async-dnspy

Release 0.1.1

Jan 15, 2020

Contents

CHAPTER 1

Installation & Testing

1.1 Installation

From PyPI

```
$ pip install asyncdnspy
```

From source

Download the latest *asyncdnspy* library from: https://github.com/abdullahselek/async-dnspy

Extract the source distribution and run:

```
$ python setup.py build
$ python setup.py install
```

1.2 Running Tests

The test suite can be run against a single Python version which requires `pip install pytest` and optionally `pip install pytest-cov` (these are included if you have installed dependencies from `requirements.testing.txt`)

To run the unit tests with a single Python version:

```
$ py.test tests -s -v
```

to also run code coverage:

```
$ py.test tests -s -v --cov-report html --cov=asyncdnspy
```

1.3 Getting the code

The code is hosted at Github.

Check out the latest development version anonymously with:

```
$ git clone https://github.com/abdullahselek/async-dnspy.git
$ cd async-dnspy
```

CHAPTER 2

Modules Documentation

2.1 DNS Message

class asyncdnspy.dns_raw_message.**DNSRawMessage**

Bases: object

Class that creates DNS raw message as byte array.

query(*domain: str*, *record_type: asyncdnspy.dnspy_enum.RecordType*, *socket_type: asyncdnspy.dnspy_enum.SocketType = <SocketType.udp: 2>*)

Creates a raw dns message query.

```
>>> from asyncdnspy.dns_raw_message import DNSRawMessage
>>> dns_raw_message = DNSRawMessage()
>>> data = dns_raw_message.query('google.com', RecordType.a)
```

Parameters

- **domain** (*str*) – Domain to query it's DNS details.
- **record_type** (RecordType) – DNS record type.
- **socket_type** (SocketType) – Socket type.

Return type bytes.

class asyncdnspy.dns_raw_message.**Flag**

Bases: enum.Enum

Enum for DNS flags.

aa = 1

ad = 2

cd = 3

ra = 4

rd = 5

tc = 6

class asyncdnspy.dns_raw_message.**State**
Bases: enum.Enum

Enum for raw message states.

have_answers = 6

have_flags = 3

have_header = 4

have_messageid = 2

have_questions = 5

initial = 1

class asyncdnspy.dns_message_header.**Header**
Bases: object

DNS message header class that has header properties.

additional_count
Additional count property of dns message header.

answer_count
Answer count property of dns message header.

authority_count
Authority count property of dns message header.

flags
Flags property of dns message header.

message_id
Message id property of dns message header.

question_count
Question count property of dns message header.

class asyncdnspy.dns_message_question.**Question**
Bases: object

DNS message question class that has question details.

class_type
Class type property of dns message question.

name
Name property of dns message question.

question_type
Question type property of dns message question.

class asyncdnspy.dns_message_resourcerecord.**ResourceRecord**
Bases: object

Class that holds resource record details.

class_type
Class type property of dns message question.

get_ipv4_address()
Returns ipv4 address if record type A.

get_ipv6_address()
Returns ipv6 address if record type AAAA.

get_txt()
Returns readable TXT if record type TXT.

name
Name property of dns message question.

record_data
Record data property of dns message question.

record_data_length
Length of record data property of dns message question.

record_type
Record type property of dns message question.

ttl
Time to live property of dns message question.

class asyncdnspy.dns_message.**DNSMessage**
Bases: object

additional
Additional property of dns message.

answers
Asnwers property of dns message.

authority
Authority property of dns message.

header
Header property of dns message.

questions
Questions property of dns message.

2.2 DNS Message Decoder

class asyncdnspy.dns_message_decoder.**DNSMessageDecoder**
Bases: object

Decoder class that decodes raw bytes to meaningful dns message.

static decode(*buffer: List[T], received_data: List[T], record_type: asyncdnspy.dnspy_enum.RecordType, socket: asyncdnspy.dnspy_enum.SocketType = <SocketType.udp: 2>*)
Decodes dns message response into a readable dns message.

Parameters

- **buffer** (*list*) – Raw byte list of dns message query.
- **received_data** (*list*) – Raw byte list of dns message response.
- **record_type** (RecordType) – DNS message record type.
- **socket_type** (SocketType) – DNS client socket type.

Return type *DNSMessage*

static decode_name(*bytes: List[T], name_start_index: int, name_length: int*)
Decodes list of unsigned 8-bit array to domain str.

Parameters

- **bytes** (*list*) – Raw byte list of dns header.
- **name_start_index** (*int*) – Starting index of name.
- **name_length** (*int*) – Length of name.

Return type array

offset = 0

static read_header(*bytes: List[T]*)
Reads byte list and creates a high level dns message header.

Parameters **bytes** (*list*) – Raw byte list of dns header.

Return type *Header*

static read_questions(*bytes: List[T], count: int*)
Reads byte list and creates a high level dns message questions list.

Parameters

- **bytes** (*list*) – Raw byte list of dns header.
- **count** (*int*) – Question count.

Return type list (list of question/s)

static read_resource_records(*buffer: List[T], received_data: List[T], count: int*)
Reads byte list and create resource records.

Parameters

- **buffer** – Raw byte list of sent dns message.
- **received_data** (*list*) – Raw byte list of sent dns message response.
- **count** (*int*) – Question count.

Return type list (list of resource record/s)

socket_type = 2

static string(*bytes: List[T]*)
Creates str from bytes.

Parameters **bytes** (*bytes*) – List of raw bytes.

Return type str

static ui16(*bytes: List[T]*)
Creates unsigned 16 bit int.

Parameters **bytes** (*list*) – List of raw bytes.

Return type unsigned 16 bit int

static ui32 (*bytes: List[T]*)
Creates unsigned 32 bit int.

Parameters **bytes** (*list*) – List of raw bytes.

Return type unsigned 32 bit int

2.3 Enums

class asyncdnspy.dnspy_enum.**RecordType**
Bases: enum.Enum

DNS message and response record types.

a = 1

aaaa = 28

afsdb = 18

apl = 42

caa = 257

cdnskey = 60

cds = 59

cert = 37

cname = 5

csync = 62

dhcid = 49

dlv = 32769

dname = 39

dnskey = 48

ds = 43

hip = 55

ipseckey = 45

key = 25

kx = 36

loc = 29

mx = 15

naptr = 35

ns = 2

nsec = 47

nsec3 = 50

nsec3param = 51

openpgpkey = 61

ptr = 12

rp = 17

rrsig = 46

sig = 24

smimea = 53

soa = 6

srv = 33

sshfp = 44

ta = 32768

tkey = 249

tlsa = 52

tsig = 250

txt = 16

uri = 256

class asyncdnspy.dnspy_enum.**SocketType**

Bases: enum.Enum

Socket types.

tcp = 1

udp = 2

2.4 Clients

class asyncdnspy.udp_client.**UDPClient**(*address: str*, *port: int*, *blocking: bool = False*, *timeout: int = 10*)

Bases: object

UDP client for connecting to a DNS server over UDP.

close()

Closes the socket connection.

```
>>> from asyncdnspy.udp_client import UDPClient
>>> udp_client = UDPClient('8.8.8.8', 53)
>>> udp_client.connect()
>>> udp_client.close()
```

connect()

Connects to given address and port.

```
>>> from asyncdnspy.udp_client import UDPClient
>>> udp_client = UDPClient('8.8.8.8', 53)
>>> udp_client.connect()
```

receive(*bufsize: int = 1024*, *flags: int = 0*)

Receives result of send data.

```
>>> from asyncdnspy.udp_client import UDPClient
>>> from asyncdnspy.dns_raw_message import DNSRawMessage
>>> from asyncdnspy.dnspy_enum import RecordType
```

```
>>> udp_client = UDPClient('8.8.8.8', 53)
>>> udp_client.connect()
>>> dns_raw_message = DNSRawMessage()
>>> data = dns_raw_message.query('google.com', RecordType.a)
>>> result = udp_client.send(data)
>>> response = udp_client.receive()
```

Parameters

- **bufsize** (*int*) – Buffer size.
- **flags** (*int*) – Flags to send data.

Return type bytes

send(*data: bytes*, *flags: int = 0*)

Send bytes over UDP.

```
>>> from asyncdnspy.udp_client import UDPClient
>>> from asyncdnspy.dns_raw_message import DNSRawMessage
>>> from asyncdnspy.dnspy_enum import RecordType
```

```
>>> udp_client = UDPClient('8.8.8.8', 53)
>>> udp_client.connect()
>>> dns_raw_message = DNSRawMessage()
>>> data = dns_raw_message.query('google.com', RecordType.a)
>>> result = udp_client.send(data)
```

Parameters

- **data** (*bytes*) – Bytes created by DNSRawMessage.
- **flags** (*int*) – Flags to send data.

Return type int

class asyncdnspy.tcp_client.**TCPClient**(*address: str*, *port: int*, *blocking: bool = False*, *timeout: int = 10*)

Bases: object

TCP client for connecting to a DNS server over TCP.

close()

Closes the socket connection.

```
>>> from asyncdnspy.tcp_client import TCPClient
>>> tcp_client = TCPClient('8.8.8.8', 53)
>>> tcp_client.connect()
>>> tcp_client.close()
```

connect()

Connects to given address and port.

```
>>> from asyncdnspy.tcp_client import TCPClient
>>> tcp_client = TCPClient('8.8.8.8', 53)
>>> tcp_client.connect()
```

receive(*bufsize: int = 1024*, *flags: int = 0*)

Receives result of send data.

```
>>> from asyncdnspy.tcp_client import TCPClient
>>> from asyncdnspy.dns_raw_message import DNSRawMessage
>>> from asyncdnspy.dnspy_enum import RecordType
```

```
>>> tcp_client = TCPClient('8.8.8.8', 53)
>>> tcp_client.connect()
>>> dns_raw_message = DNSRawMessage()
>>> data = dns_raw_message.query('google.com', RecordType.a, socket_
↪type=SocketType.tcp)
>>> result = tcp_client.send(data)
>>> response = tcp_client.receive()
```

Parameters

- **bufsize** (*int*) – Buffer size.
- **flags** (*int*) – Flags to send data.

Return type bytes

send(*data: bytes*, *flags: int = 0*)

Send bytes over UDP.

```
>>> from asyncdnspy.tcp_client import TCPClient
>>> from asyncdnspy.dns_raw_message import DNSRawMessage
>>> from asyncdnspy.dnspy_enum import RecordType
```

```
>>> tcp_client = TCPClient('8.8.8.8', 53)
>>> tcp_client.connect()
>>> dns_raw_message = DNSRawMessage()
>>> data = dns_raw_message.query('google.com', RecordType.a, socket_
↪type=SocketType.tcp)
>>> result = tcp_client.send(data)
```

Parameters

- **data** (*bytes*) – Bytes created by DNSRawMessage.
- **flags** (*int*) – Flags to send data.

Return type int

CHAPTER 3

Introduction

Pure Python asynchronous DNS client and server library. It works with Python 3.x.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index

G

H

I

K

L

M

N

O

P

Q

R

S

T

U