
requests-doh

Release 0.3.1

mansuf

Jun 11, 2023

CONTENTS

1	Installation	3
1.1	Installation	3
2	API usage	5
2.1	API usage	5
3	API reference	7
3.1	API Reference	7
4	DoH (DNS-over-HTTPS) providers	11
4.1	DoH (DNS-over-HTTPS) providers	11
4.2	Changelog	12
	Index	15

DNS over HTTPS resolver for python requests using [dnspython](#) module

INSTALLATION

1.1 Installation

1.1.1 Stable version

With PyPI

```
# For Windows
py -3 -m pip install requests-doh

# For Linux / Mac OS
python3 -m pip install requests-doh
```

1.1.2 Development version

Warning: This version is not stable and may crash during run.

With PyPI & Git

NOTE: You must have git installed. If you don't have it, install it from here <https://git-scm.com/>.

```
# For Windows
py -3 -m pip install git+https://github.com/mansuf/requests-doh.git

# For Linux / Mac OS
python3 -m pip install git+https://github.com/mansuf/requests-doh.git
```

With Git only

NOTE: You must have git installed. If you don't have it, install it from here <https://git-scm.com/>.

```
git clone https://github.com/mansuf/requests-doh.git
cd requests-doh
python setup.py install
```


2.1 API usage

2.1.1 Easy usage

```
# for convenience
from requests_doh import DNSOverHTTPSSession

# By default, DoH provider will set to `cloudflare`
session = DNSOverHTTPSSession(provider='google')
r = session.get('https://google.com')
print(r.status_code)
```

2.1.2 Basic usage with adapters

```
import requests
from requests_doh import DNSOverHTTPSAdapter

adapter = DNSOverHTTPSAdapter(provider='cloudflare-security')
session = requests.Session()
# For HTTPS
session.mount('https://', adapter)
# For HTTP
session.mount('http://', adapter)

r = session.get('https://google.com')
print(r.status_code)
```

2.1.3 Add or remove custom DoH (DNS over HTTPS) provider

```
import requests
from requests_doh import DNSOverHTTPSSession, add_dns_provider, remove_dns_provider

# Adding a new DoH provider
add_dns_provider("another-dns", "https://another-dns.example.com/dns-query")
```

(continues on next page)

(continued from previous page)

```
session = DNSOverHTTPSSession("another-dns")
r = session.get("https://google.com/")
print(r.status_code)
```

```
import requests
from requests_doh import DNSOverHTTPSSession, add_dns_provider, remove_dns_provider

# Remove DoH provider
remove_dns_provider("another-dns", fallback="cloudflare")
```

3.1 API Reference

3.1.1 Session

`class requests_doh.DNSOverHTTPSSession(*args, **kwargs)`

A ready-to-use DoH (DNS-over-HTTPS) `requests.Session`

Parameters

- **provider** (`str`) – A DoH provider
- **cache_expire_time** (`float`) – Set DNS cache expire time

3.1.2 Adapters

`class requests_doh.DNSOverHTTPSAdapter(provider=None, cache_expire_time=None, **kwargs)`

An DoH (DNS over HTTPS) adapter for `requests.Session`

Parameters

- **provider** (`str`) – A DoH provider
- **cache_expire_time** (`float`) – Set DNS cache expire time
- ****kwargs** – These parameters will be passed to `requests.adapters.HTTPAdapter`

3.1.3 DNS resolver session

`requests_doh.set_resolver_session(session)`

Set http session to resolve DNS

Parameters `session` (`requests.Session`) – An http session to resolve DNS

Raises `ValueError` – session parameter is not `requests.Session` instance

`requests_doh.get_resolver_session()`

Returns Return an http session for DoH resolver

Return type `requests.Session`

3.1.4 DoH (DNS-over-HTTPS) Provider

`requests_doh.add_dns_provider(name, address, switch=False)`

Add a DoH provider

Parameters

- **name** (`str`) – Name for DoH provider
- **address** (`str`) – Full URL / endpoint for DoH provider
- **switch** (Optional[`bool`]) – If True, the DoH provider will automatically switch to newly created DoH provider

`requests_doh.remove_dns_provider(name, fallback=None)`

Remove a DoH provider

If parameter name is an active DoH provider, `get_dns_provider()` will return None. You must set `fallback` parameter to one of available DoH providers (`fallback` and `name` parameters cannot be same value) or you can call `set_dns_provider()` after calling this function in order to get DoH working

For example:

```
from requests_doh import DNSOverHTTPSSession, add_dns_provider, remove_dns_provider

# Add a custom DNS and set it to active
add_dns_provider("another-dns", "https://another-dns.example.com/dns-query",
↪switch=True)

# At this point, the session is still working
session = DNSOverHTTPSSession("another-dns")
r = session.get("https://example.com")
print(r.status_code)

# Let's try to remove the newly created DNS
remove_dns_provider("another-dns", fallback="cloudflare")

# Or we can call `set_dns_provider()`
# if we didn't set `fallback` parameter
# set_dns_provider("cloudflare")

# At this point DoH provider "another-dns" is removed
# and "cloudflare" is set to active DoH provider
# the session is still working
r = session.get("https://google.com")
```

But what will happen if we didn't add `fallback` parameter or didn't call `set_dns_provider()`? Well error will occurred, take a look at this example:

```
from requests_doh import DNSOverHTTPSSession, add_dns_provider, remove_dns_provider

# Add a custom DNS and set it to active
add_dns_provider("another-dns", "https://another-dns.example.com/dns-query",
↪switch=True)

# At this point, the session is still working
```

(continues on next page)

(continued from previous page)

```

session = DNSOverHTTPSSession("another-dns")
r = session.get("https://example.com")
print(r.status_code)

# Let's try to remove the newly created DNS
remove_dns_provider("another-dns")

# If we send request to this URL, it would still working
r = session.get("https://example.com")
print(r.status_code)

# An error occurred when we send to another URL
# Because we didn't set `fallback` parameter in `remove_dns_provider()`
# (or calling function `set_dns_provider()`)
# `get_dns_provider()` will return `None` and thus resolving DNS will be failed
# Because there is no valid endpoint where we wanna resolve DNS of the host
r = session.get("https://google.com")

```

Parameters

- **name** (*str*) – DoH provider that want to remove
- **fallback** (*str*) – Set a fallback DoH provider

Raises *DoHProviderNotExist* – DoH provider is not exist in list of available DoH providers

`requests_doh.set_dns_provider(provider)`

Set a DoH provider, must be a valid DoH providers

Parameters **provider** (*str*) – An valid DoH provider, see *DoH (DNS-over-HTTPS) providers*

Raises *DoHProviderNotExist* – Invalid DoH provider

`requests_doh.get_dns_provider()`

Returns Return current DoH provider

Return type *str*

`requests_doh.get_all_dns_provider()`

Returns Return all available DoH providers

Return type *tuple[str]*

3.1.5 DNS Cache

`requests_doh.set_dns_cache_expire_time(time)`

Set DNS cache expired time in seconds

Parameters **time** (*float*) – An expire time

`requests_doh.purge_dns_cache(host=None)`

Purge DNS cache

Parameters **host** (*str*) – Cached DNS host want to be purged, if *host* is *None*, all DNS caches will be purged.

3.1.6 Exceptions

exception `requests_doh.RequestsDOHException`

Base exception for requests_doh library

exception `requests_doh.DNSQueryFailed`

Failed to query DNS from given host

exception `requests_doh.DoHProviderNotExist`

DoH provider is not exist in list of available DoH providers

DOH (DNS-OVER-HTTPS) PROVIDERS

4.1 DoH (DNS-over-HTTPS) providers

google

Basic google DNS (8.8.8.8 and 8.8.4.4)

cloudflare

Basic cloudflare DNS (1.1.1.1 and 1.0.0.1)

cloudflare-security

cloudflare DNS with malware protection (1.1.1.2 and 1.0.0.2)

cloudflare-family

cloudflare DNS with malware protection and blocking adult content (1.1.1.3 and 1.0.0.3)

opendns

Basic OpenDNS (208.67.222.222 and 208.67.220.220)

opendns-family

OpenDNS with adult content filter (208.67.222.123 and 208.67.220.123)

adguard

Default AdGuard DNS with ads, tracking and phishing protection (94.140.14.14 and 94.140.15.15)

adguard-family

AdGuard DNS with default features + adult content filter + safe search (94.140.14.15 and 94.140.15.16)

adguard-unfiltered

AdGuard DNS with no default features and family protection (94.140.14.140 and 94.140.14.141)

quad9

Default Quad9 DNS with malware protection (9.9.9.9 and 149.112.112.112)

quad9-unsecured

Quad9 DNS with no malware protection (9.9.9.10 and 149.112.112.10)

4.2 Changelog

4.2.1 v0.3.1

This update fix requests dependencies because of [CVE-202-32681](#)

Dependencies

- Bump requests from v2.28.2 to v2.31.0

4.2.2 v0.3.0

New features

- Added ability to add custom DNS over HTTPS provider #1
- Added ability to remove DNS over HTTPS provider

Improvements

- Improved performance for querying DNS over HTTPS

4.2.3 v0.2.4

Dependencies

- Updated requests from v2.28.1 to v2.28.2
- Updated dnspython from v2.2.1 to v2.3.0

4.2.4 v0.2.3

Fix bugs

- Fixed missing dependencies resulting error `dns.query.NoDOH: Neither httpx nor requests is available.`

4.2.5 v0.2.2

Improvements

- Improved DoH resolving

4.2.6 v0.2.1

Fix bugs

- Fixed unhandled exception if host doesn't contain AAAA type

4.2.7 v0.2.0

Warning: Broken, do not use this version. Instead use v0.2.1

New features

- Added `get_all_dns_provider()`, returning all available DoH providers.
- Added new DoH providers
 - cloudflare-security
 - cloudflare-family
 - opendns
 - opendns-family
 - adguard
 - adguard-family
 - adguard-unfiltered
 - quad9
 - quad9-unsecured
- Added `DNSOverHTTPSSession` for ready-to-use DoH requests session

Breaking changes

- Starting from v0.2.0, requests-doh rely on `dnspython` module for extending it's library usage and query to many public and private DNS.

4.2.8 v0.1.1

Fix bugs

- Fix ipv6 addresses is not handled properly

4.2.9 v0.1.0

New features

- Added DoH local caching

Fix bugs

- Fix requests for http prefix (`http://`) is hanging up

4.2.10 v0.0.1

Initial release

A

add_dns_provider() (in module requests_doh), 8
 adguard
 command line option, 11
 adguard-family
 command line option, 11
 adguard-unfiltered
 command line option, 11

C

cloudflare
 command line option, 11
 cloudflare-family
 command line option, 11
 cloudflare-security
 command line option, 11
 command line option
 adguard, 11
 adguard-family, 11
 adguard-unfiltered, 11
 cloudflare, 11
 cloudflare-family, 11
 cloudflare-security, 11
 google, 11
 opendns, 11
 opendns-family, 11
 quad9, 11
 quad9-unsecured, 11

D

DNSOverHTTPSAdapter (class in requests_doh), 7
 DNSOverHTTPSSession (class in requests_doh), 7
 DNSQueryFailed, 10
 DoHProviderNotExist, 10

G

get_all_dns_provider() (in module requests_doh), 9
 get_dns_provider() (in module requests_doh), 9
 get_resolver_session() (in module requests_doh), 7
 google
 command line option, 11

O

opendns
 command line option, 11
 opendns-family
 command line option, 11

P

purge_dns_cache() (in module requests_doh), 9

Q

quad9
 command line option, 11
 quad9-unsecured
 command line option, 11

R

remove_dns_provider() (in module requests_doh), 8
 RequestsDOHException, 10

S

set_dns_cache_expire_time() (in module requests_doh), 9
 set_dns_provider() (in module requests_doh), 9
 set_resolver_session() (in module requests_doh), 7