Interacting with Domain Name Services

Applications use the network to communicate with servers for domain name service (DNS) operations like converting between hostnames and IP addresses. `asyncio` has convenience methods on the event loop to take care of those operations in the background, to avoid blocking during the queries.

Address Lookup by Name

Use the coroutine `getaddrinfo()` to convert a hostname and port number to an IP or IPv6 address. As with the version of the function in the socket module, the return value is a list of tuples containing five pieces of information.

1. The address family
2. The address type
3. The protocol
4. The canonical name for the server
5. A socket address tuple suitable for opening a connection to the server on the port originally specified

Queries can be filtered by protocol, as in this example, where only TCP responses are returned.

```
# asyncio_getaddrinfo.py

import asyncio
import logging
import socket
import sys


TARGETS = [
    ('pymotw.com', 'https'),
    ('doughellmann.com', 'https'),
    ('python.org', 'https'),
]


async def main(loop, targets):
    for target in targets:
        info = await loop.getaddrinfo(
            *target,
            proto=socket.IPPROTO_TCP,
        )

        for host in info:
            print('{:20}: {}'.format(target[0], host[4][0]))


event_loop = asyncio.get_event_loop()
try:
    event_loop.run_until_complete(main(event_loop, TARGETS))
finally:
    event_loop.close()
```

The example program converts a hostname and protocol name to IP address and port number.

```
$ python3 asyncio_getaddrinfo.py

pymotw.com          : 66.33.211.242
doughellmann.com    : 66.33.211.240
python.org          : 23.253.135.79
```

Name Lookup by Address

The coroutine `getnameinfo()` works in the reverse direction, converting an IP address to a hostname and a port number to a protocol name, where possible.

```
# asyncio_getnameinfo.py

import asyncio
import logging
import socket
import sys


TARGETS = [
    ('66.33.211.242', 443),
    ('104.130.43.121', 443),
]


async def main(loop, targets):
    for target in targets:
        info = await loop.getnameinfo(target)
        print('{:15}: {} {}'.format(target[0], *info))


event_loop = asyncio.get_event_loop()
try:
    event_loop.run_until_complete(main(event_loop, TARGETS))
finally:
    event_loop.close()
```

This example shows that the IP address for `pymotw.com` refers to a server at DreamHost, the hosting company where the site runs. The second IP address examined is for `python.org`, and it does not resolve back to a hostname.

```
$ python3 asyncio_getnameinfo.py

66.33.211.242  : n821.com https
104.130.43.121 : 104.130.43.121 https
```

See also

- The socket module discussion includes a more detailed examination of these operations.

Using SSL

Working with Subprocesses

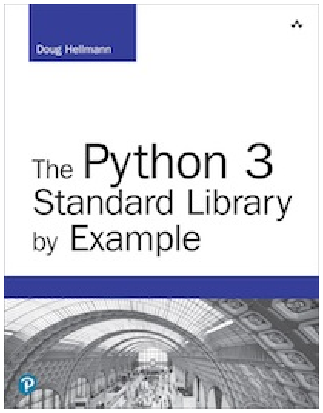
Quick Links

Address Lookup by Name
Name Lookup by Address

This page was last updated 2018-12-09.

Navigation

Using SSL
Working with Subprocesses



Get the book

The output from all the example programs from PyMOTW-3 has been generated with Python 3.7.1, unless otherwise noted. Some of the features described here may not be available in earlier versions of Python.

Looking for examples for Python 2?

This Site

Module Index
Index

© Copyright 2019, Doug Hellmann

Other Writing

Blog
The Python Standard Library By Example