shards in the block you can look in block.json at header.chunks_included

How to use it

We have created the [NEAR Lake Framework](#) to have a simple straightforward way to create an indexer on top of the data stored by NEAR Lake itself.

NEAR Lake Framework You can check the NEAR Lake Framework release announcement on the [NEAR Governance Forum](#). We have prepared this video tutorial with a simple example to give you an overview and some practical ideas. [Edit this page](#) Last updated on Sep 1, 2023 by Damián Parrino Was this page helpful? Yes No

[Previous Migrate from Lake framework](#) [Next Getting started](#)