# NEAR Lake indexer basic tutorial

Source code for the tutorial frolvanya/near-lake-raw-printer : source code for the tutorial on how to create an indexer that prints block height and number of shards Recently we havepublished a Python version of the NEAR Lake Frameworkon pypi.org

We want to empower you with a basic tutorial on how to use the Python Package. Let's get started!

## Create a project

Create an indexer project:

```
mkdir near-lake-raw-printer && cd near-lake-raw-printer touch main.py
```

## Install dependencies

Installnear-lake-framework

```
pip3 install near-lake-framework
```

## Importnear-lake-framework

In themain.py file let's import the necessary dependencies:

from near_lake_framework import near_primitives , LakeConfig , streamer We've imported the main functionstreamer which will be called to actually run the indexer,near_primitives andLakeConfig type we need to contruct.

## Create a config

Add the instantiation ofLakeConfig below:

# config

```
LakeConfig . mainnet ( ) config . start_block_height =
```

69030747 config . aws_access_key_id = os . getenv ( "AWS_ACCESS_KEY_ID" ) config . aws_secret_key = os . getenv ( "AWS_SECRET_ACCESS_KEY" ) Just a few words on the config, functionmainnet() has sets3_bucket_name ,s3_region_name for mainnet. You can go toNEAR Explorer and getthe most recent block height to setconfig.start_block_height .

## Starting the stream

Let's callstreamer function with theconfig :

```
stream_handle , streamer_messages_queue = streamer ( config ) while
```

```
True : streamer_message =
```

await streamer_messages_queue . get ( ) print ( f"Block # { streamer_message . block . header . height } Shards: { len ( streamer_message . shards ) } " ) And an actual start of our indexer in the end of themain.py

# loop

```
asyncio . get_event_loop ( ) loop . run_until_complete ( main ( ) )
```

## All together

```
import asyncio import os
```

```
from near_lake_framework import LakeConfig , streamer , near_primitives
```

async

def

main ( ) : config = LakeConfig . mainnet ( ) config . start_block_height =

69030747 config . aws_access_key_id = os . getenv ( "AWS_ACCESS_KEY_ID" ) config . aws_secret_key = os . getenv ( "AWS_SECRET_ACCESS_KEY" )

stream_handle , streamer_messages_queue = streamer ( config ) while

True : streamer_message =

await streamer_messages_queue . get ( ) print ( f"Block # { streamer_message . block . header . height } Shards: { len ( streamer_message . shards ) } " )

# loop

asyncio . get_event_loop ( ) loop . run_until_complete ( main ( ) ) That's it. Now we runmain.py

python3 main.py You should see something like the following:

Received 400 blocks from S3 Block #69030747 Shards: 4 Block #69030748 Shards: 4 Block #69030749 Shards: 4 Block #69030750 Shards: 4 Block #69030751 Shards: 4 Block #69030752 Shards: 4 Block #69030753 Shards: 4 Block #69030754 Shards: 4 You can stop the indexer by pressing CTRL+C

Credentials To be able to access the data fromNEAR Lake you need to provide credentials. Please, see theCredentials article You can find thesource code for this tutorial on the GitHub . Edit this page Last updatedonDec 9, 2023 bygagdiez Was this page helpful? Yes No

Previous JS basic tutorial Next NFT Indexer