
asyncbg Documentation

Release 0.8.0

Erik Moqvist

Aug 03, 2020

Contents

1 Asyncio background tasks	3
2 Installation	5
3 Examples	7
3.1 Call	7
3.2 Process pool	7
4 Functions and classes	9
Index	11

CHAPTER 1

Asyncio background tasks

Asyncio background tasks in Python 3.7 and later.

Run CPU intensive long running tasks without blocking the asyncio loop, implemented as a lightweight asyncio layer on top of the multiprocessing module.

Project homepage: <https://github.com/erimoq/asyncbg>

Documentation: <https://asyncbg.readthedocs.org/en/latest>

CHAPTER 2

Installation

```
pip install asyncbg
```


There are more examples in the `examples` folder.

3.1 Call

Call `work(a, b)` in another process. The script output is `Result: 9`.

```
import asyncio
import asyncbg

def work(a, b):
    return a + b

async def main():
    result = await asyncbg.call(work, 4, 5)
    print(f'Result: {result}')

asyncio.run(main())
```

3.2 Process pool

Create a process pool with two workers, and call `work()` three times in it (up to two callbacks called in parallel).

```
import asyncio
import asyncbg

def work():
    pass

async def main():
```

(continues on next page)

(continued from previous page)

```
pool = asyncbg.ProcessPoolExecutor(max_workers=2)
await asyncio.gather(pool.call(work),
                    pool.call(work),
                    pool.call(work))

asyncio.run(main())
```

Functions and classes

`asyncbg.call (callback, *args, **kwargs)`

Coroutine calling given callback with given arguments in another process.

Returns the value returned by the callback, or raises the exceptions raised by the callback.

Callback positional and keyword arguments can not be used for output, as the multiprocessing module does not support that.

Call `work ()` in a worker process:

```
>>> def work():
>>>     pass
>>>
>>> asyncio.run(asyncbg.call(work))
```

class `asyncbg.ProcessPoolExecutor` (*max_workers=None, mp_context=None, initializer=None, initargs=()*)

Same as `concurrent.futures.ProcessPoolExecutor`, but with the `call ()` method added.

C

`call()` (*in module `asyncbg`*), 9

P

`ProcessPoolExecutor` (*class in `asyncbg`*), 9