Ori

Neocrym Records Inc.

Aug 23, 2020

CONTENTS:

Ori is a high-level wrapper around Python's *concurrent.futures* module, designed to make multithreading and multiprocessing as easy as possible.

CHAPTER

ONE

ORI MODULES

The tools that Ori provides are divided into several modules.

`ori.asyncio` – Tools to integrate Python asyncio code into a synchronous codebase, and vice-versa.

`ori.concurrency` – Tools to run Python functions in the background using multithreading or multiprocessing.

`ori.poolchain` – A way to chain function calls for parallel processing over any list or other iterable.

`ori.subprocess` – Tools for running external commands as subprocesses and efficiently collecting the standard output and standard error.

CHAPTER

TWO

FREQUENTLY ASKED QUESTIONS (FAQS)

Who made Ori?

Ori was written by James Mishra and incubated at Neocrym, a record label that uses artificial intelligence to find and promote musicians. Neocrym heavily relies on Ori to make their I/O-bound Python code run faster.

The source code for Ori is owned by Neocrym Records Inc, but licensed to Ori under the MIT License.

Why should I use Ori over directly interfacing with concurrent.futures?

The Python module concurrent.futures was introduced as a high-level abstraction over lower-level interfaces like *threading.Thread* and *multiprocessing.Process*. However, *concurrent.futures* merely moves the problem away from managing threads or processes to managing *executors*. Ori has the ambitious goal of also abstracting away the executors–making multithreading or multiprocessing no harder than writing single-threaded code.

Is Ori a good replacement for Python's asyncio?

For the hardcore asyncio user, probably not. Ori is focused on providing high-level abstractions over Python's concurrent.futures module that provides speed boosts for synchronous, I/O-bound Python.

What do I need to know to contribute to Ori?

Ori manages itself with the Python packaging tool Poetry. You can install Poetry on your system with:

```
pip3 install poetry
poetry install
```

To check that your changes to Ori's codebase match our coding standards, and to reformat any errant code to meet our standards, run this command:

```
poetry run make lint
```

To run Ori's unit tests in the Python virtualenv created by Poetry, just run:

```
poetry run make test
```

We can also run tests across multiple versions of Python with Tox, but it requres your system has Docker and Docker Compose installed. If so, just run:

```
poetry run make tox
```

Where did the name "Ori" come from?

The name "Ori" is a reference to the god-like villains in the Stargate TV shows. There is no meaningful connection between the villains or concurrency.

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search