
asyncudp Documentation

Release 0.10.0

Erik Moqvist

Jun 15, 2023

CONTENTS

1	Installation	3
2	Example client	5
3	Example server	7
4	Test	9
5	Examples	11
5.1	Client	11
5.2	Server	11
6	Functions and classes	13
	Index	15

Asyncio high level UDP sockets.

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

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

INSTALLATION

```
$ pip install asyncudp
```


EXAMPLE CLIENT

```
import asyncio
import asyncudp

async def main():
    sock = await asyncudp.create_socket(remote_addr=('127.0.0.1', 9999))
    sock.sendto(b'Hello!')
    print(await sock.recvfrom())
    sock.close()

asyncio.run(main())
```


EXAMPLE SERVER

```
import asyncio
import asyncudp

async def main():
    sock = await asyncudp.create_socket(local_addr=('127.0.0.1', 9999))

    while True:
        data, addr = await sock.recvfrom()
        print(data, addr)
        sock.sendto(data, addr)

asyncio.run(main())
```

CHAPTER
FOUR

TEST

```
$ python3 -m unittest
```


EXAMPLES

5.1 Client

```
import asyncio
import asyncudp

async def main():
    sock = await asyncudp.create_socket(remote_addr=('127.0.0.1', 9999))
    sock.sendto(b'Hello!')
    print(await sock.recvfrom())
    sock.close()

asyncio.run(main())
```

5.2 Server

```
import asyncio
import asyncudp

async def main():
    sock = await asyncudp.create_socket(local_addr=('127.0.0.1', 9999))

    while True:
        data, addr = await sock.recvfrom()
        print(data, addr)
        sock.sendto(data, addr)

asyncio.run(main())
```

FUNCTIONS AND CLASSES

async `asyncudp.create_socket`(*local_addr=None, remote_addr=None, packets_queue_max_size=0, reuse_port=None*)

Create a UDP socket with given local and remote addresses.

```
>>> sock = await asyncudp.create_socket(local_addr=('127.0.0.1', 9999))
```

class `asyncudp.Socket`(*transport, protocol*)

A UDP socket. Use `create_socket()` to create an instance of this class.

close()

Close the socket.

sendto(*data, addr=None*)

Send given packet to given address *addr*. Sends to *remote_addr* given to the constructor if *addr* is *None*.

Raises an error if a connection error has occurred.

```
>>> sock.sendto(b'Hi!')
```

async `recvfrom()`

Receive a UDP packet.

Raises `ClosedError` on connection error, often by calling the `close()` method from another task. May raise other errors as well.

```
>>> data, addr = sock.recvfrom()
```

getsockname()

Get bound information.

```
>>> local_address, local_port = sock.getsockname()
```


INDEX

C

`close()` (*asyncudp.Socket method*), [13](#)

`create_socket()` (*in module asyncudp*), [13](#)

G

`getsockname()` (*asyncudp.Socket method*), [13](#)

R

`recvfrom()` (*asyncudp.Socket method*), [13](#)

S

`sendto()` (*asyncudp.Socket method*), [13](#)

`Socket` (*class in asyncudp*), [13](#)